Theory + Practice Workbook

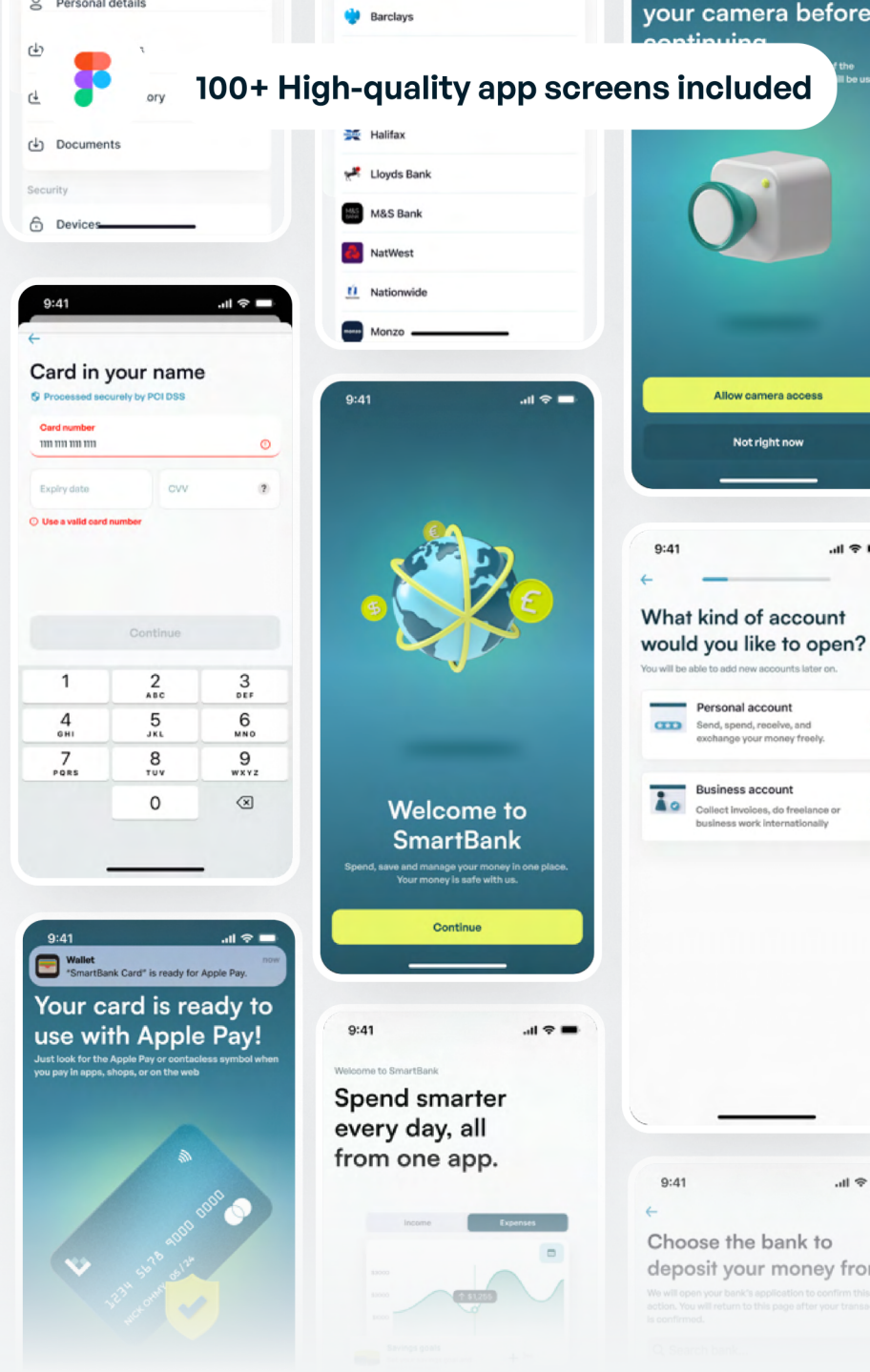
DESIGN.RIP
無料教育

the UI Professional's

Design Manual

the ultimate guide to master mobile design

Learn and practice mobile design on a real Figma project
with auto-layout and responsive constraints.



iOS Theory | Fintech UI Kit | Auto-layout ready | UX/UI Design Templates | Style Sheet | Design System | Figma Practice Files

Table of Contents

Theory

Introduction 7

Who is this ebook for and why use this format? 15

A few words about resources 21

UI vs. UX 23

Design roles 33

Good vs bad design 39

How to understand design 47

What makes a design good 51

Get better at design quick 55

iOS Design princinples 65

iOS Elements 103

Figma design basics 121

Auto-layout 1 on 1 133

Constraints 143

Fintech App

Your Fintech UI Kit 153

Intro to project 157

Project brief 161

Lean UX - our approach 164

Our design framework 169

UX research & market analysis 171

UX process overview 179

Paper sketches 189

Low vs Mid vs High-fidelity Wireframes 191

Inspiration board & Mood boards 195

Style guide and brand identity guidelines 205

Grids & layout 219

Please, play the video first! 225

Prepare your practice files 227

Ready, set, Figma! 233

Design Manual

Launch Screen 241

Onboarding 255

The Sign up 275

Account setup 303

Phone verification 327

Legal information 348

Verify identity 379

Selfie verification 407

Order a card 427

Top up account 455

Home 489

Open a balance 527

Money transfer 549

Profile settings 569

Upload avatar 595

Outro

Design hand-off 603

What's next? 611

Final word 621

If you're reading this right now, the final version of this ebook hasn't been released yet. I'm currently writing 10 more chapters to explain our Fintech app flows and have two more bonuses in store for you! In the meantime, feel free to suggest a chapter you'd like to read about. Send it over to hello@uiadrian.com, and I'll make sure to include it!

All future updates and bonuses are completely free. I'll be sending out emails to let you know about new releases.

Thank you again for purchasing this ebook; I hope you'll like it!

To find all bonuses, templates, and mock-ups, look at the files provided with this e-book. I'm sharing my entire design framework, so make sure you don't miss anything :)

Copyright © 2022 uiadrian. All rights reserved.

No portion of this e-book may be reproduced in any form without permission from the publisher (myself). For permissions to distribute or translate this e-book, contact: hello@uiadrian.com.

Hello there! 👋

Adrian here. I am one of the designers behind this eBook. I just wanted to personally thank you for purchasing “the UI’s Professional Design Manual”!

Thank you so much for getting yourself a copy; now I’ll let on my secret design process :)

This e-book is inspired by my own experiences, struggles, and overwhelming confusion I’ve been through at the beginning of my app design journey, to running a design subscription agency and then landing a job as a product design lead in one of the fastest-growing product design agencies in London (shoutout to Solace!), as well as getting myself to the point of living a very comfortable life at the comfort of my own desk with only my laptop (and my wife) by my side.

The last part is not easily replicable, though, so I can’t promise you’ll find yourself a wife (or husband) after reading this e-book 😂. You will, however, become more valuable to design agencies and clients alike, knowing to properly design an entire app from start to finish using the best iOS design principles.

Let’s give you a proper introduction first.



This is Natalie - she’s the one who meticulously planned all the contents of this e-book and made sure everything is where it needs to be!

… and that’s me → Adrian, you probably know me by that 3D face popping up in your IG feed every now and then.

Nice to meet you, friend!



We’re really flattered to see you here and we are forever grateful for all your continuous support. Without you, we wouldn’t have been able to write this e-book and it still feels surreal to think that we’ve built such an amazing community in just 8 months! Thank you so much for being a part of it. ❤️

Why did we write this ebook?

I hope this won't sound the wrong way - Instagram is full of creators preaching design best practices and sharing design tips without actually being a designer by trade. Now, I'm not saying you have to work in this industry to be posting helpful content on social media. That's not the point. You don't have to be a designer to post about design tips, but you might need to build an app or two before you teach others about app design. Isn't it like learning about business from someone who never really ran a real business himself?

I really do enjoy all the helpful tips and guides people put out, and it's a fantastic thing seeing our design community grow so fast. The problem is that most of the information has already been posted before, and it's just repurposed to appeal to new audiences. I know this because I've been guilty of that myself, searching for inspiration for my next big post, looking at posts of the big players (shoutout to @ui.sergio, @designill4u, @ui.martin, and a bunch of others talented designers) seeing what performs best on Explore and what doesn't, and building my content around a similar topic.

That's why I want to slowly transition into more hands-on, real-life scenario guides where you might actually use this new knowledge and put it into practice. That's also why this ebook was born. We were missing a guide on approaching a project, using a design framework to plan it, design it, execute it, and present it to get more exposure.

A project in line with iOS standards, with well-structured design files and an organized and easy-to-follow design process.

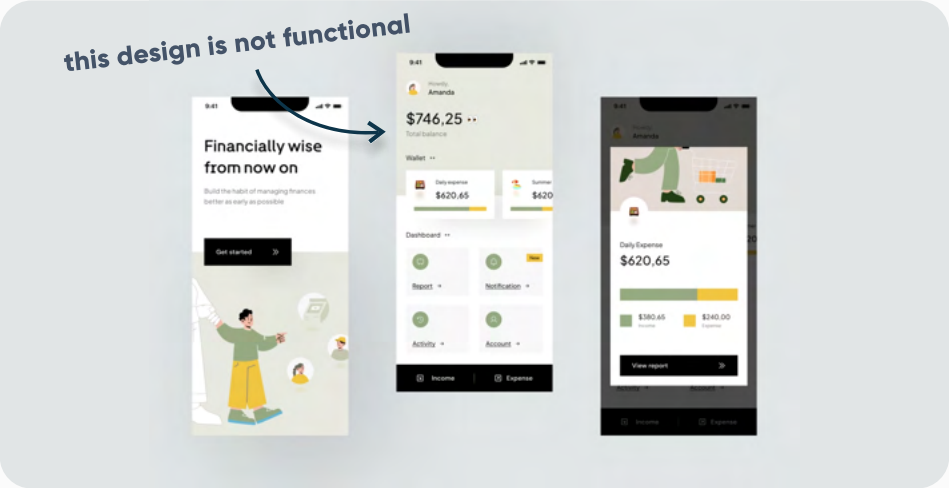
This is why in this e-book, we're going to go through the very basics of iOS design (which you shouldn't skip, by the way!) and only then move on to a live project, taking you through the entire design process, from the start to the design hand-off, while actively following my steps and designing with me, so that you can get the most out of it.

All the stages of the design process are supplemented by an appropriate template that you can use in Figma for your own future projects. You have also received a file with our Fintech UI Kit (100+ screens and growing) and a practice file with your very own practice frames where we will replicate the entire design by following the Design Manual part of this e-book. And if you ever decide you need additional information on the design basics, I suggest you read my free e-book (it's in the e-book folder called "How to design better") or check out my Instagram (@uiadrian) - I cover all there is to know about UX and UI - elements, patterns, and components that you can find in every design.

This e-book is divided into 4 segments - the first two segments are theoretical, going over the UI/UX basics, specifically applied to the iOS app design, and the Fintech app preparation. The last two are more practical, where we get our hands dirty and replicate the Fintech App ourselves.

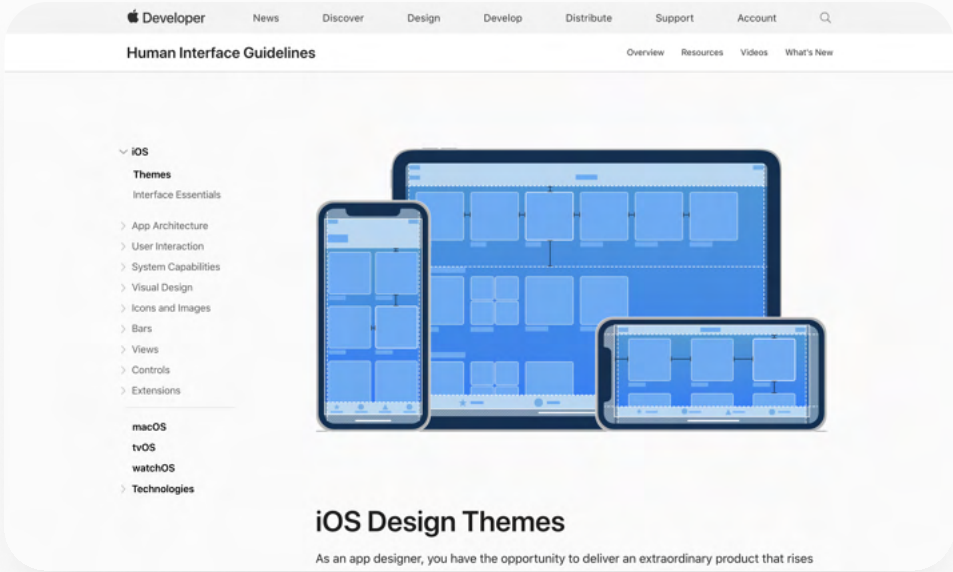
You might be thinking, why am I going through the basic principles of iOS if this e-book is all about designing an app?

Well, let's be clear. You won't get better at UI design without understanding the principles that dictate the industry's standards. You need to understand iOS design and see the difference between the shiny but not-so-functional designs on Dribbble and Behance and the daily apps that don't utilize the same crazy blurs and shadows, oversaturated gradients, random font sizes, huge whitespaces, super fancy shapes, and beautiful stock images.



A random top performing post on Dribbble

You might be super talented and recognize good design patterns without understanding them correctly. Still, there will come a time when your superficial knowledge will be questioned by another product designer, senior designer, or developer that will see the gaps in your understanding of the iOS system and the design basics.



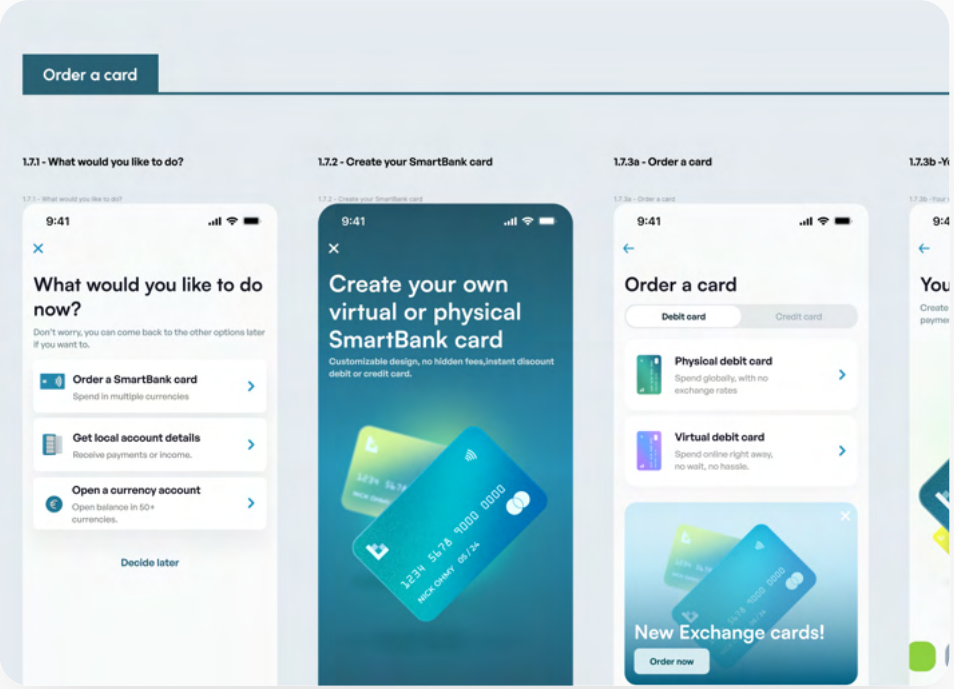
Human Interface Guidelines website

Human Interface Guidelines are here to help you avoid that. And I'm here to help you understand the most important HIG principles in a more streamlined, visual, and simplified way.

I also feel like a disclaimer is due here. By no means I'm not saying that this is the one and only approach to designing a fintech app, at least from the UX perspective. It's worth noting that I've just finished working on a huge app for the biggest savings bank in the UK, so it's safe to assume that some data and knowledge we gained working on that project was used to design this Fintech app. I have to highlight that all our design decisions are not backed by any accurate testing, and the design research and planning phase was simplified for obvious reasons - not to make this e-book too long and boring.

If enough people are interested in the UX, we'll probably write a new book to explain the UX process in much more detail. But for now, we'll go through most UX stages, and to keep things simple, we're mainly focusing on the design part. Otherwise, this e-book would be 800 pages long (which it will, after the 2nd update 😂), and I feel you'd easily get lost in the process.

All the screens we're going to be thoroughly analyzing will be named so that you can locate them quickly in your practice file and follow along. I suggest you fire up your Figma and design in parallel with me. I'll tell you how to set everything up when it's time, so don't worry about it for now.



*Please note that this is a really ambitious project covering theory, UX research basics, and UI design of a really complicated app in a demanding niche - Fintech. I wanted to share as much of my design process with you as possible and guide you through it, but in real life, there's little to no possibility that you'd be working on this kind of app all by yourself.

In a real-life setting, the project would be proceeded with a kick-off meeting, a UX workshop, user interviews, market research, SWOT analysis, stakeholder meetings, storyboarding, and mapping out user journeys. There are around 50-100 hours of work we're deliberately skipping here to focus on the UI more than the UX.

A quick encouragement for my designer newbies 👇

I know it feels daunting to start in UI (or UX) without experience, because how can you get experienced without working in the industry, and how can you work in the industry without getting the experience first? The reality is simple, you have to practice, publish your work, and get noticed. But to get noticed, you have to design "good" interfaces; by good, I mean standardized and in line with basic design principles. The following chapters will show you the exact way to design good interfaces by using the best design principles, following a simple design process, and adhering to iOS guidelines.

Let's now talk about who will benefit from this e-book the most and why we've used the writing format you see here.

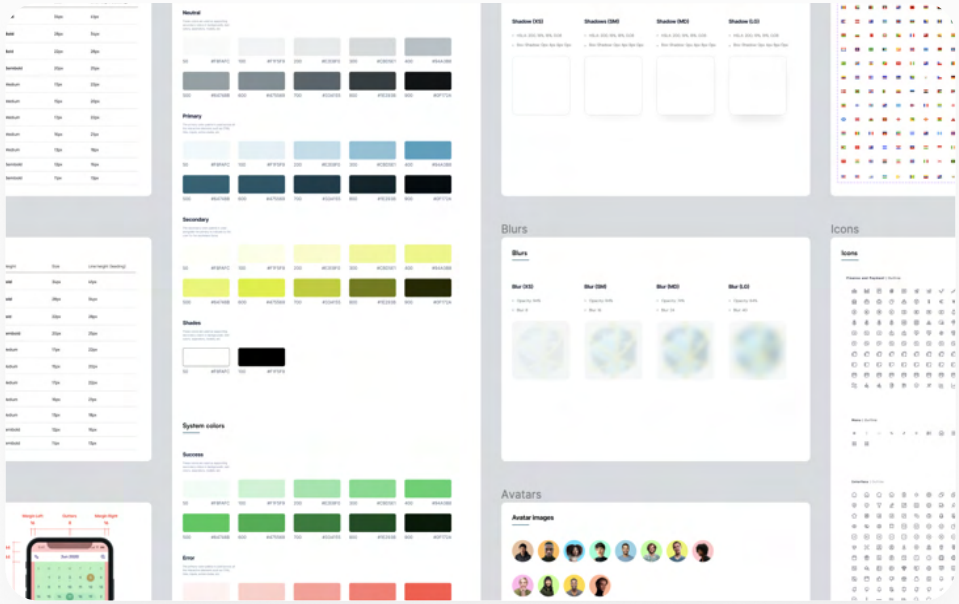
Who is this for?

Who is this ebook for?

This e-book is for every creative person. It doesn't matter whether you are already a UI or UX designer or a graphic designer or if you're from a completely non-related industry. This e-book will give you a basic understanding of good design principles, an in-depth look at the iOS design guidelines, and hands-on practical knowledge to create a full-fledged iOS app made in Figma that's fully responsive and aligned with Human Interface Guidelines (In a later release, we'll also make it into a prototype!).

This book will show you the building blocks you need to obtain to become a better designer and help you design your own iOS concept projects in the future. We wrote this e-book thinking about everyone who's lost in the learning process, looking for answers everywhere but getting bits of unrelated information that's so spread out you can't make any real sense of it.

We've compiled all the information you need to create a successful design and show you the exact design process I go through with every app I'm working on. In the next 600 pages we cover almost all there is to know about UI design, and iOS design in particular, so there's a lot to process, practice and learn from. This e-book alone should give you a very good foundation for designing better interfaces as our final test of knowledge requires you to design an entire Fintech App from start to finish using our Design Manual and a Practice file.

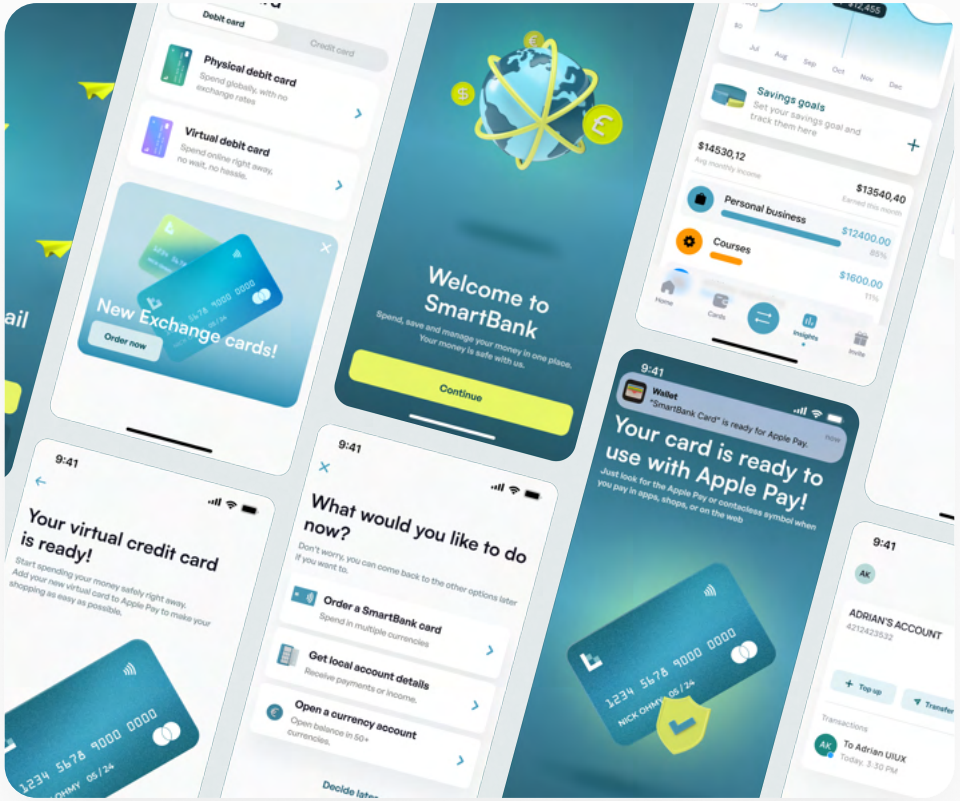


Style guide- Fintech UI Kit

We'll take you through market research, UX audits, user personas, information architecture, user flows, user journeys, inspiration boards, mood boards, brand guidelines, style guides, brainstorming solutions, building wireframes, working on mid-fidelity designs, and taking them into high-fidelity, prototyping (2nd release) and creating a hand-off documentation file for developers.

To supplement the theoretical knowledge you'll get by reading this e-book, we've also added templates for each of the design stages I've described, so you can get started right away and build your own products from the get-go.

On top of that, the cherry on top is the Fintech UI Kit, 100+ high-quality, responsive screens that we're going to replicate by following the practical part of this e-book. We've fully designed and explained 15 chapters, but 9 more are coming in the second release.



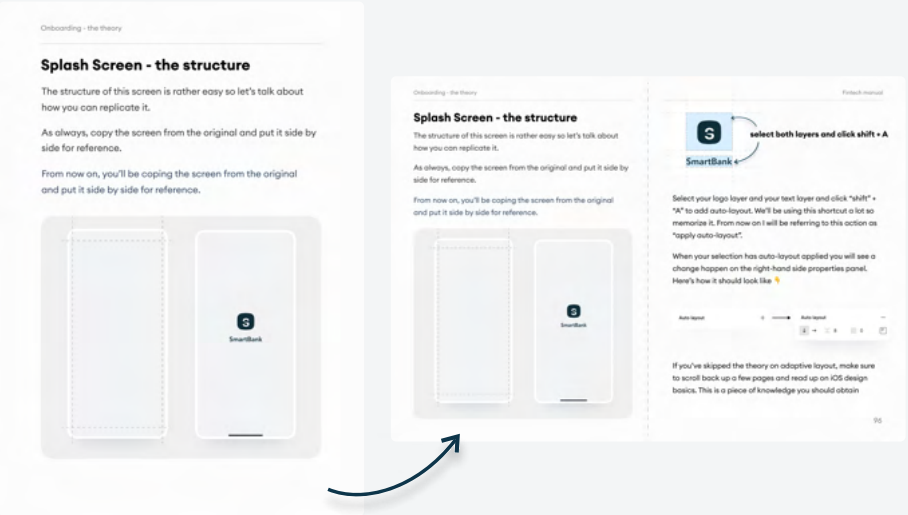
Fintech UI Kit - SmartBank

Our UI Kit and this e-book will evolve in time, and we'll keep releasing new updates to it. Treat it as a real MVP that will be iterated, tested, and improved upon, and you, being the first supporter of this work, will get lifetime free updates, no questions asked, no strings attached.

Why use this ebook format?

When we started compiling all the information from the practice part and explaining our design decisions and the structure of each layout, we found that having a regular vertical A4 format created issues while showing all the visuals we wanted to display on a single page. It was practically impossible.

A lot of the designs we describe are hard to understand when you don't see both pages and the information we're referencing simultaneously. You would need to scroll up and down to look at the sections from a previous part to get additional context.



We wanted this e-book to feel like a real book you read holding in your hands. If you ever find it too difficult to read it like that, please let us know, and we'll try to release an additional, regular A4 version. Also, make sure your preview program is set to fit the screen or adjust the zoom values manually to find the perfect resolution for you to read. It's easiest to consume if the content spans across the entire screen.

Hope you'll enjoy it as much as we enjoyed writing it for you because we had a blast!



We've put close to 600 hours into writing this piece, and it would mean the world to us if you could leave a positive review when you've finished reading it (and enjoyed it!). It will help us create more content in the future, and it will bring us enormous pleasure to know that this e-book has helped you in any way.

If you, however, found it lacking or not up to your expectations, please reach out to me at hello@uiadrian.com. I'd love to hear your feedback on how we can improve this e-book for future releases.

Thank you! 🙏

Resources list

As you might have noticed, we've attached a number of different resources to this e-book. Let's explain what all of them are and how you should use them.

Welcome video - This is my quick personal thank you note for you :)

Instructional video - You'll find the links to this video right before we jump in to the Design Manual's practical part. I go over the file setup, and explain how you should replicate the designs yourself.

Fintech UI Kit - This file contains all the unedited Fintech App screens and is not meant for practice. Do with it whatever you want, use the app screens as a reference for your own work, get inspired or modify them and create your own awesome designs!

- **Design system** - A simplified version of the design system used for the Fintech App.
- **Components library** - Dynamic components used in the UI Kit

Fintech UI Kit Pratice Files - that's the file we created for you to practice your designs. Each Fintech App screen has an empty frame next to it. This is where you should work on replicating, tracing, or copying the UI kit. So far, 15 chapters have been explained, and you should be able to progress through them just by following the Design Manual part of this e-book. The remaining 9 flows are coming soon!

UX/UI Design framework - This file contains all templates we use for our UX and UI design process, from user journeys, IA, and user flows through mood boards to style guides, and brand guidelines.

Font pack - Font installation files we used for the project.

Hand-off documentation (coming next week!) - All you need for a successful design hand-off. We'll cover this part in later chapters.

How to design better - My free design e-book as a freebie. You might have read it or not, but hey, it's free, so I wanted to make sure you get it too :)

Dribbble & Instagram presentation mockups - simply upload your beautiful shots on a frame and create amazing compositions. These are attached to the main Fintech UI Kit.fig file as a BONUS

The resource folder will grow as we release more updates and cover more flows. Make sure you get familiar with all files, and I hope you'll make good use of them!

I think that covers everything we needed to mention before jumping into our first theory chapter. It's time to dive deep into the vast and still uncovered world of User Interface and User Experience design.

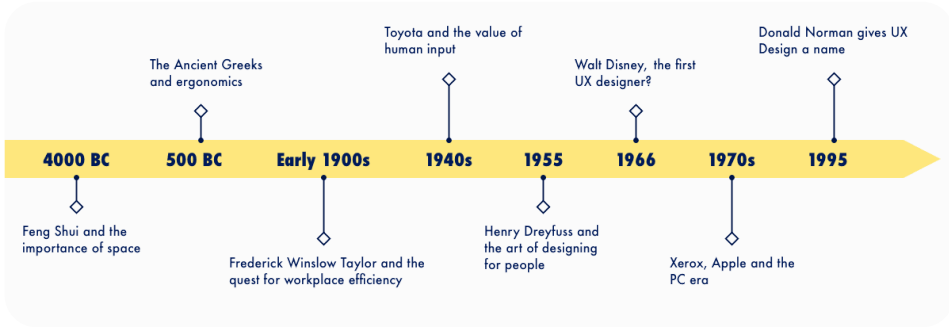
Follow me to the next page if you're ready!

UI vs UX

2. A brief history of UX

To get the whole picture about the design niche and UX in general, let's travel back in time to its origins and quickly jump through the history of how user experience came to fruition.

At the beginning of the XIX century, User Experience started emerging as a separate branch of the product industry. In the 1940s, Toyota developed a sociotechnical system called Toyota Production System, which was known as the first-ever production system centered around the human. It focused on the interaction between humans and technology. Toyota's philosophy was based on constant development, improvement, decreasing waste, and empathizing with respect toward their employees. It's where the UX started to become a thing.



Source: thenextweb.com

First examples of UX go as far as 4000 BC to the beginning of Feng Shui. Both UX, and Feng Shui aim to improve people's lives through design and require good understanding of human behavior.

Fast-forward a few decades, we're jumping to the beginning of the 1980s, when the Human-Computer Integration (HCI) was established as a discipline concentrated around IT and human factor engineering. Yeah, that's a lot of industry-specific jargon. To put it simply, HCI researchers observed and still observe how humans interact with computers and design technologies that allow humans to interact with computers in new, different ways. It wasn't until the 90s, when a psychologist named Donald Norman joined the team in Apple as an Advanced Technology Group Director and formulated the term "User Experience" in the same sense as it is known today. Over the past 30 years, it has evolved greatly and become the foundation for designing any product in any industry.

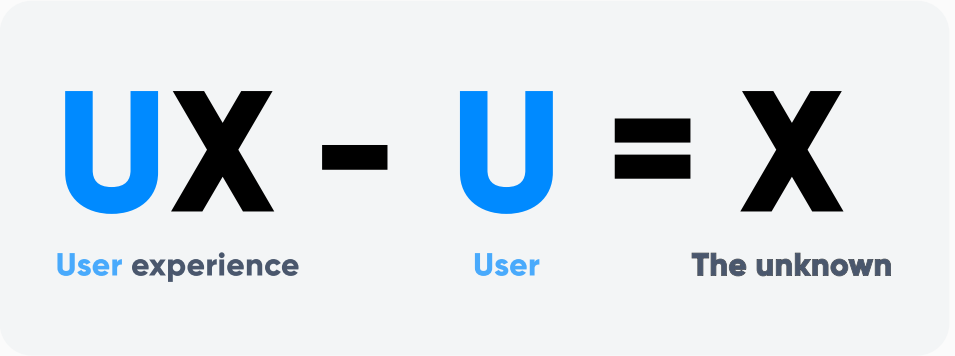
UI is not UX, UX is not UI

This brings us to the modern days, where UX and UI are often mistaken and considered synonyms, which is frustrating for UX and UI designers. Both UI and UX concentrate on usability and improving a user's experience, but they do it in their own exciting ways.

UI focuses more on the visual part of the project and its brand perception and creates the visual impact that makes the app pleasant to look at, accessible and easy to use by all users. UX has a more analytical approach and covers the very essence of the product - what it does, how it works, how it solves the user's problem, and how it benefits the owner(s) of a product.

The UX without the U

When I started out, I just couldn't find much value in the UX process. Like many beginner designers, I jumped feet first into high-fidelity designs, forgetting all about the research part. I mean actual research, not just looking at what competitors designed and borrowing design patterns that looked and felt good.



And if this is, or it was you, it's not just our fault that we used to skip this part. UX is not that popular among small businesses where budgets are minimal, and there's no room for proper analysis and research. You'll understand that UX is at the heart of every successful product when you start progressing through the ranks and begin working with better and more financially sound businesses.

User Experience goes much deeper than the visuals. It explores and recognizes customer needs that allow companies to stand out from competitors, gain trust, and build powerful brand loyalty. It's crucial for building successful products.

UI designers - what do they do?

They are very detailed and thorough with their application of design patterns that align with the all accessibility standards to create functional products, pleasant to look at and easy to use. They are a creative bunch; they can be called visioners - they are, in fact, trendsetters that are constantly looking for the next big thing. That's why Apple is so beloved by designers nowadays, transforming the industry with their use of vibrant colors, blurred backgrounds, slick interfaces, and very dynamic and fluid animations that stood out against flat, monotone standard interfaces of the past. Using the collected data and principles of design, UI designers create digital products such as websites, mobile apps, and all kinds of interfaces.

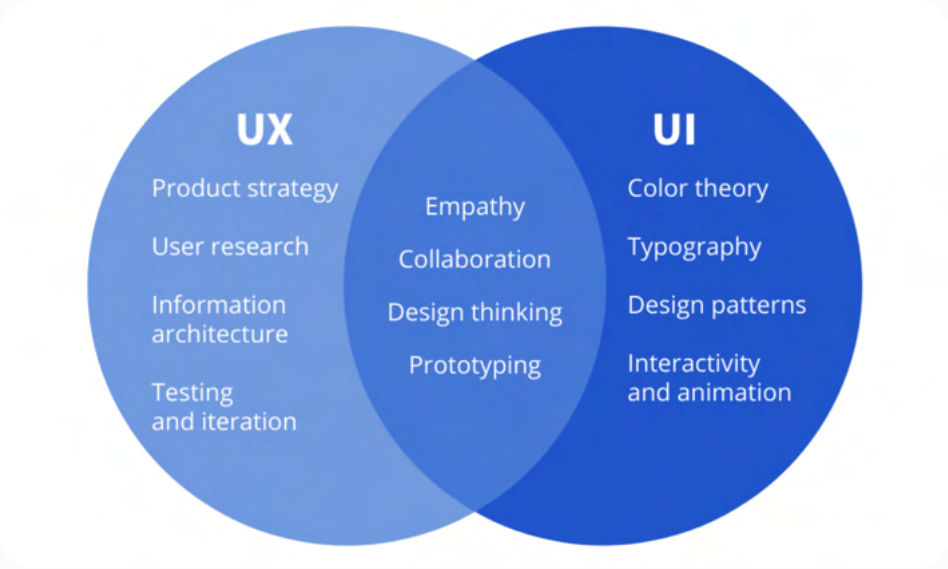
UI in UI designer stands for User Interface, which is all about designing the product's appeal.

A user interface is like a joke. If you have to explain it, it's not that good.

Martin LeBlac

User Experience design, on the other hand, can be described as designing the visitors' experience (hence the name!). It focuses on providing the user with the most pleasant experience in contact with the product because, let's face it, how the users perceive a product the outcome will dictate how successful it will be.

UX designers use analytical tools and heuristic evaluations to focus on what the user sees, hears, feels, and thinks. User Experience is honesty, emotions, and genuine feelings of real people who will use the product or are already its users. The task of a UX designer is to learn as much as possible about a person in the context of a product to help improve the overall experience.



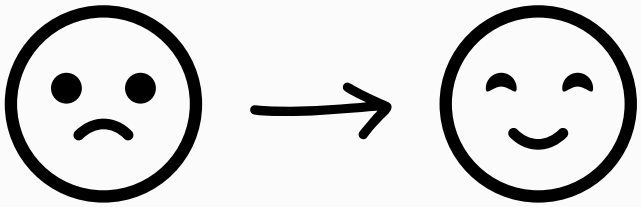
Credit: Coursera

UI designer, in turn, deals with everything related to visuals because UI (user interface) design is designing the experience on the surface. What's at play here is the typography, hierarchy of content, colors, appropriate spacing between elements and photos, consistency in design, recognizable design patterns, movement, and animation.

UX designer creates the architecture of a product, wireframes, and prototypes and proposes solutions, conducts research with users, and creates personas and user journey maps. There's no denying that they're the real O.G.s here; they play the first orchestra, an executive role balancing on the border of the business (very often their own sanity) and product user market.

The role of a UI designer is no less critical; it's very much based on data, but not as much as in the case of a UX designer. Every UI designer needs to know how people read, how they scan content, what are the accessibility principles for people with disabilities, and so on, and so forth.

User Experience is collectively a variety of impressions, feelings, and reactions experienced by a person who comes in contact with a product. From the first interaction with a digital product, there are feelings related to its use, they may be positive or negative, and **UX designers' job is to always make sure they are positive.**



poor UX **good UX**

UI vs UX described with a meme - the Heinz ketchup bottle

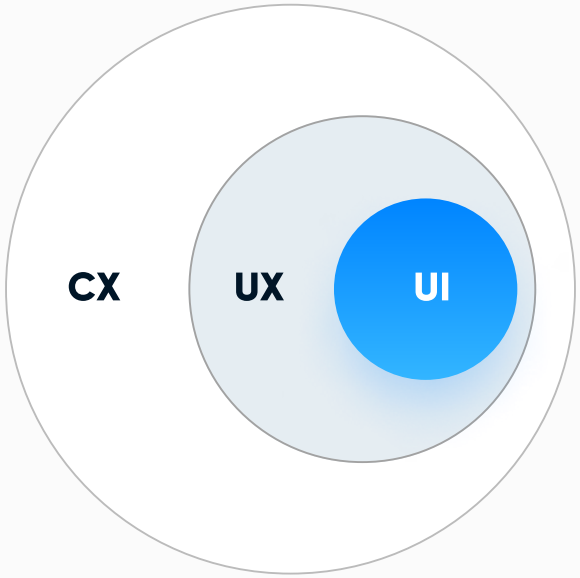


Illustration by Yannis Abelas

The Heinz glass bottle example goes deeper than just a simple UI vs. UX meme. The company decided to transition into a more accessible bottle design when they discovered that a typical 5-year-old consumes about 60% more ketchup than a typical 40-year-old. When they realized the bottle design wasn't the best solution for children, they focused on solving a problem and improving their product's usability. Giving your kids a knife to soak it into the bottle or bang at the end of it to spurt out the ketchup didn't seem like a perfect solution, so they came up with an EZ squirt bottle made out of soft plastic and a nozzle. Much lighter to hold and much easier to operate, it turned out to be a hit, and the total Heinz ketchup consumption grew by over 12 percent. The glass bottle didn't disappear forever, though; it returned a couple of years later as a nostalgic limited-edition you might get in your local stores.

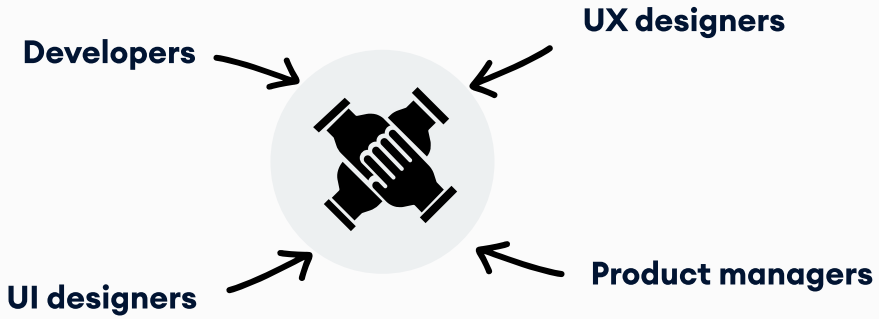
Where do UI and UX fit in all of this?

UI is a part of the UX process. UX process, however, is part of a much more significant element - Customer Experience. The CX involves UX, but not only, but it also requires branding, marketing, customer support, and how your entire company operates. The diagram below shows the relation between these elements.



If you're working or planning on joining a small agency, there's a big chance that you will also be carrying the burden of a UX designer as a UI designer. In more prominent agencies, though, the roles are more divided. UX designers focus solely on the UX process, while UI designers focus on the UI. There are also other design-related positions as design goes even deeper than interface and experience.

We're all in this together, so make sure you always cooperate with your team members and take some time to understand their work process. It's only beneficial for you to improve your understanding of all the product design stages to be a better designer.



Who knows, maybe at some point in time, you decide that you want to transition to UX or UI, or even better, become a well-equipped full-stack wireframes-slinging, a design-system-shooting product designer on a hiring list of all the popular tech startups on the market.

I hope this gave you more clarity on the UX and UI and the core differences between the two. Now that you're equipped with this knowledge let's discuss the key positions and their responsibilities in a standard design agency or a bigger organization with a proper, divided, creative department.

Designer roles

3. Designer roles

You probably know this by now, the design niche is enormous. There are so many things you could do and I, for one, had no clue where to start. I didn't know where to begin or find information about who does what in a design agency, where should I focus my energy the most, or if I should just learn everything at once and become a jack-of-all-trades but a master of none? I kept asking myself, who's a product designer and a product owner? Who is a Scrum Master, and why does he sound like a class from an MMORPG game?

When I started out in design, I was confused with all the names and positions the design industry offers. In the following chapter, I'd like to show you the key responsibilities of each designer position and the role they play in a typical design agency. Maybe this will give you some more clarity and help you decide on a branch of design you'd like to focus on later in your journey. Without any further ado, here are the most common designer roles:

UX Designer

The primary responsibilities of a UX designer are creating wireframes and prototypes, conducting research and interviews, creating user personas, user journey maps, information architecture, and designing every experience related to the core functionalities of a product.

UI Designer

UI designers deal with the visual design of the components and elements, applying styles, building systems and component libraries. Usually, UI designers create interfaces, high-fidelity wireframes, prototypes, style guides, and design systems.

Product Designer

The leading roles and responsibilities of Product Designers usually involve both UX designers' and UI designers' work. A Product Designer must demonstrate knowledge in developing and creating a product, including the research and design phase.

Product Designers participate in every stage of a product design work. They also balance the users' needs and business goals, ensuring everyone included in the design process is happy.

Motion designer

Motion designers focus on animations and micro-interactions. They’re responsible for creating smooth transitions between elements on an app or a website. Motion designers may create animations and apply visual effects to bring their design ideas to life.

They create intricate prototypes that capture the micro-interactions and animations needed for the end-product development. They are usually highly specialized UX designers.

Interaction Designer (IXD)

The primary responsibility of an interaction designer is to make sure that designed products function properly when users interact with them.

They're responsible for fully understanding users' goals when interacting with a product, application, or website, compiling user, business, and technical requirements, and creating logical product information architecture and interfaces for users to navigate.

Scrum Master

Scrum master works with the Product Owner and the Scrum Team to refine and improve processes. They make sure that team members are well-trained in Agile methodologies.

They host daily stand-up meetings that last no longer than 15 minutes and allow each team member to share their progress on the assigned tasks. They keep every design task in check.

Product owner

A Product Owner is a part of the scrum team. The key responsibilities of a Product Owner are to define user stories and create and maintain the backlog of all tasks for the current product.

The Product Owner is the primary point of contact on behalf of the customer to identify the product requirements for the development or design teams working on the product.

UX Researcher

UX researchers conduct in-depth research on target consumers in order to collect and evaluate data that will help in the product design process. UX researchers generally work with two types of research, qualitative (how users feel while using a product, or what dificutlies they have in completing the step) and quantitative (numbers and statistics).

They conduct user interviews, create and analyze studies, and carry out usability testing to collect and evaluate data.

Web designer

The primary skills and requirements for this role are the basic knowledge of CSS, JavaScript, HTML, and web design principles.

Web designers design pixel-perfect websites that are documented and delivered to developers. They might also create them themselves using no-code platforms like Webflow and Elementor.

UX Copywriter

Their primary roles are analyzing data and using the cognitive psychology of humans. UX copywriters must empathize with the user and his way of thinking to create the content of a digital product in an easy to consume, intuitive way. UX copywriters concentrate on writing clear, short text that can assist users in making the best decision possible, which is business-focused.

They are highly specialized copywriters that have a deeper understanding of users' psychology and their needs.

Good vs Bad in UI

4. Good vs bad designer

Good design is not just about good looks. What's happening behind the scenes is what makes or breaks the product. A good designer strives for good design; that's pretty self-explanatory.

How can one define a good designer? What knowledge should he acquire? What kind of skills? Or perhaps this is all about having a (super)natural design tingle, and the rest is screwed?

What do you need to be a good designer?

For one, you don't need a degree. Even though it definitely helps, countless self-taught designers are very successful in this industry without having a degree. You can participate in different courses online, hire a mentor, or join a private Bootcamp but remember, there are many skills you can acquire on your own, and you don't need any design school to tell you that. However, there comes a time when you might need the help of an expert to push you to that next level, and you need to be very cautious about where you get your information from. If you're learning from the internet, remember that there are things that people might not tell you about unless you take part in their expensive course, or if you're lucky (and you are!), an inexpensive course that will show you what's happening behind the curtain.

Don't get me wrong, I learned a lot for free, and internet has been a real lifesaver for so many of my projects. I was never fulfilled, though.

I took part in dozens of paid courses, read tons of books, and even got my master's degree in design to "understand" design. And while all of the above was crucial in forming my understanding of UX and UI and got me to where I am right now, I wouldn't say you have to follow the same path I took. What you should do instead, is you need to practice your skills all the time. It's a never-ending process, and nothing will replace that. As the saying goes - practice makes perfect, and you must practice a lot. You're asking me on IG for tips on how to learn UI design fast. Well..there are no real shortcuts to learn design without really understanding the basics. Some people consume the knowlegde faster, some slower, there is no trick that will make you a great designer in a week or a month. Sometimes it takes years. But there's one thing that'll help you get there much faster. Don't fall into the trap of designing super visual concepts that are not usable. Even though you'll get tons of likes on Dribbble or IG from inexperienced designers, this won't bring you the proper attention of real clients. The reality is relatively simple. Look at the most famous shots on Dribbble, then grab your phone and open your favorite apps. Do they use the same design patterns? Sure, you'll find resemblances for some designs, but if you dig deeper, you'll quickly find out that most popular apps are looking great, sure, but without being too flashy or overly creative. They have the right amount of balance. Why is that? Why aren't they more hip and trendy? Think about it for a second, and we'll circle back to that in a bit.

Let's ask this again, do you need natural talent to be a good designer?

A question that can be answered both with a yes, and a no. If you want to become a designer (UX or UI), you don't necessarily need any natural talent. Of course, it helps to learn faster and recognize design patterns when the rest is still chasing the basics. Really though, the secret is in creating designs, replicating them, analyzing them, iterating, and improving until you're at a point where you can think about specific design patterns and recall them from memory without looking at any references. The key is to consciously look at ways to design and understand them. How often have you looked at an app and didn't even notice how it's designed? What font sizes does Uber use, and why is it so easy to navigate? Why can you just mindlessly swipe through your Instagram without thinking about it?

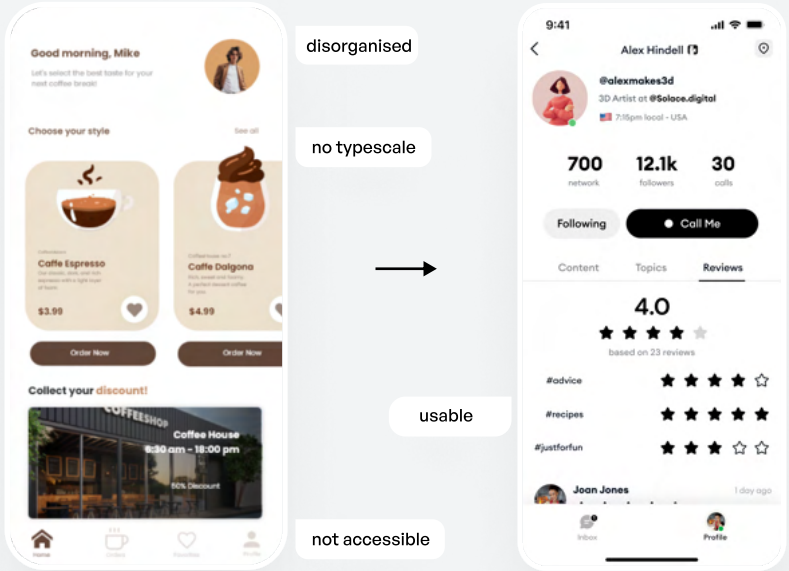
It's because good design is invisible.

Now you need to understand what makes it that way. Do you see Airbnb launching its new app update with a fresh redesign? Download it and play around with it. See what you like and don't like, take a few screenshots, replicate what they did, analyze it and make conclusions. It all comes down to that very detail.

Practice makes perfect, and talent, talent won't replace hard work.

I know this sounds like some cliché motivational speech, but it's true. By practicing, you can work on your weak points and improve areas that need improvement.

Even the best, super-talented designers started somewhere. They all had their share of bad, clunky projects that they would laugh at nowadays. You don't even have to look that far back; even when looking at my work from 2 years ago, I often ask myself how I could have been proud of this design?



You'll keep improving your skills and notice how imperfect your designs were and how much you've improved over the years.

What defines a good designer?

There are certain traits that define good designers. Let's look at them.

Innovative approach

A designer should look for new things, be up-to-date with trends and technologies, be eager to learn and search for the new, stay open-minded, and be able to think quickly and on the go.

Problem-solving capabilities

Remember that any digital product has a goal, whether it's solving a problem, giving information, or providing any sort of utility to a user. Creating products that don't have an end goal means creating a product that's not useful for anyone, which misses the whole point of the design. Think of all the flashy, beautiful designs with no real purpose in solving user problems. You don't want that.

A good sense of aesthethics

It is crucial to have a sense of aesthetics in design. A good design is simple, clean, and at the same time, doesn't require much thought to process. Remember our last quote? Good design is invisible, and you probably can attest to that when using your everyday apps.

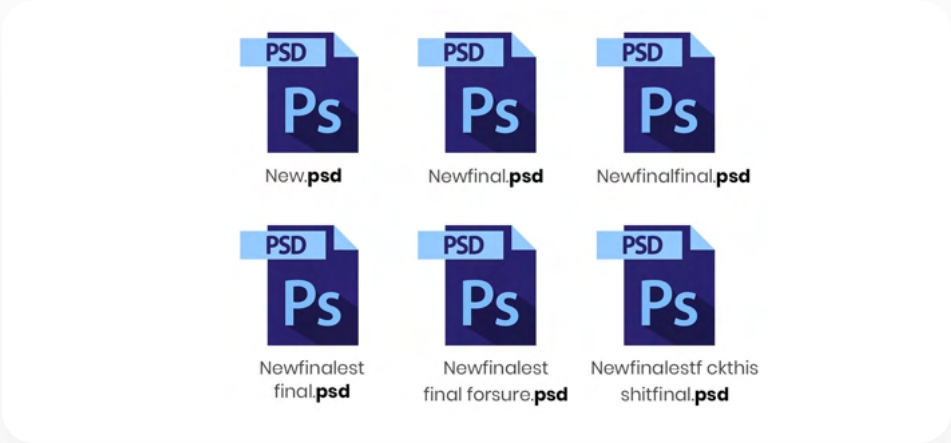
It's like you don't even notice the design at all.

Attention to detail

When every element is refined and carefully planned, the design becomes aesthetic and remains consistent. The detail makes the design beautiful and eye-catching and brings it to life; the consistency in design strengthens the user's understanding of the product.

Meticulously organised

It’s not only about keeping your desk uncluttered but, most importantly, your design files. Name your pages accordingly, create a file structure, date it, name your layers, organize your project files, and keep the project documentation and briefs in their respective folders. We all know this meme about naming project files in Photoshop. With Figma, we don’t have to save them anymore, but I’m sure some of you remember those good ol’ PSD crashing days 😂



Source: farasnear.com

Empathetic

Putting the very user in the center is essential to designing a product dedicated to a specific audience. Without knowing the users' needs, problems, and goals, we are designing for everyone and no one at the same time.

A good designer knows this and tries to get into the person's shoes. He's trying to solve the problem. This requires a lot of research and analysis, which, in any case, shouldn't be skipped.



Source: agentestudio.com

As designers, we solve user problems and improve their lives by making processes more accessible and simply better. Empathy toward users is one of the most essential traits for a designer.

How to understand design

How to understand design

UI design isn't only about making things look good. While this may be true to some extent, there's more meaning to it. Remember each time Facebook dropped a new redesign? People went nuts over the changes and didn't like them at all, even though we got to see an improved, better version of the interface every time. The design has to be invisible otherwise you'll get in trouble.



Source: memegenerator

People get used to how things look and making drastic changes can only annoy regular users. We, as designers, appreciate it a lot when our favorite app gets a refresh, but that's not the case with everyone.

Remember, people won't notice if your design is excellent, but they will be the first to point out when it's terrible. You have to find the right balance and understand when enough creativity is enough.

How can you know it? Well, first, you have to understand the basics.

Know the basic principles of UI design

Contrary to popular belief, designing an interface isn't just "painting with pixels." Design is not art. It follows (or at least should follow) very strict and precise rules to create a visually stunning product that's consistent and speaks to the genuine brand's values.

You need a solid blueprint to build a beautiful house. Otherwise, you're left with many poorly assembled materials and components that were patched up on the go without much thought and now are about to fall apart. We don't want such chaos in our work.

There are certain principles, constraints, and regulations when designing for specific devices, and we need to adhere to them to create products that are easy to use and look at.



Even a simple button has to be designed according to specific rules.

Know the system you're designing for

Android design is much different from iOS, and mobile design is an entirely different story than website design. You need to know the system you're designing for in and out, and most importantly, you need to understand the core philosophies that guide them.

We'll cover the entirety of iOS design in the following chapters, so let's circle back to that.

Know your design tools

To design efficiently and make the most out of your designs, you must first master the tools. Get better at Figma, understand the basics, memorize the keyboard shortcuts, and get to know auto layout, layer-naming, and working with constraints. You'll feel so much more confident in your skills, and your work efficiency will skyrocket if you first take the time to understand your design tools.

We will cover many tips, tricks, and shortcuts you can use in Figma to speed up your workflow in our Theory chapter and put them into good use in the Practical segment of this e-book.

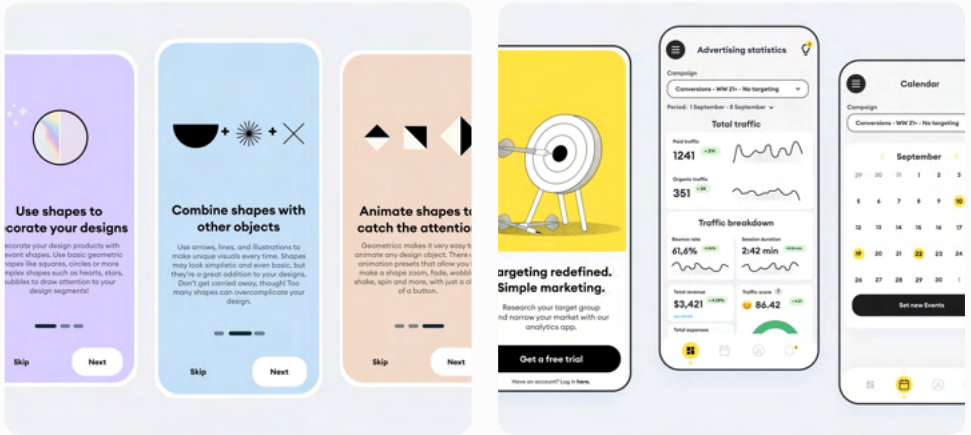


What makes design good

What makes design good?

Best designs are invisible - I can't overstate this. As UI designers, we aim to create simple and straightforward interfaces for users to get things done quickly and effortlessly.

Your content should be the hero of your UI; everything else is secondary and should be used to support your content. Your interface should be stripped down to the core aesthetic. Whenever you consider adding a new element to your UI, ask yourself a question - is this really necessary? Try to steer away from heavy gradients, textures, and layering shadows. Focus on the very basics and try to improve from there. Let's take a look at the following example.



These are one of my top-performing shots on Dribbble. Are they better than the rest? Not really, one is just a walkthrough flow, and the second is a bunch of patched-up widgets and components I created for a different product. Is it a bad design? Again, not really.

I guess they are okayish, but why did they reach over 200k users and generate close to 300 likes? It's because they stand out, are colorful, and are fun. The presentation is what makes them successful.

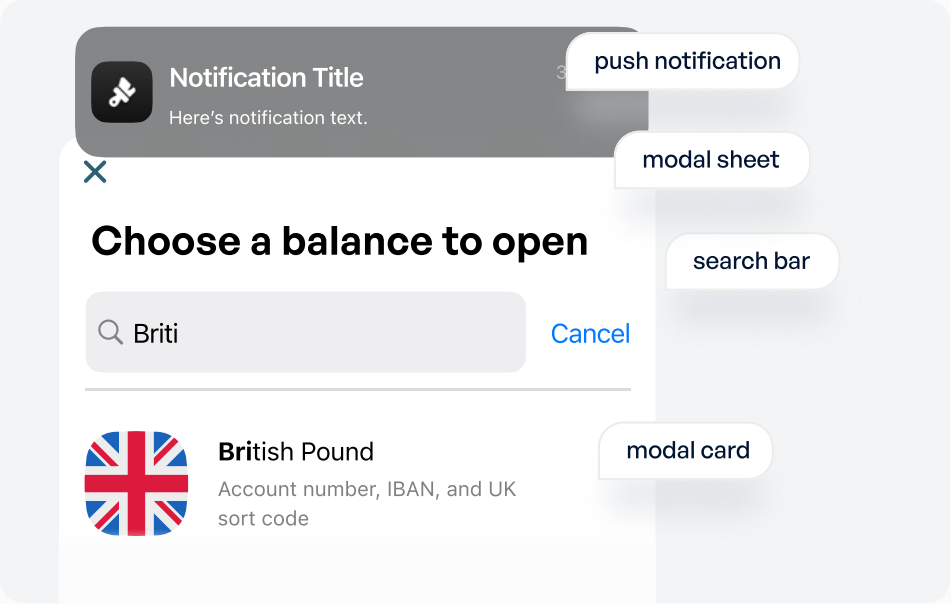
Nothing else. Good design very often doesn't stand out, and that's normal. But hey! Shouldn't likes on Instagram or Dribbble measure how good or bad a design is? That's the biggest misconception since the creation of social media. Quantity never goes on par with quality.

So how can you tell if a design is good or bad?

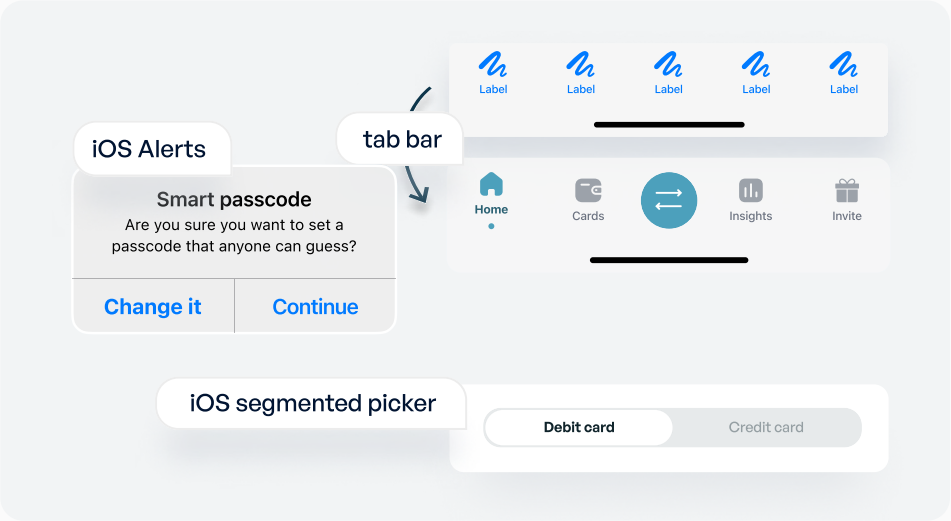
Maybe let's drop the word "bad" and use "incomplete" instead; we don't want to offend anyone.

Good design is simple. A screen is more powerful and valuable if more is said with less. Every element of the design must support the purpose of the app. If you remove all the noise and clutter from your designs, create more space, and make your content the focal point of your app, you're halfway there.

We're going to practice what we preach here and cover so much more about the good and the bad in iOS design in the following chapters, so keep reading!



We'll cover almost every iOS element you need to design and show you how to implement them into a real app. The right way.

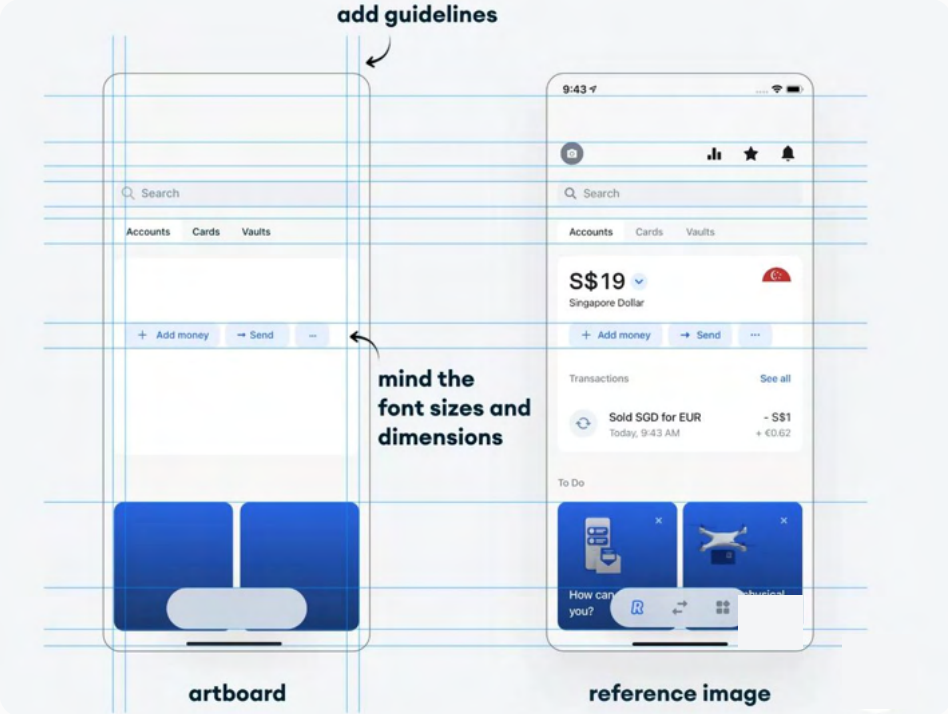


Get better at design

*fast

5. How to get better at design fast

Many designers feel like it's unacceptable to copy someone else's work. It seems as if you need to be able to conjure design ideas out of thin air, or else you're not a real designer. I will tell you otherwise. Copying is not bad at all. It can actually do wonders for you!



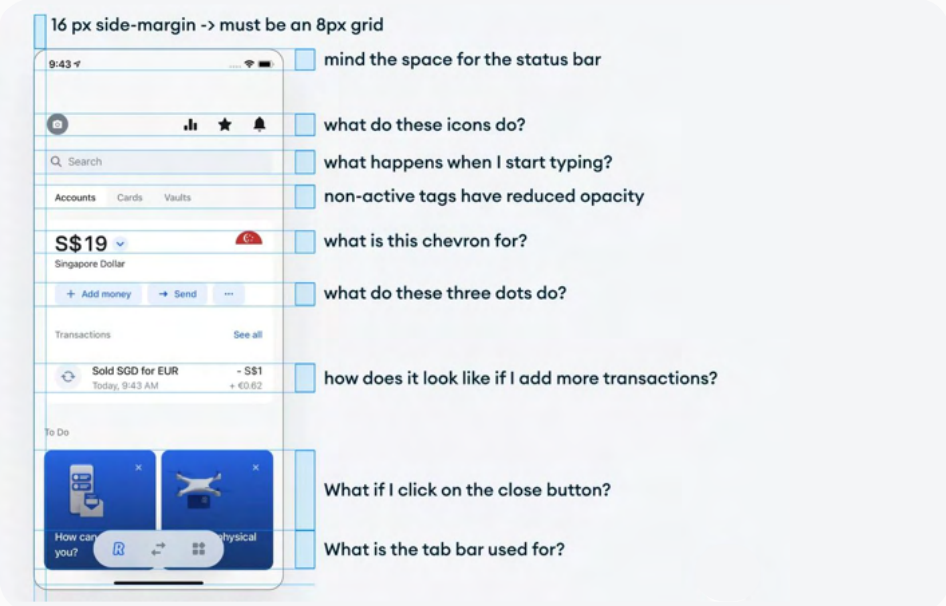
Stealing is wrong, though, don't steal whole concepts. Borrow ideas, copy what works, and analyze it. Taking someone else's work and claiming it as your own; that's terrible. But copying, copying is, in my humble opinion, the greatest way to truly understand a design, or an app, or software application, and it's the quickest way to learn how it was designed and why it may have been designed this way.

Everyone starts somewhere, and if you want to be the best at what you do, you simply start by copying the best.

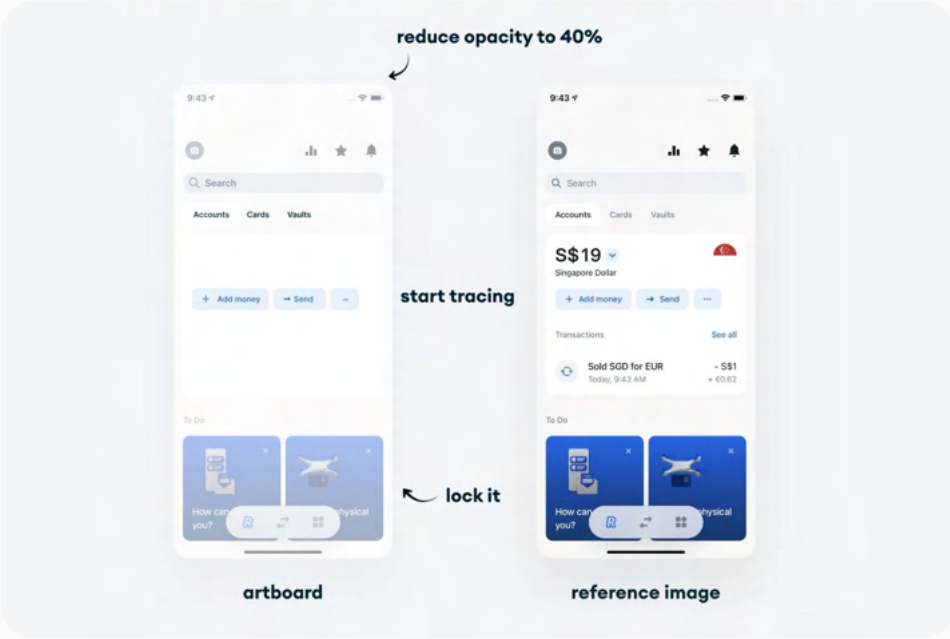
This, my friend, is the simplest and fastest way for you to learn design. Let's talk about why this is so effective.

Why copying work is so effective?

You have to consciously think about what you are copying in such detail to replicate it. By profoundly analyzing other designs, you will begin to understand how interfaces are constructed. You will start to recognize design patterns, font sizes, spacing between elements, margins, and the use of color, but what's most important, you have a chance to get into the head of a designer who created the interface to analyze his decisions and learn from them.



You get to enjoy the best of UI design by diving straight into visual design. Every design decision has been planned out, tested, and improved for you, so you can just enjoy the craft now. You no longer have to stare at your empty frame looking for design inspiration for your next concept for practice, unsure where to even start, and never creating your design in the end.



If you are new to design, copying the interface of existing apps can help you take your designs to another level, visually and literally. **You get to learn new visual patterns and start recognizing them**, able to think of new solutions on the spot. If you trace 5 designs of a few popular apps in each niche, you'll improve your pattern recognition by a mile. It's that powerful, trust me.

Where to find apps to replicate?

Now, this is a tricky question. Every day, we are overwhelmed with thousands of different designs on many creative platforms. These are mostly designs made by beginners and practicing designers, but if you know where to look, you'll find some great works by amazing creators showcasing beautiful concepts and commercial work.

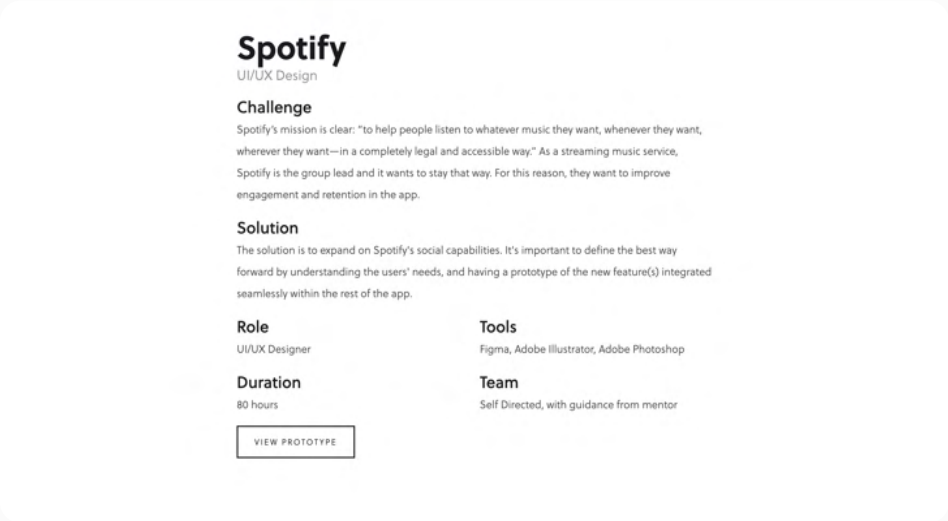
For the most part, they are lovely eye-catching candies that can serve as a great source of inspiration but are not always the best example to take as direct inspiration to practice your design skills. There are many beautiful-looking designs that, at first glance, look amazing, but if you inspect them closely, you quickly find that they don't even meet the basic standards and principles of good design. The font sizes are all over the place, colors are inconsistent, tab bars have elements floating way above the safe space obscuring important sections of the interface, and the composition doesn't even make sense, logically. A design that would rapidly annoy real users.

Commercial work is different from conceptual fairy tales, and you need to understand the key differences.

There's one particular question I get a lot in my DMs - where do I look for good design inspiration? The answer is simple - look at the big players. One of the best free ways to find good design inspiration, both UX and UI, is to look at the case studies of well-known brands.

Case study websites

You can find tons of inspiring work in all niches, with real actionable insights and results documented during the entire design process. An excellent place to look is simply to type in Google - "(UX)UI Design Case study" and add the name of your favorite product or an app in whatever niche you're interested in designing for.

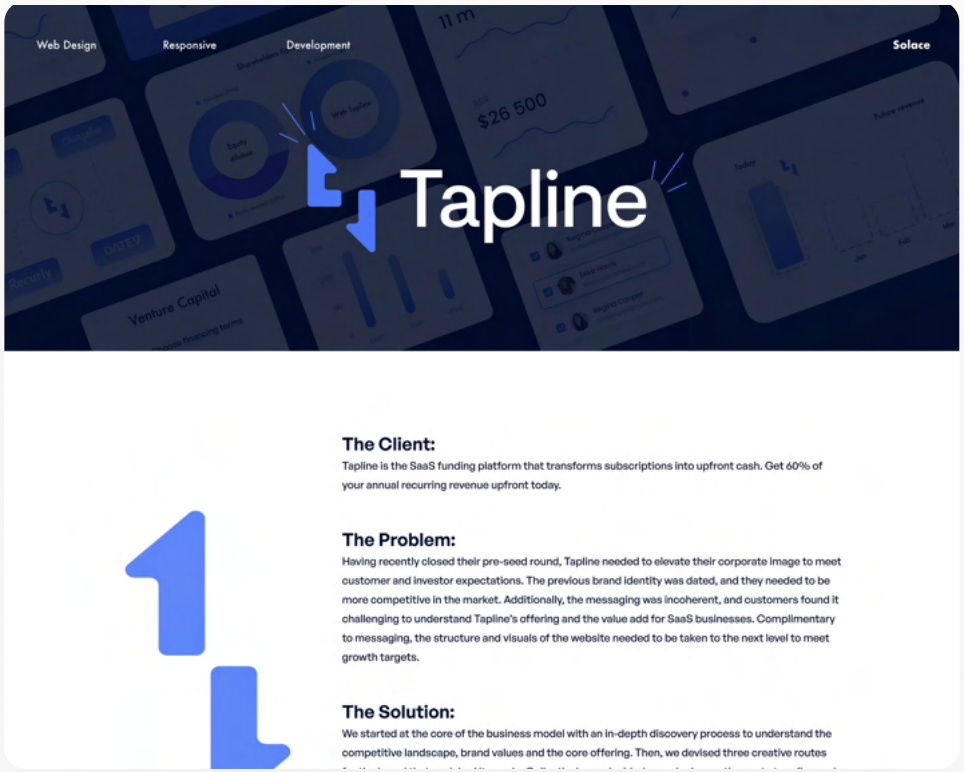


Source: Hiromejia.com

Behance & design agencies websites

Behance is also a great place to find detailed case studies with interesting information. Find the most popular design agencies in the industry and look up their Behance portfolio. They'll likely have a case study prepared for each of their previous client's products.

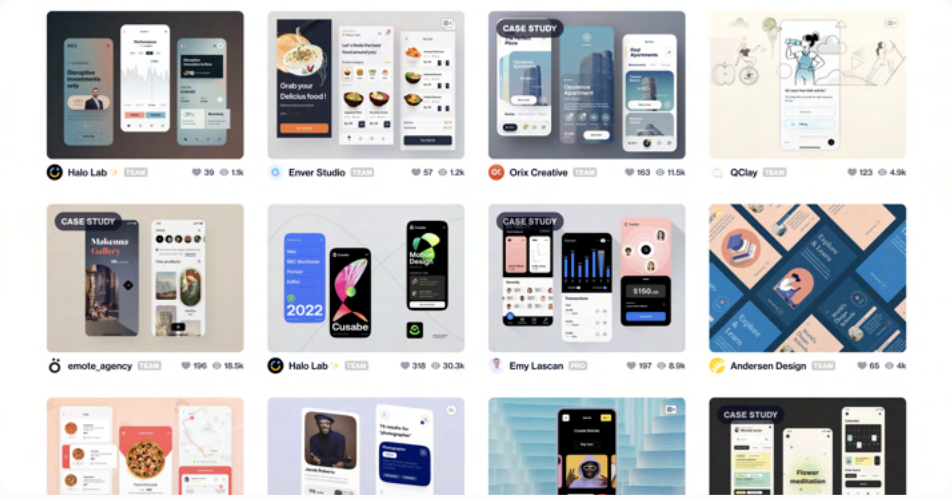
Many design studios share their most challenging case studies and projects with detailed explanation of what's been done and how it affects the company and its users. Look at their sites, too; companies share their work documentation, and show their entire process, from UX to low-fidelity mockups, wireframes, animation, and final live products. It's the process that attracts big clients. Design studios know that and they go over and beyond to show off their work in the best light possible. That's where you can learn from following their thought process and replicating their work for practice.



Creadit: Tapline Case study by Solace Digital

The truth about Dribbble

Dribbble is fantastic for all things visual. When looking for visual inspiration, it's my first place to go. It's literally all I need to create a mood board for any new product. But, when it comes to UX research and looking for real solutions to clients' problems, it might not be the best site to start your analysis.



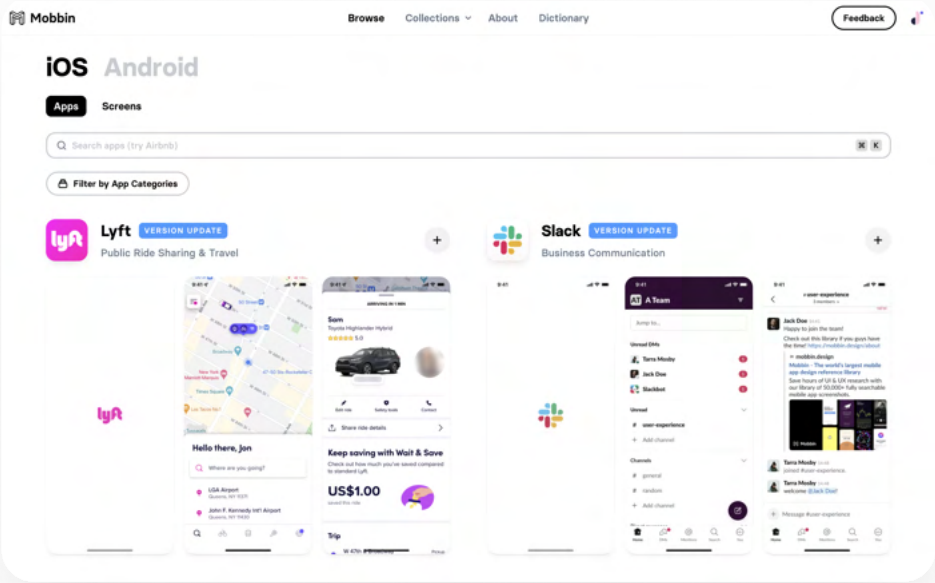
Source: Dribbble.com

I mean, you can find almost everything there. The problem is that Dribbble is fast-paced. You can't create a detailed case study on Dribbble, nor can you genuinely explain your process and ideas. You post one or two shots and move on. Plus, the algorithm on Dribbble really favors the "old" users, and the most active designers on the platform. Excellent work is often lost and never actually discovered. You have to really dig deep to find good examples, which doesn't mean it's not possible. It's just hard.

My top pick to find real inspiration

My personal best place to look for design inspiration with real practical solutions and logical interfaces is mobbin.com. You might have seen me recommending this platform many times before, but it's simply the best site for app design inspiration.

And I'm saying this without being affiliated with them at all. Mobbin has a massive collection of real apps released to the market, organized in segments and flows, so you can explore entire apps, filter your results, and find exactly what you need for your next project.



Source: mobbin.design

Get it as soon as you read this e-book because you still have 550 pages to go through and 100++ screens of an iOS Fintech App design to analyze with me, replicate, and understand :D

I'm holding back no secrets here, and you'll get to see the entire design framework I follow in coming up with new ideas and organizing my work so that I am as efficient with my designs as possible.

It's as simple as that. I don't want to be overly excited about this e-book, but I wish I had access to this kind of product back when I started. When I found out about directly tracing real apps, it was like a revelation. Why didn't I think about this before? You get to see the best of UI design, and you get to enjoy the process.

Practice makes progress. Each drawing one makes, each study one paints, is a step forward.

Vincent Van Gogh

Follow my steps replicating this app, and you'll get to understand all the most critical design practices like the proper use of grids, paddings, and margins, the size of display fonts, headings, callouts, body texts, use of colors, white space, iOS components, masking, designing for accessibility, and usability. It's all in there!

iOS design principles

6. Intro to iOS Design

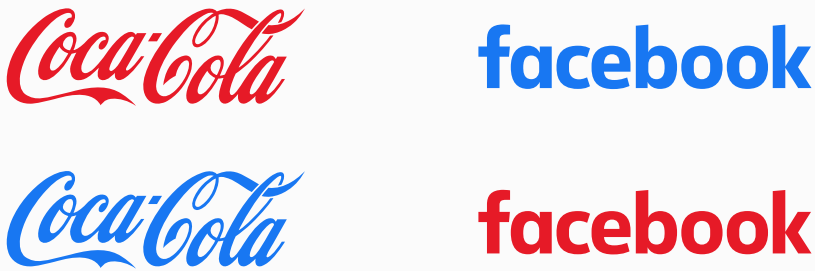
A few words before we dive into this chapter. I don’t want to focus too much on the design principles in this e-book because we’d have to spend the following 200 pages describing just the basics. Instead, we’ll focus on the best principles of iOS design, which will also talk about colors, typography, white space, grids, layouts, etc., applied to iOS specifically. There’s a big overlap between the two so you’re not really missing out on anything, and if you really want to understand the basics, check out my free e-book that’s included with this one.

Let’s talk about colors

Color psychology is the science of color influenced by emotions, cognition, and human behavior. Colors do not affect everyone equally. The perception of colors is determined by age, gender, and cultural environment, even something as unique as a stimulus associated with past experiences.

The color scheme of a website or an app makes it memorable, trustworthy, appealing, and profitable. It's all about making an excellent first impression. When creating a digital product, it's crucial to think about your color scheme. Which colors to pick from the color wheel and why. Different colors send different messages to the user, influencing their mood and perception of the product.

Some colors are so deeply engraved in a brand's history that they are sometimes as easily recognizable as the brand itself.



Imagine Coca-Cola being blue or Facebook being red. Feels weird, right? Color is an amazingly powerful tool for brand-building, storytelling, and conveying the right emotions.

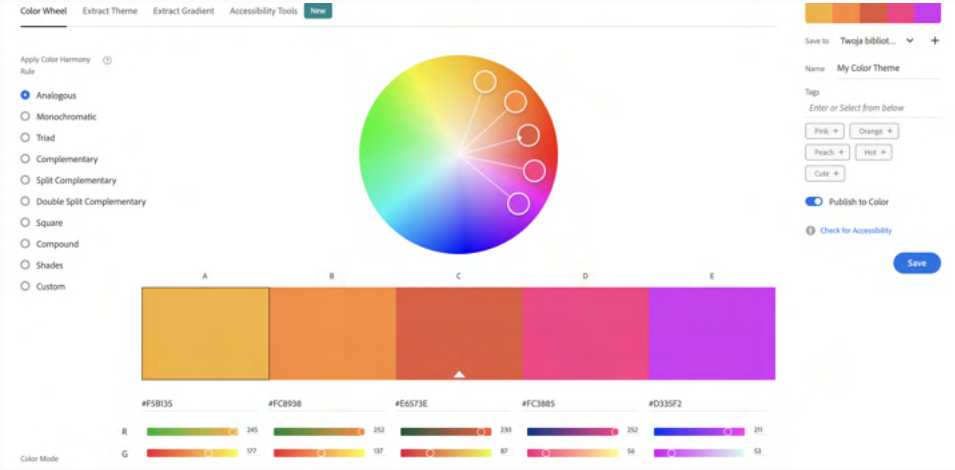
How to pair colors

As designers, we realize how difficult it can be to choose the appropriate colors and color combinations for our designs.

Here are the main guidelines for choosing your color schemes:

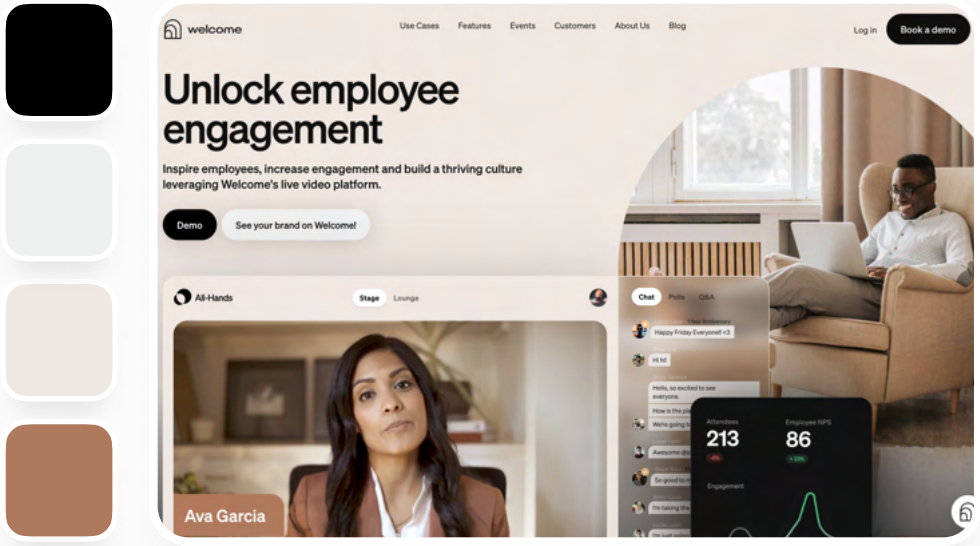
1. Find your primary color – App designers use different methods to pick the best color palette for their apps. The most used methods are the analogous method and the monochromatic method of color choice. Some do it by intuition or use color tools available on the web.

One of the most popular tools you can use to create your color palette manually is Adobe Colors. It's a bit more complicated and if you're a beginner I wouldn't worry about it for now. We'll cover this later.



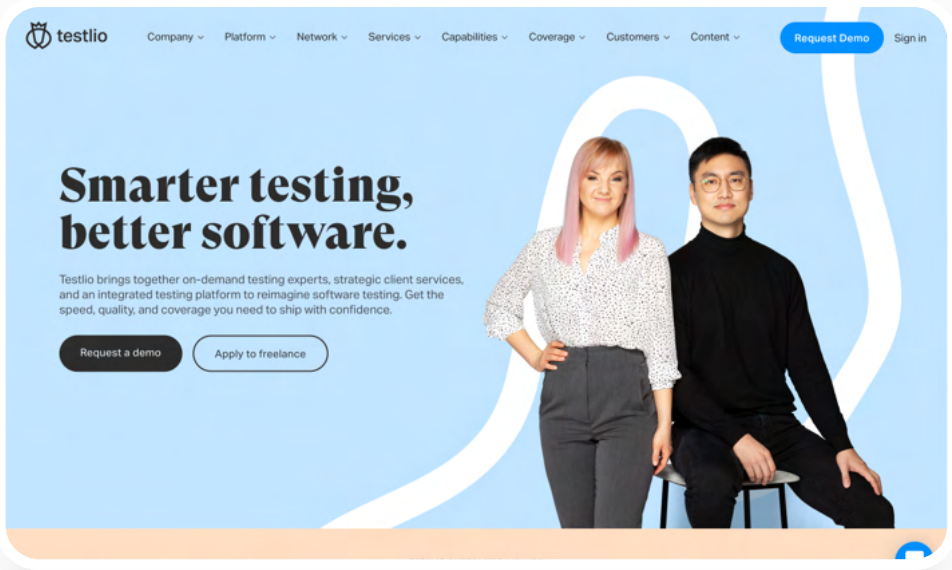
Adobe Colors

2. Decide on the number of colors you're going to use. It's best to have one primary, one secondary, and if you really need, one tertiary color with two neutrals - dark and light.



experiencewelcome.com

3. Use secondary colors when needed – underlining a secondary Call To Action or establishing a hierarchy between your content.

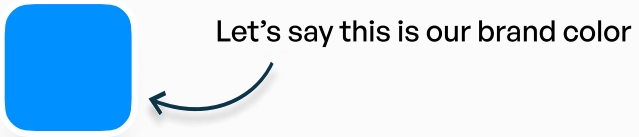


Testlio.com

Notice how Testlio uses its brand colors to establish a hierarchy between the Call-To-Action buttons. Can you tell which one of the three draws the most attention?

4. Use neutral colors – neutral colors are most often used for text to create contrast between the elements and make them stand out. They can also be used as background colors. The neutral colors are white, black, and gray, with all their shades and tints.

You shouldn't use pure black or gray, though. There's a little trick to make your designs more attractive and add more of your brand personality. The key is to slightly tint your grays and blacks. Let's say our brand's color is blue. Look at the example below.



How bland and boring would your grays look if you went just plain black. Also, blue feels a bit out of place next to pure black.

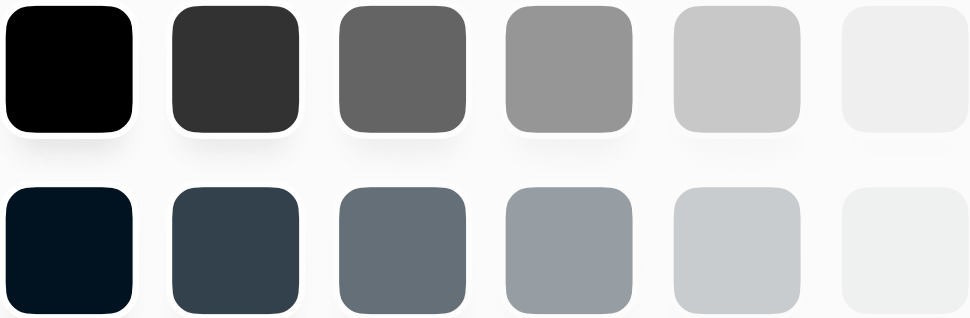


Now, let's use our brand color to tint our black. Open up your color picker settings in Figma and drag the pointer to the bottom-right corner of the screen and stop around 80-90% from the pure black





Can you tell the difference? That’s a tiny change, but it already feels much better. Besides, avoiding pure black also reduces eye strain.



We’ll talk more about color psychology and how we chose our color palette for the Fintech app in the Practical part of this ebook.

iOS Color palette

We can learn a lot about colors by looking at how other apps use them. There are specific colors used by Apple in their native apps, and they all have a good reason behind it. Blue is used for buttons, icons, and actionable elements across the system.

Other colors, such as yellow, are used for notes, red for Apple Music, green for Messages, ocean blue for iMovie, etc. Different colors can be used to establish the brand. If you're unsure about what color should you use, just go with blue, it's the safest bet.

While designing, certain colors are assigned to specific actions; red is used to present errors, and green is used to show progress or a successfully completed task.



However, it’s important not to rely solely on color to differentiate between actions or communicate important information. Apart from color, we should provide text labels or glyph shapes so users with color blindness or other visual disabilities can understand it.

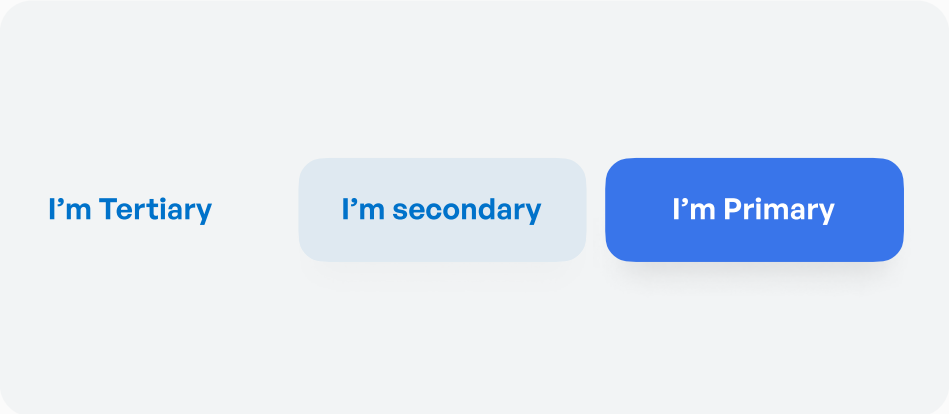
Consider choosing a limited color palette that coordinates with your app logo. Subtle use of color is a great way to communicate your brand language.

Dynamic System Colors

In addition to system colors, iOS provides access to several defined dynamic system colors that automatically adapt to both light and dark modes. These can be divided into two sets of background colors - system and grouped - each having primary, secondary, and tertiary alternatives. They have to establish hierarchy through the color of links, labels, placeholder text, separators, etc. A more comprehensive guide can be found in the Human Interface Guidelines documentation, which I highly suggest you study and analyze.

Neutral tones

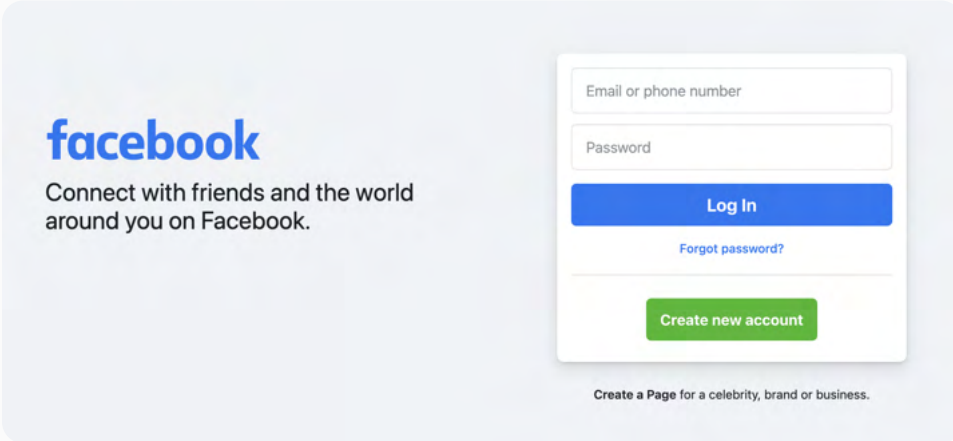
In iOS design, neutral tones act as a supporting character. These colors are usually made for secondary actions and help establish a hierarchy between buttons or actions. They help balance your design, are easy on the eyes, and bring more attention to the content. These could be the following.



Contrast

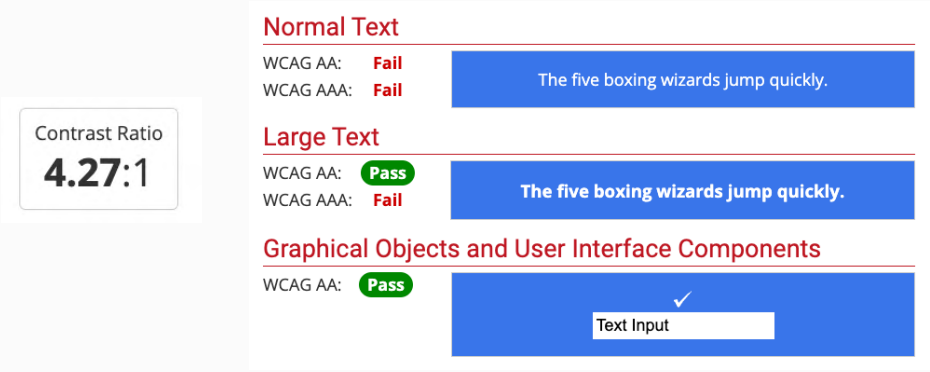
When talking about colors, we need to mention contrast. It's one of the most important factors when it comes to accessibility. Your colors should allow for your content to be legible. For the best possible contrast, try to go on the opposite ends of the color spectrum, white against dark shade, light blue against darker blue, light green against dark green, etc. Use high brightness against low.

The best way to measure your contrast values and see if your content is legible is to use the WCAG contrast checker tool. Put in your background and foreground color values and check them against the double-A and triple-A standards. Your text will rarely pass both tests. That's not always the case with the GUI components (buttons, links, icons). Look at Facebook's example.



facebook.com

Facebook have recently changed the shade of their blue to a much lighter and saturated version of blue. The hex code I picked from the button is #3975EA. Let's test it against the #FFFFFF foreground color used for the "Log In" text.



Even though they wouldn't have passed the standard and large text test, they would do just fine with interface elements. As you can see, the test passes the "UI Components" double A standard, which is enough contrast for us to work with.

For people with low vision or color blindness, all the controls they need to interact with must have enough contrast to be easily distinguishable from their background



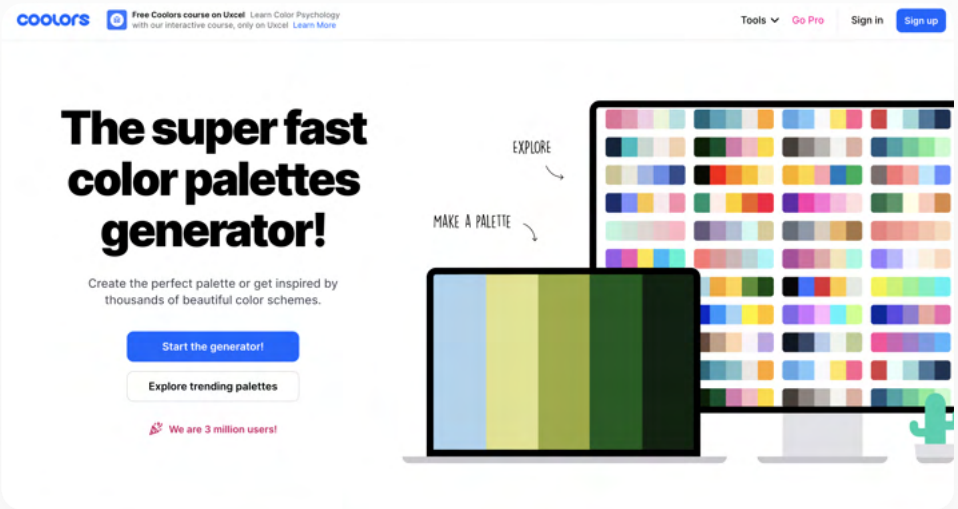
Handy color tools

If you don't know what colors to use or how to combine them to create beautiful palettes, fear not! There are numerous free resources available online to help you generate color palettes. Let's cover a few!

Coolors.co - Great for color idea generation if you're up for some random fun and don't know which direction to go.

Adobe Colors - Create color palettes with the color wheel or image, or browse thousands of color combinations from the Adobe community.

Paletton - A professional designer tool for creating fantastic color combinations. Choose a scheme (mono, complementary, analogous, etc.), change one color, and the rest changes to match it.



Credit: Coolors

iOS Typography

If you're interested in a general introduction to typography, I explain the basics and the terms used to describe type in my free "How to design better" ebook. We'll not cover the type anatomy in this part since it's not relevant. Instead, let's focus on iOS typography and the basic principles and most common sizes used in iOS apps.

Apple uses two type families in their own apps. They're free to download and use, and I suggest getting them from the official site.

The first font, which you'll probably use the most, is the San Francisco (SF) font. San Francisco is a sans serif type family that includes SF Pro, SF Pro Rounded, SF Mono, SF Compact, and SF Compact Rounded. SF Pro is the system font in iOS, macOS, and tvOS; SF Compact is the system font in watchOS.

The phrase 'The quick brown fox jumps over the lazy dog.' shown in San Francisco Pro.

New York (NY). New York is a serif typeface that provides a unique tone to complement the SF fonts. NY works as well in a graphic display context (at large sizes) as it does in a reading context (at text sizes).

The phrase 'The quick brown fox jumps over the lazy dog.' shown in New York.

You can download the San Francisco and New York fonts from the official Apple site at https://developer.apple.com/fonts/.

iPhone Typography Guidelines

This is one of the most confusing topics in app design - what sizes of fonts to use for each typography element? Here's a little breakdown.

Titles	17px	Page titles should be 28-34pt before scrolling, 17pt when scrolled
Body text	15-17px	Depending on the font
Secondary text	13-15px	Lighter color
Captions	12-13px	Used for segmented controls
Small text	11-12px	Used depending on the context
Buttons, text inputs	17px	Important buttons with medium font weight
Tab bar	10-11px	You shouldn't go smaller than 10pt

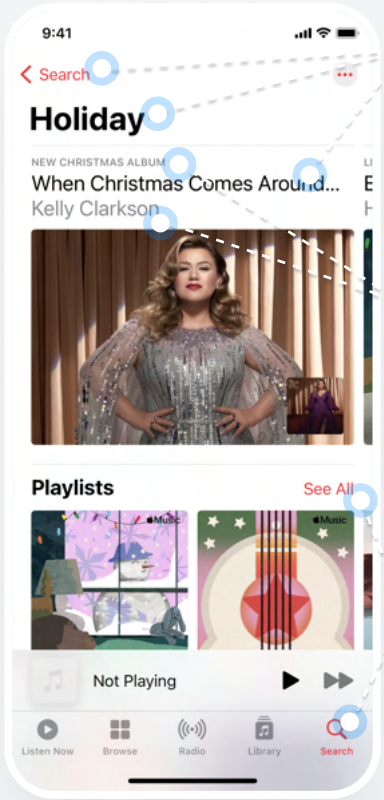
Body Font

You should always pick a font that, apart from looking clean, is legible and comfortable to read. When in doubt, refer to the most popular Google fonts or use the default iOS font - SF Pro Text. Here are my top recommendations for body text.

SF Pro Text	Inter	Open Sans
General Sans	Helvetica Neue	Roboto

Font Weight

Modern fonts come with variable or multiple font weights: Ultralight, Thin, Light, Medium, Regular, Semibold, Bold, Heavy, Black. How do you decide on which one to use for what size? The general consensus is that for sizes 11-19 px, you should use a Regular SF Text weight. For sizes 20-34 px, you should use SF Display Medium, and at 34 px+, go with SF Display Bold. A similar idea applies to all your custom fonts.



Take note of all the font weights and sizes Apple uses for their iOS apps.

Apart from different sizes, they use higher/lower contrast text to add more visual hierarchy.

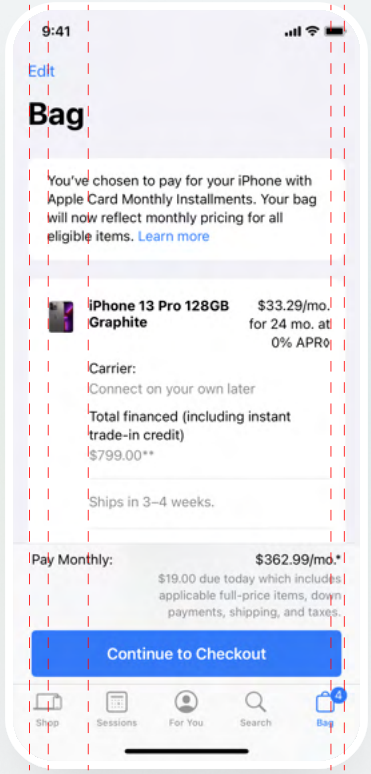
Accent colors are used for button links, icons, and active states.

Text alignment

Alignment is king; there's no question about it. When your content is aligned, it removes the clutter and makes the text much easier to read. But how can you ensure everything is aligned to one position? **Introduce grids and mind the side margin for all your components.**

We'll cover grids in the practical part, but for now, let's talk about theory and show you some examples of text alignment.

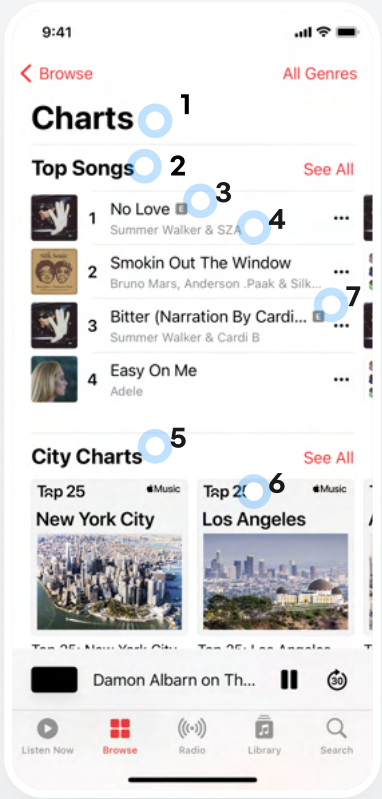
Look how the Apple Store app uses alignment to convey the hierarchy of the screen. The topmost content is aligned to the side margin (16 px); below sections are nested another 20 px inside, with the phone specs being pushed 80 px from the side margin. Your eyes follow this pattern seamlessly to the price details and then a full-width CTA button to take the desired action. What's also interesting is that the button remains in one position and is always visible. Why is that? You want to have at least one Call-to-Action visible at all times.



Hierarchy

Typography is one of the main elements that create hierarchy, the second being the white space. It makes it easier to distinguish the order of importance by looking at how text elements are positioned on a screen and how high their contrast is against the background.

Let's analyze a relatively busy Apple Music screen.



I labeled by numbers the order in which I looked at this screen without overthinking the meaning of the content.

Look how Apple uses fonts to convey hierarchy. We start with 34 px Display text, move into a 22 px Headline, and follow with a 17 px song title, with artists being 13 px at around 50% reduced contrast.

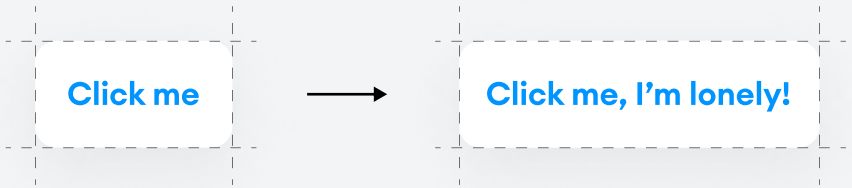
You can learn A LOT about hierarchy just by analyzing popular apps.

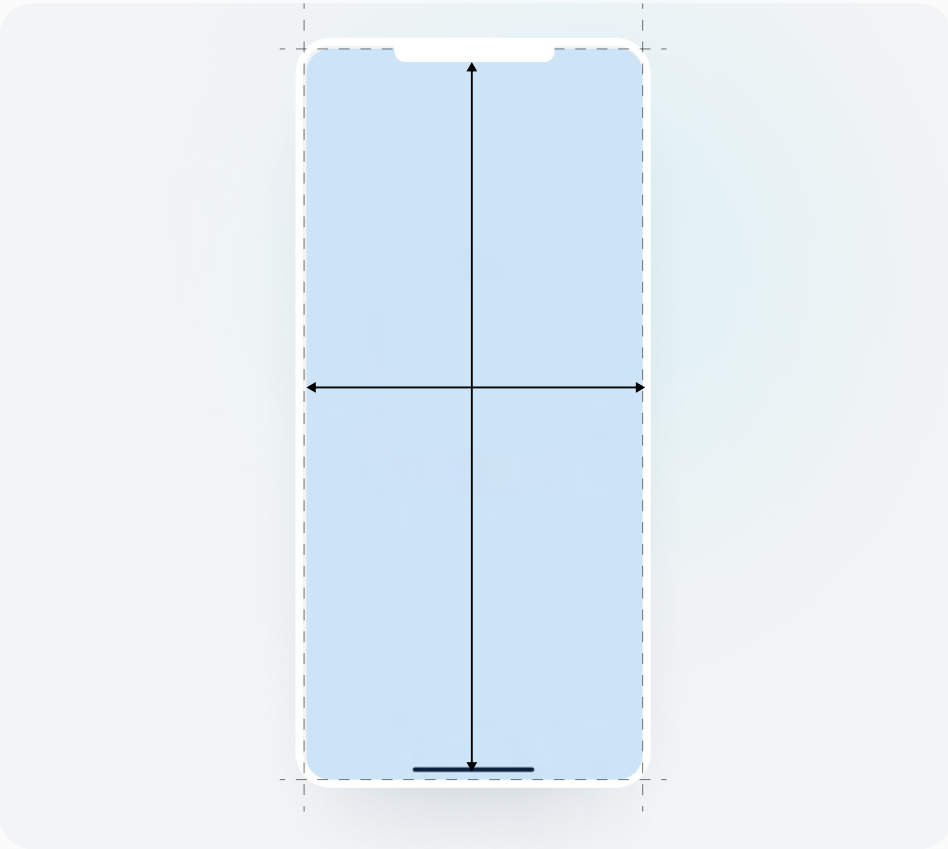
Adaptivity and Layout

The adaptive layout is fundamental when designing and developing an application. This chapter will discuss some ways to ensure adaptivity while creating our application. With an adaptive layout, our hypothetical users can use the application on different devices. Our design remains responsive and adapts to set boundaries.

Auto layout basics

As the name suggests, the auto layout is dynamic. It adjusts the container height when you're adding or removing child elements or even when you're resizing your containers manually to test out different placements. We'll utilize auto layout on every component we design in our app, so you'll get to see and understand how it works when adding and removing elements from our containers.





Why should you use auto-layout?

When designing using auto-layout, you don't have to worry about aligning and resizing. You don't have to worry about making space. When you plan out your content correctly, you won't have to worry about stretching and squashing elements each time you have to fit something within a container; you don't have to worry about consistent spacing when you're duplicating things like lists or cards.

Before we move on to practice auto layout in our design tool (Figma), we need to cover a few more things that will help us understand this concept a little bit better.

Screen Size

When designing for iOS, we must understand different screen sizes and what they mean for our application. We need to ensure that our app works flawlessly on all the available mobile devices and that our content is designed in a way that it adapts to all screen sizes without manually adjusting the design at different mobile breakpoints. Let's see what devices we should be designing for.

iPhone 13 Pro Max	428x926 pt (1284x2778 px @3x)
iPhone 13 Pro	390x844 pt (1170x2532 px @3x)
iPhone 13	390x844 pt (1170x2532 px @3x)
iPhone 13 mini	375x812 pt (1125x2436 px @3x)
iPhone 12 Pro Max	428x926 pt (1284x2778 px @3x)
iPhone 12 Pro	390x844 pt (1170x2532 px @3x)
iPhone 12	390x844 pt (1170x2532 px @3x)
iPhone 12 mini	375x812 pt (1125x2436 px @3x)
iPhone 11 Pro Max	414x896 pt (1242x2688 px @3x)
iPhone 11 Pro	375x812 pt (1125x2436 px @3x)
iPhone 11	414x896 pt (828x1792 px @2x)
iPhone Xs Max	414x896 pt (1242x2688 px @3x)
iPhone Xs	375x812 pt (1125x2436 px @3x)
iPhone Xʀ	414x896 pt (828x1792 px @2x)
iPhone X	375x812 pt (1125x2436 px @3x)

Why only these resolutions and not lower? Well, the most important dimension we should consider is the width - which in our case should be 375 px. Therefore, it's thought to be the best to start designing on the latest, but at the same time, the smallest iPhone frame available, which by the time of writing this e-book, is the iPhone 13 mini. You have to remember that designing for the smallest resolution ensures that you can, in fact, fit all the design elements you need into a frame, and scaling our designs up is always easier than scaling them down.

You might be asking shouldn't we design for iPhone 8,7,6, etc., since a lot of people are still using these models? They still have the exact width dimensions; the only thing that changes is their length, which isn't that important because if we set our constraints the right way, our design will just cut off at an earlier point but remain responsive based on our device width.

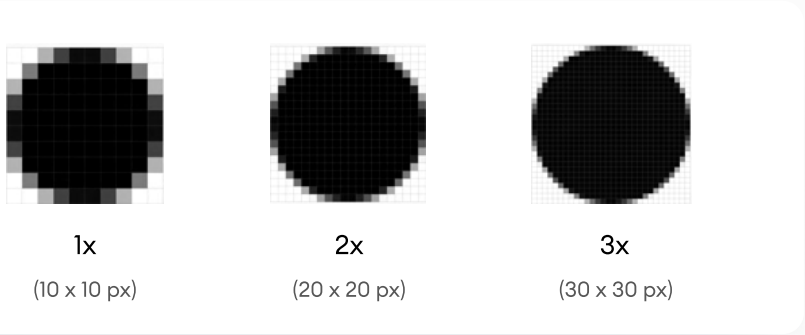
Pixels vs Points

This is a confusing topic for a lot of beginner designers. What's the deal with points, and how are they different from pixels?

Let me explain. A standard-resolution display has a 1:1 pixel density (the @1x stands for), where 1 pixel equals one point.

Scroll down to the next page to see the example in action.

Higher resolution displays have a higher pixel density, increasing by 2.0 (@2x) and 3.0 (@3x). With the introduction of high DPI screens, the industry needed a unified measurement. Look at the example below.



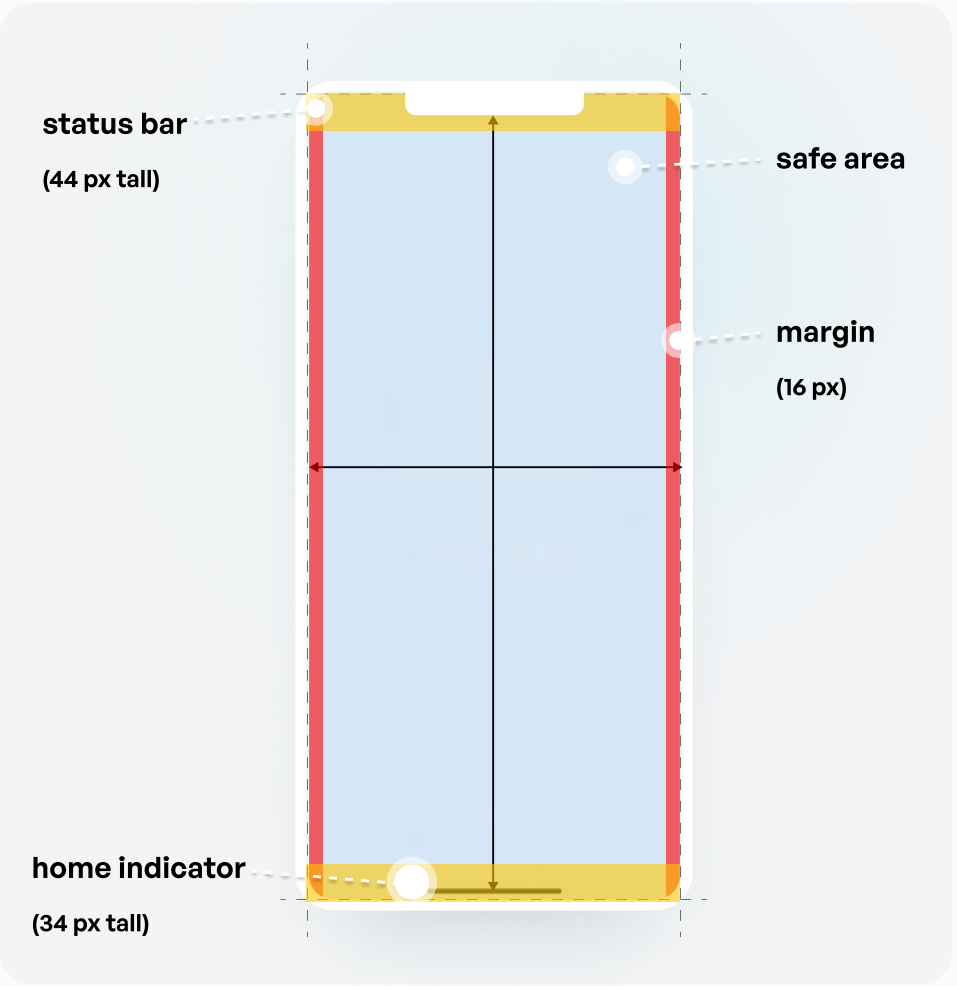
When we are designing our app, we will use pixels to describe our font sizes, padding, and margin values. The only thing you need to remember is that our 1 pixel equals 1 point, and that's all you need to worry about. Points are just a measure of scale, and it's mainly needed as a reference point for our developers.

Since our iPhone 13 has a high DPI screen with three times the density of pixels per inch, all our values will be multiplied by three when speaking about real-size phone dimensions, not that we'll have to. We'll cover that in more detail in our design hand-off documentation at the end of this e-book, right after we're done with our Fintech app design. For now, treat 1 pixel as 1 point, and don't worry about the rest.

Let's move on!

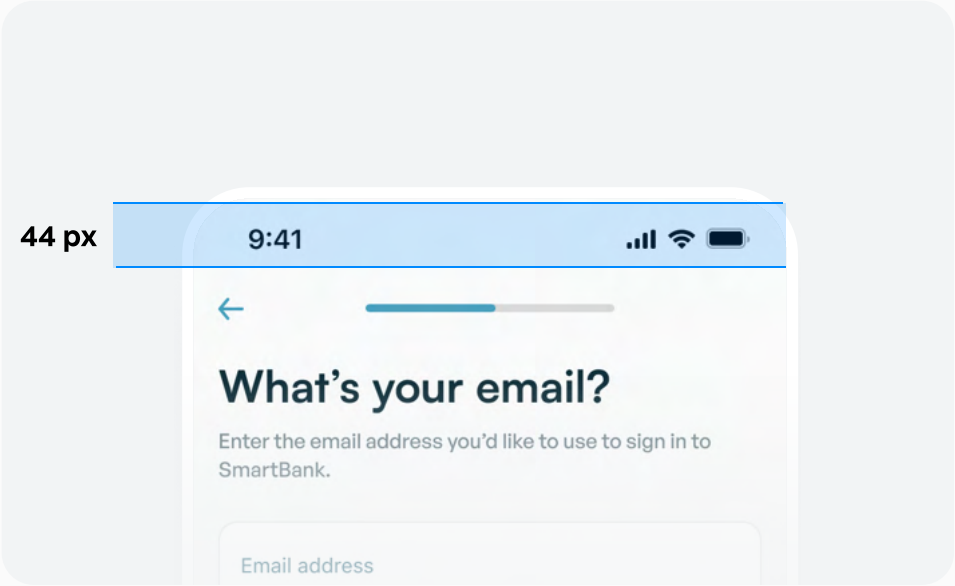
Layout guides and safe areas

When designing for iPhone, you have to remember the rounded corners and the notch clipping the content. To ensure your designs aren't obscured by these elements, you should use a safe area layout guide. Here's what it looks like.

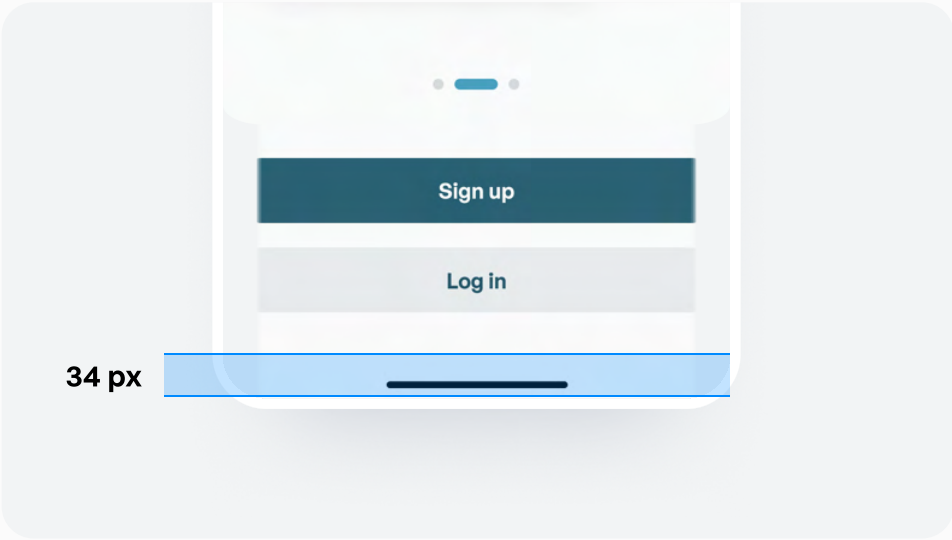


What are the exact values you should mind when creating your designs? Let's talk about that.

The status bar - is 44 px tall, so you should always make sure there's no clipping there and leave some space below your screen. The best is to paste your status bar on every frame to have a visual reference for how much space you should go.



Home indicator - safe space for home indicators should be 34 px tall, although I always try to place my bottom-most CTAs and content at least 40 px from the bottom of the frame to avoid accidental clicks or swipes. Default settings for these two components you should set once and then copy-paste them to all your frames.



Side margins - according to iOS, the minimum side margins should be at least 14-16 px wide. For most apps, you'll find 16 px as the default value, which provides the highest amount of screen estate without compromising the legibility of your content.

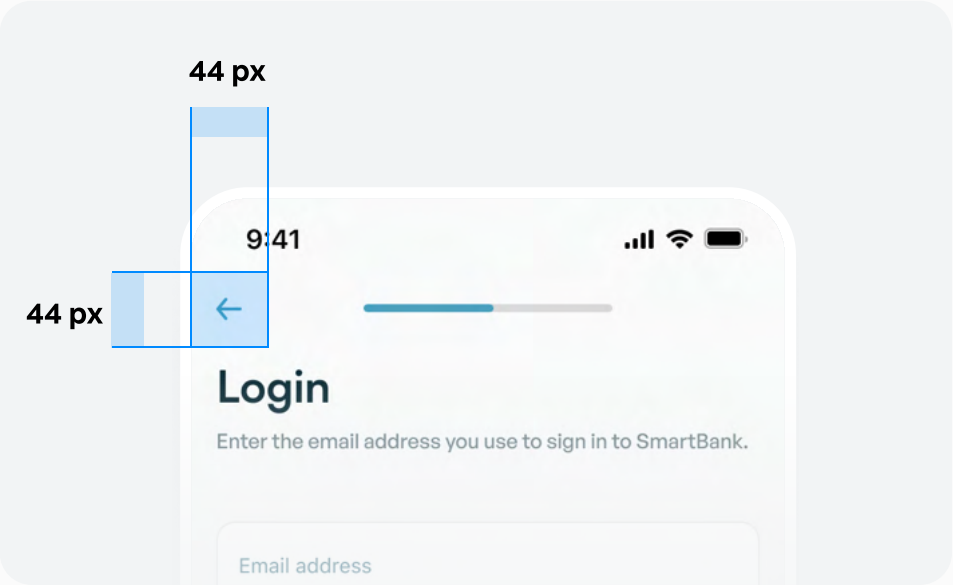
However, navigation controls such as back buttons will extend a bit further to the left end of the screen to around a 10-14 px margin. Why is that? It's because of their touch target size. Even though the icon itself might be 18 px wide and tall, the container it's sitting in is at least 44 x 44px big, which is the minimum size recommended by iOS to be considered a touch target.

What are touch targets? Open the next page and see for yourself.

Touch targets

Buttons and other interactive elements should be of a size that allows the user to precisely tap the target with their fingertip. As we mentioned on a previous page, a standard and minimum touch target for iOS apps should be 44 px wide and 44 px tall. Touch targets should go beyond an element's visible boundaries.

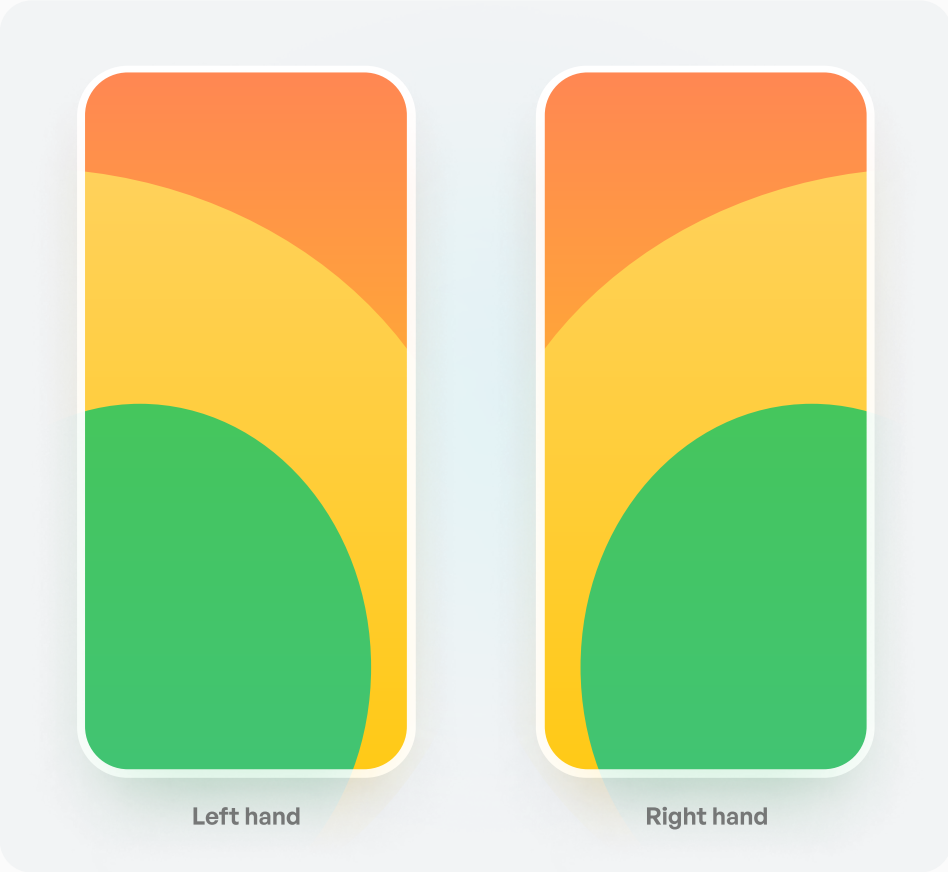
A button or an icon, for example, can appear to be much smaller, yet the surrounding padding contains the entire 44 x 44 px touch target.



Pixels are just a reference measure for designing interfaces. Real dimensions of a touch target should be considered in mm but using the minimum required px (pt) values we ensure our design meets the minimum iOS criteria.

Thumb zone

Most mobile users use their thumbs to control, click and navigate their mobile devices. The thumb zone on the screen is the area that is comfortable and easy to reach using just the thumb. Unless you're a giant and can reach the top of your phone with a thumb, you shouldn't consider the top of the screen as easily accessible.



The above example shows how the thumb zone looks depending on which hand is used.

Also, this image doesn't consider different versions of the phone. Users with smaller devices will have a wider natural area, while the user with much bigger devices like the iPhone Pro Max series will have a much more difficult time navigating the screen with one hand.

Reach is determining how easy or hard it will be to navigate the product. The hamburger menu design pattern is not ideal for mobile design - it's in the least favorite spot you can imagine.

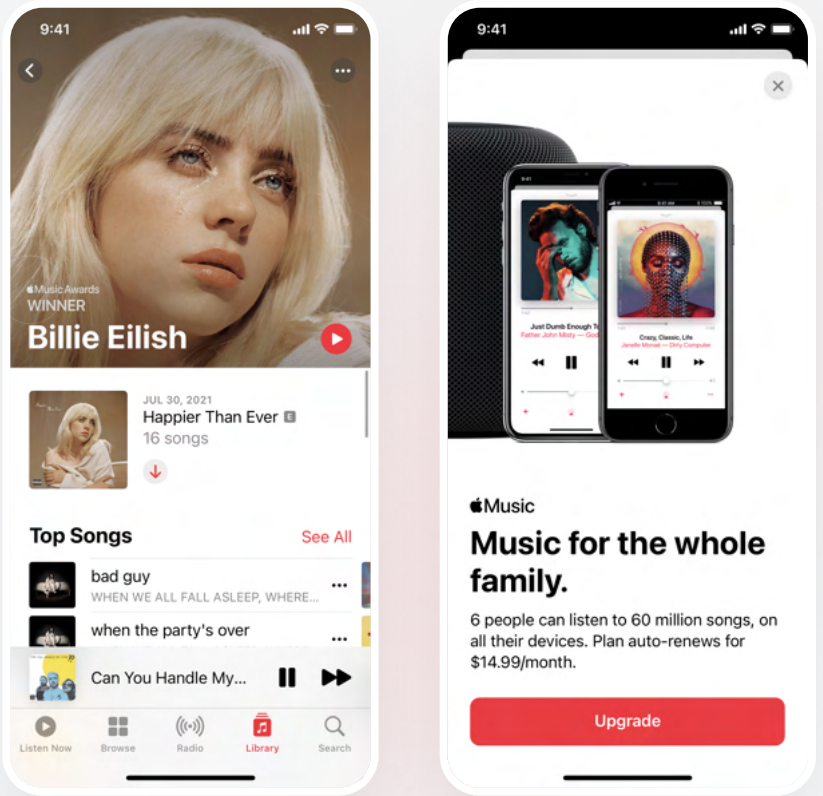


Having a tab bar at the bottom of the screen is the most ideal navigation pattern you can have in a mobile app.

Consider hiding secondary information that's not necessary for your app functioning in elements like hamburger menus, avatars, or "meatballs" icons.

Navigation elements placement

Navigating your app should be as easy as possible. Therefore, you always have to ensure that your app's flow is simple. The main app areas are accessible through your tab bar or a modal, popover, or any navigation element overlaid on top of your active screen. The navigation has to be consistent across your entire app.



Notice how Apple's CTAs and controls are always in the thumb zone.

iOS Materials

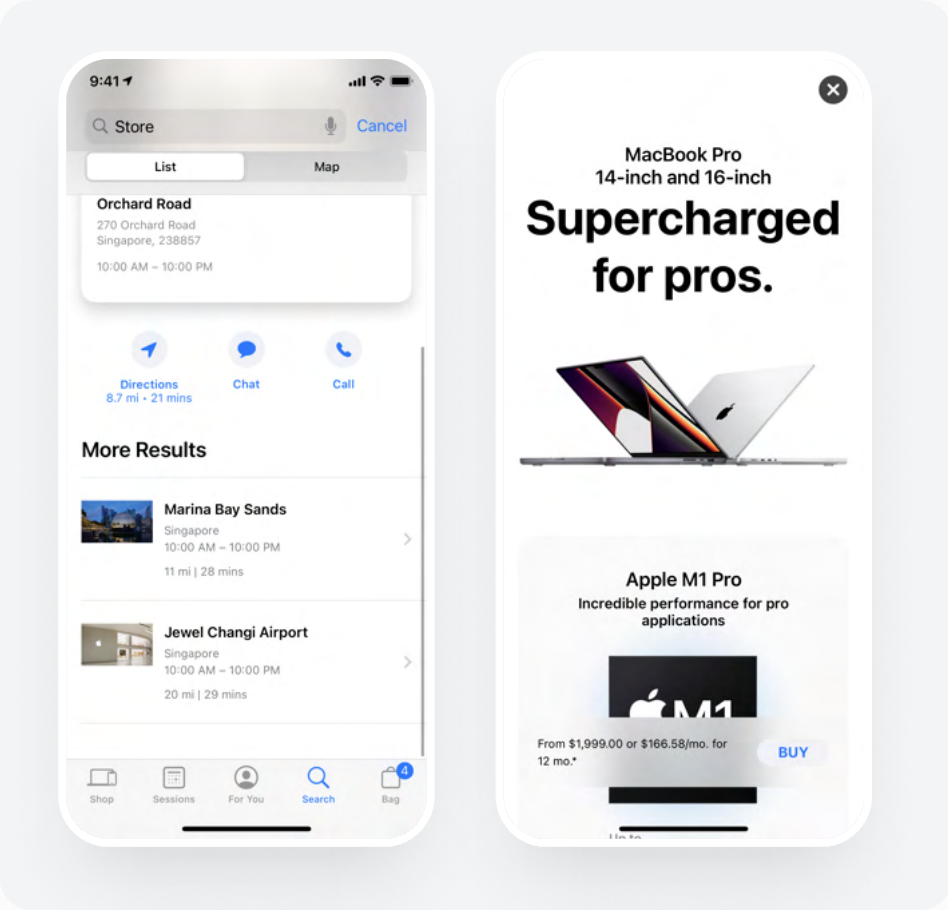
iOS provides materials/blur effects that create a translucent effect you can use to add another layer of depth to your interface. The slight addition of a blur lets (iOS) views and controls hint at background content without distracting users from foreground content.

To produce this effect, iOS materials allow background layout to subtly pass through foreground content while blurring the background enough to maintain legibility.

iOS defines materials you can use in specific areas to control the visual separation between the foreground content and the background appearance. The system-provided materials include light and dark variants that work well against most backgrounds. We'll replicate these blur variants in Figma in the "Practical" part of this e-book, so don't worry about the specifics yet.

Background blur's do's and don'ts

Blurs should not be used everywhere unless there is overlaying material. Blurs should be used in a few critical areas to draw the user's attention to crucial information. Most of the time, the contrast between background and foreground material is sufficient enough to avoid it. As with almost everything in UI design, it's more about knowing when and where to use it than using it all the time.



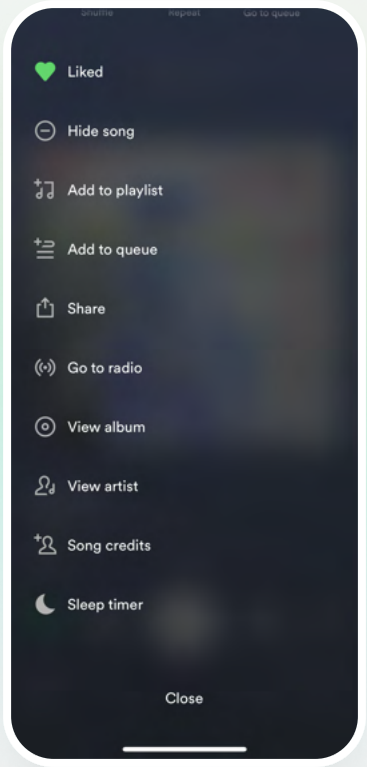
Credit: Apple Store

Notice how the background and foreground content get separated to make the text more readable without drastically changing the background color.

Blur is also used to indicate underlying content that a user is about to scroll down to and explore, hinting at the next element.

Bring in the focus

By adding a blur behind a modal window or an iOS view, we can direct users' attention to the important content by ensuring it's appropriately highlighted. This blur is especially prominent with apps like Apple Music and Spotify. Take a look below.

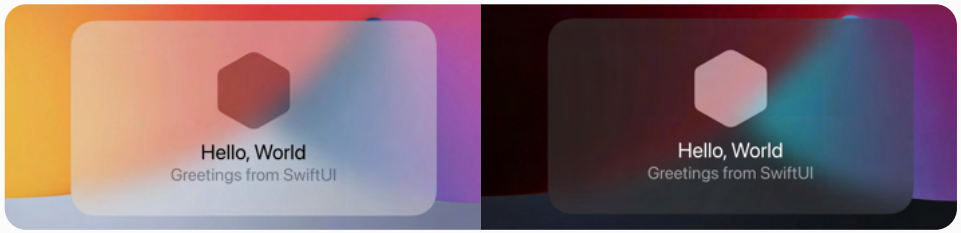


Credit: Spotify song details

Adding blur is especially useful to separate popovers and modal views.

System defined materials

We all can use iOS defined system materials in specific areas to control the visual separation between the foreground content and the background appearance to match the default theme. These materials have different translucency, blur, and vibrancy values applied for you to choose the best version you need.

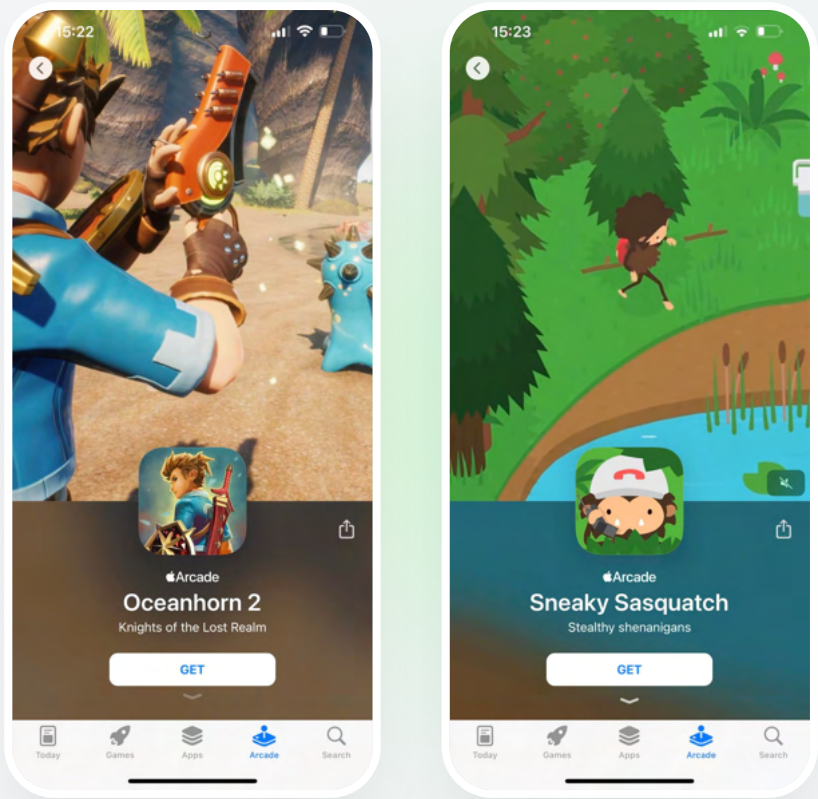


Vibrance

One of the most significant benefits of using a blur is its ability to adapt to different background colors while still maintaining enough legibility of the foreground content to be easily readable. Using solid colors as separators wouldn't allow you that flexibility. It's an exciting way to add more liveliness to your designs and push them to another level. Look at the example on the next page.

Using the system-defined materials makes your elements look great in every context because these effects automatically adapt to the system's light and dark modes.

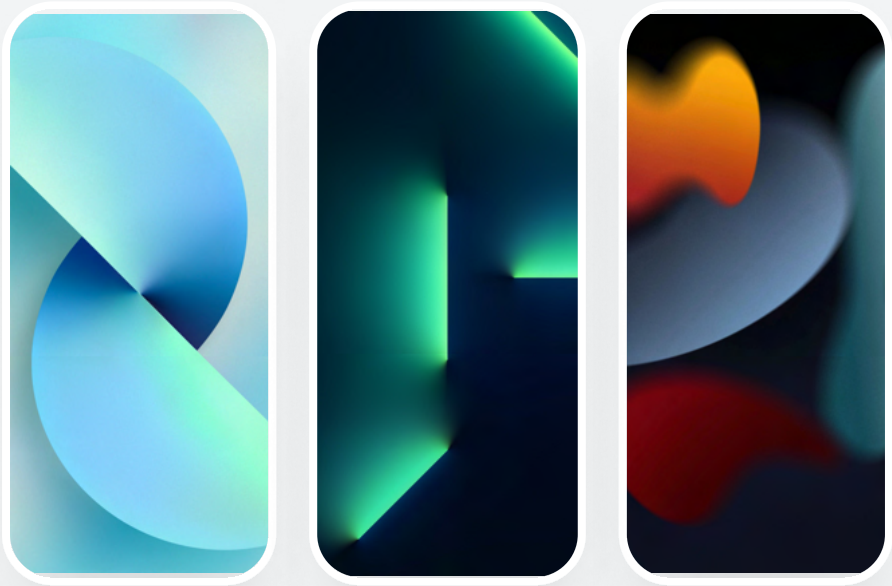




Credit: Apple Arcade

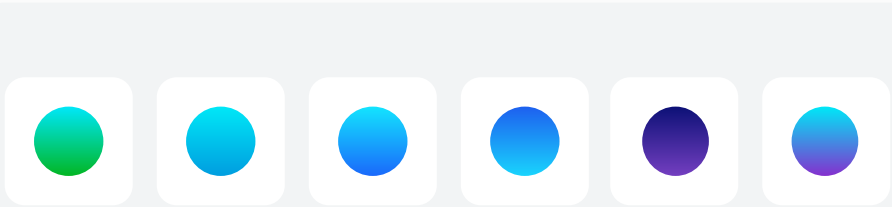
Of course, you won't be able to apply blur to every background element you design in your app. Use it with common sense and try to avoid abusing it. It's best used for your header sections and modal views. We'll show you the live examples of where we use it in our Fintech app in the following "Practical" chapters of this e-book.

Graphics and Gradients in iOS



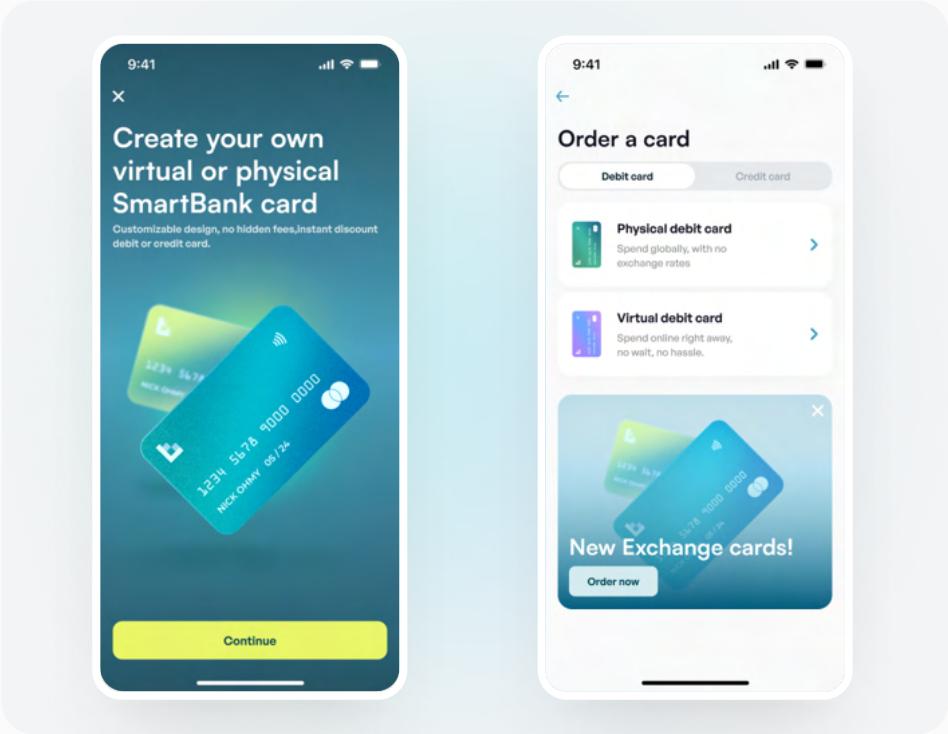
Apple 13 Pro Wallpapers

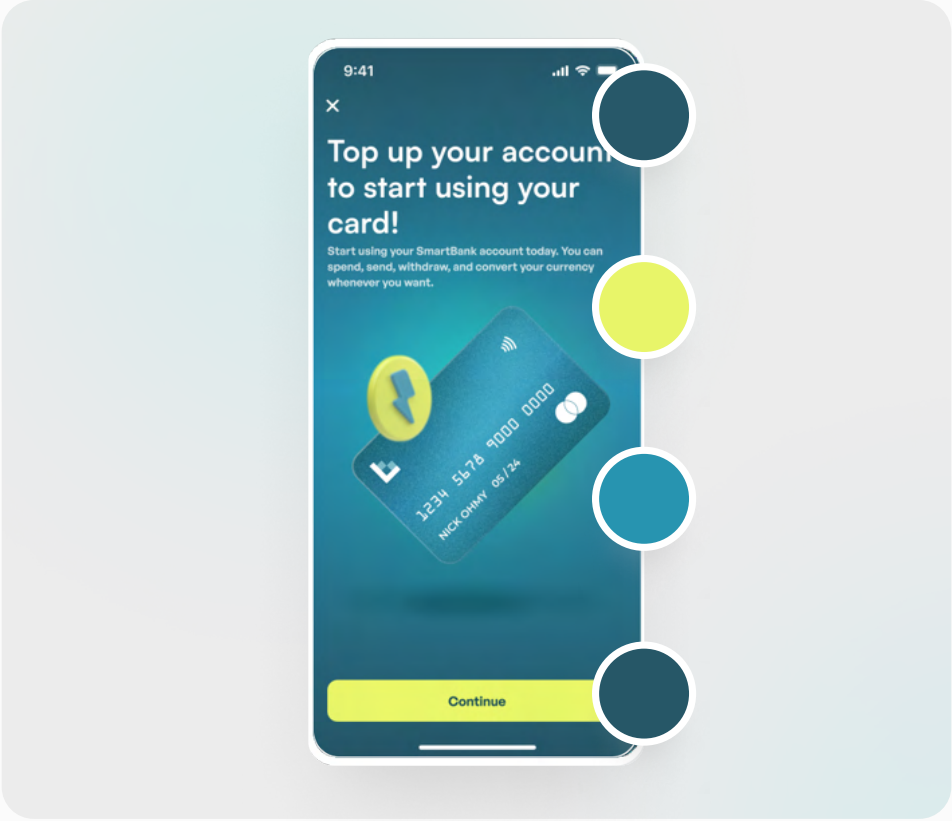
Apple is known for its incredible, stellar background wallpapers. Contrary to Android design which is mainly flat, iOS utilizes many gradient colors in its designs. Here are a few examples.



Gradients help your designs leave a more significant visual impact than simple solid colors. We'll use them a lot in our Intro screen designs for the Fintech App to bring our illustrations to the forefront.

The key to mastering gradients in design is finding the right balance and not overly saturating them or using two colors at each end of a color wheel spectrum. To learn more about the gradients, feel free to check my "How to design better UI components" e-book attached to this file. We have a separate chapter explaining different types of gradients and how to match them.





The safest bet would be to choose analogous colors for your gradients, meaning colors near each other on a color scale.

You can play around with your colors and add three different gradient points to create more dynamic-looking gradients.

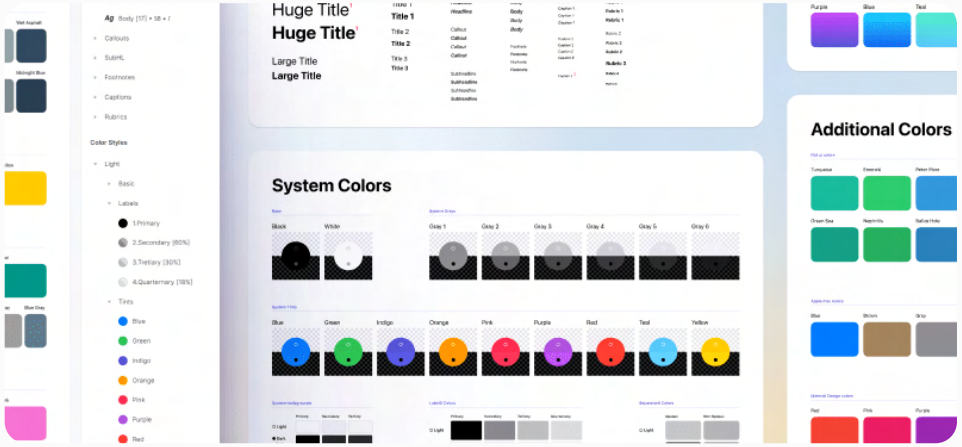
You can also add a blurred, vibrant shape behind your main illustration to bring out to the front even more and add another layer of depth to our designs.

iOS native elements

7. iOS 15 native elements

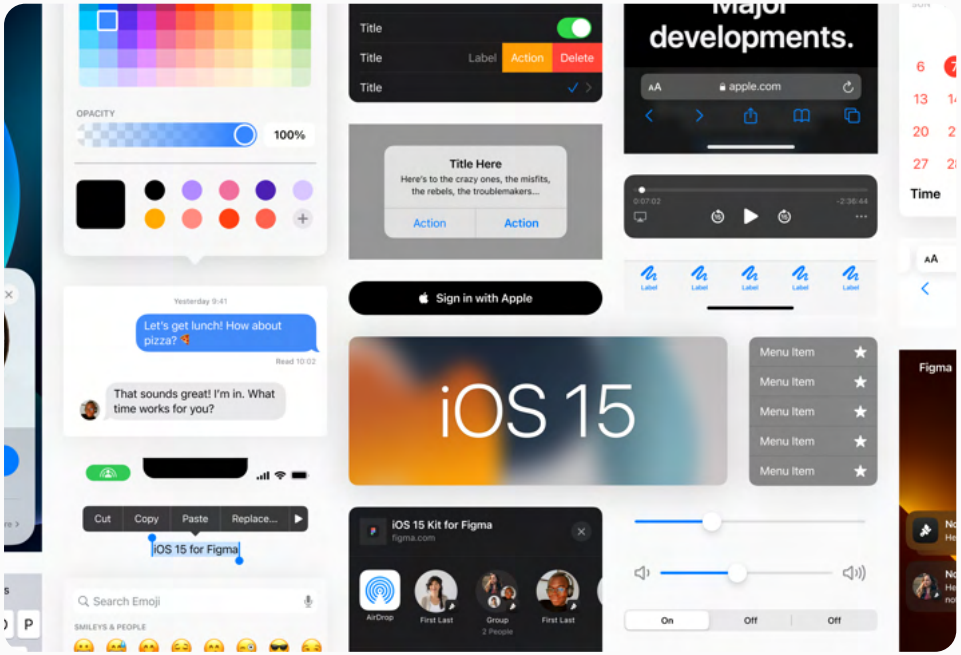
Apple has always had a clear set of design principles and guidelines that help designers and developers design beautiful apps. They are there for you to have a great resource and reference point for your own designs, but you don't have to follow them to a Tee.

Apart from visual guidelines, we were also given access to a massive library of native UI components that help set a better foundation for building your new app and integrating with the rest of Apple platforms. When designing for iOS, you shouldn't replicate new components like modals, alerts, notifications, bottom sheets, etc.



You should always refer to pre-made components and use them to avoid possible errors and make the development of your new app much easier and faster. You are, of course, free to customize them to a certain degree and get more creative each time you design a new tab bar, a segmented control, or a modal popover.

iOS 15 UI Components



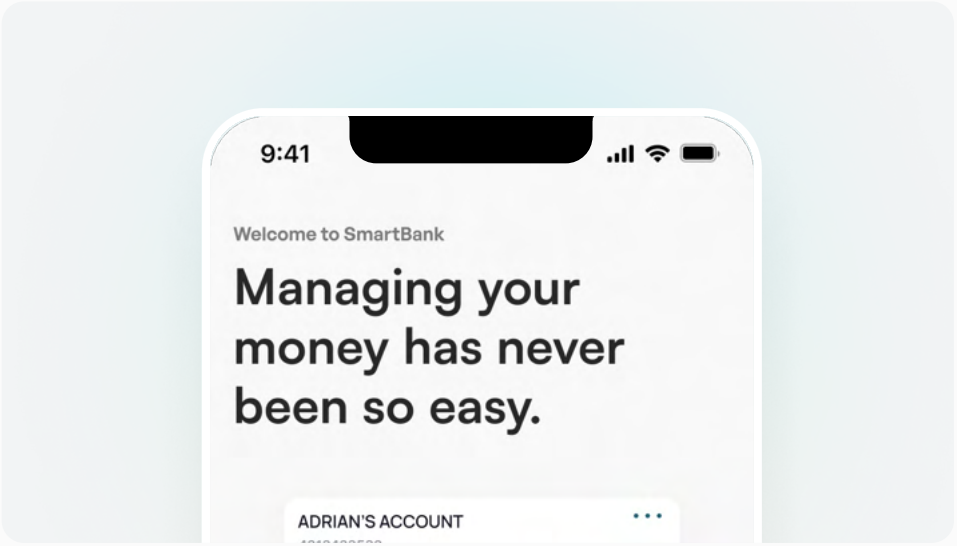
iOS 15 UI Kit by Joey Banks

This file, created by Joey Banks, is single-handedly our most valuable asset when designing iOS apps. Shared with the Figma community entirely for free, you can find it attached in our “Figma files (UI Kit + Practice file)” folder, or you can simply search it on the Figma community and duplicate it manually to your project files.

We will use it a lot. You’ll see why, but for now, let’s cover the main native UI components we’ll get to design and use in our iOS app.

iOS status bar

The status bar appears on the screen's upper border and displays information about the device's current state, time, cellular provider, and battery level. There's a clear distinction between iOS's and Android's status bars, and you should obviously be using the one provided by Apple. We'll be designing on an **iPhone 13 Mini** frame, so make sure you copy the right one from the iOS 15 UI Kit supplied with the practice files. Don't worry, though; we'll get there again.



Status bar from the iOS 15 UI Kit for Figma by Joey Banks

It’s already set up the right way, so we don’t need to play around with any auto layout setting. We need to make sure that it has the proper constraints set. I also tend to delete the notch in the middle. It’s too distracting. This we’ll also cover in the “Practical” part!

It is highly recommended to include the status bar in as many places as possible. Users are constantly looking at their app status, checking it for time, battery, notifications, and signal strength, so don’t obscure it from them.

The text and icons of the status bar can be either white or black by default and switch depending on the background theme currently used, but they can be any color you desire.

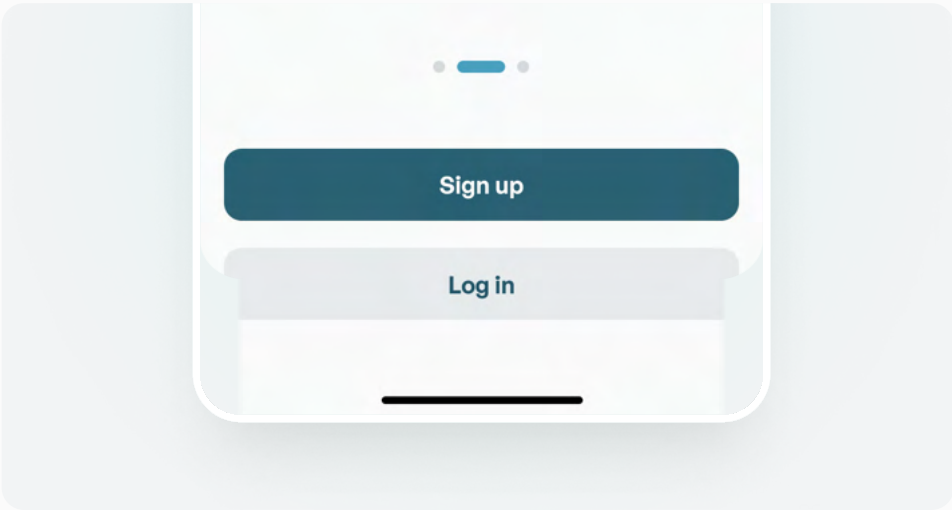


iOS native elements

iOS home indicator

Since the release of the iPhone X, all future devices lack any form of the physical home button. Apple has come up with a new interaction metaphor = the home indicator. The thin bar at the bottom of the iPhone's screen is just that.

The home indicator helps establish the basic, most fundamental, interaction model of the device - how to return to the home screen or swipe between your active apps.



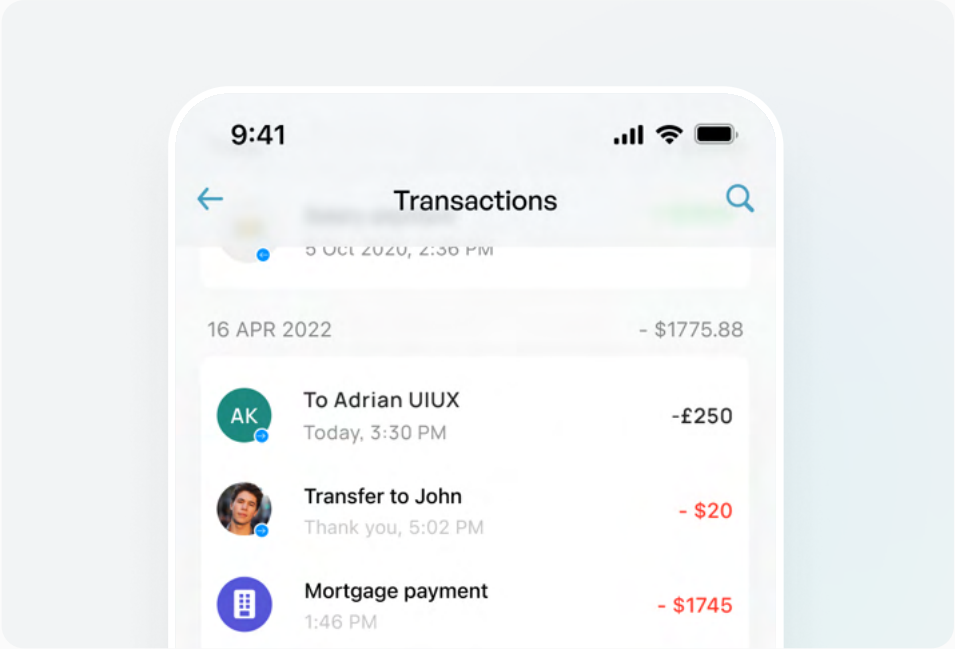
Home indicator from the iOS 15 UI Kit for Figma by Joey Banks

I usually leave it present on all frames of my app, ensuring I have enough safe space between the bottom of the screen and above the indicator - a minimum of 34 px, to be exact.

Navigation bar

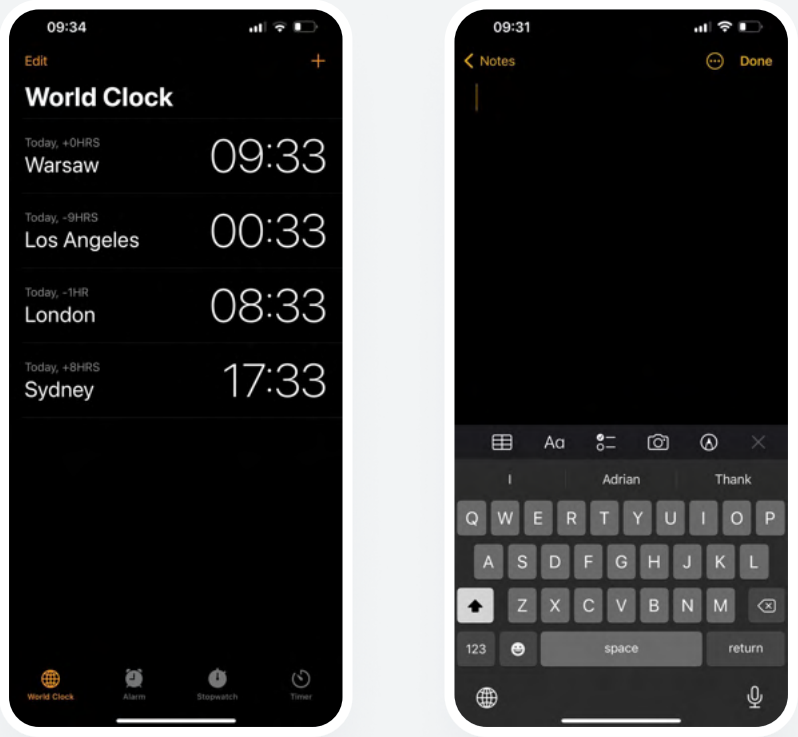
A navigation bar appears at the top of an app screen, right below the status bar, and offers an additional level of navigation in a flow of hierarchical screens. When a new screen is opened from the main starting point, a back button appears in the top left, with a title of a new screen displayed below it or centered at the top when the user scrolls past a certain viewport height.

In some cases, the right side of the navigation bar contains a control, like an Edit, Search, or the Done button, to manage the content within the active view. Navigation bars are translucent, have a background tint, and can be hidden when a gesture occurs or a view resizes.

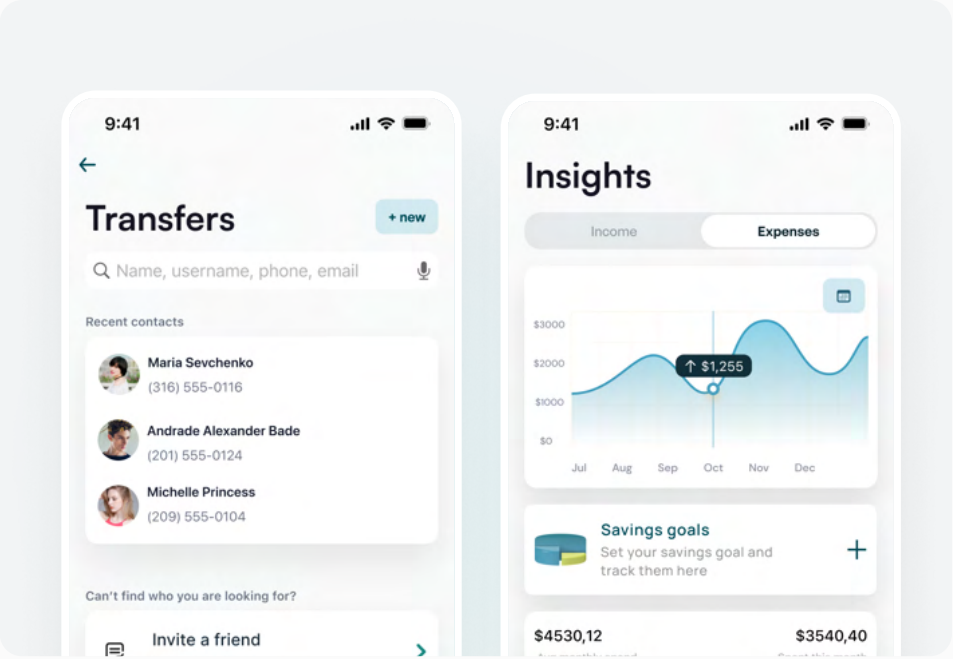


Navigation bar titles

Whenever you're designing your apps, consider showing the title of the current view in the navigation bar. That's a typical design pattern in iOS apps, plus it helps people understand what segment of an app they're looking at. However, if adding a title to a navigation bar seems too obvious, like in the case of Notes, where the first line of content gives you all the content you need (quoted by Human Interface Guidelines), you can leave it empty.

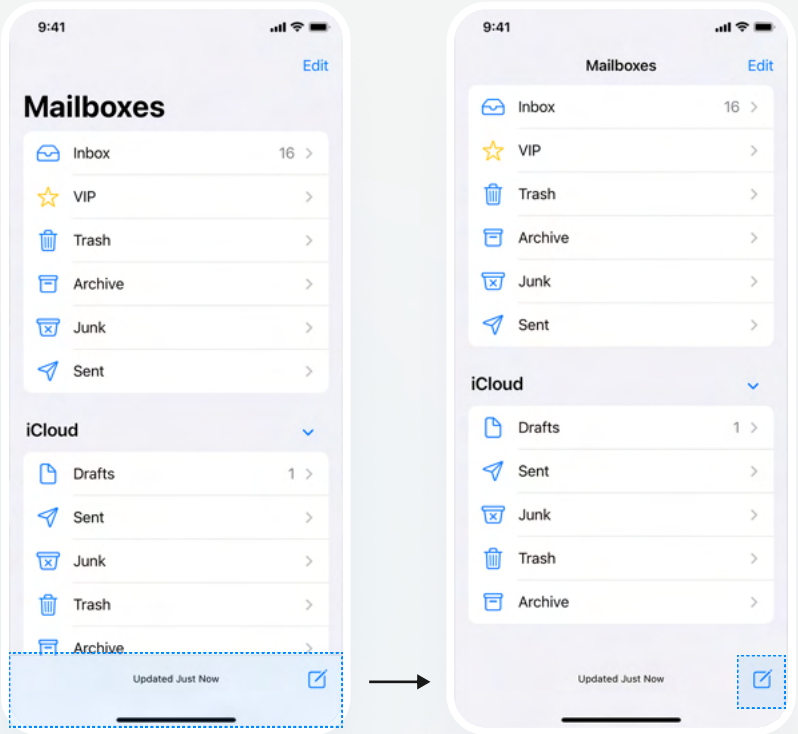


When you need to provide extra emphasis on the context of the screen - use larger titles. This is especially helpful to clarify the active tab and differentiate content areas within your app, as we will do in our Fintech example with cards, accounts, insights, profile settings, etc.



Toolbar

When you need more real estate to place your action buttons and screen status, you'll want to use the Toolbar. A toolbar appears at the bottom of a screen, similar to a tab bar, containing buttons for performing actions relevant to the current view. By default, a toolbar is translucent: It uses background material only when content appears behind it, removing the material when the view scrolls to the bottom.



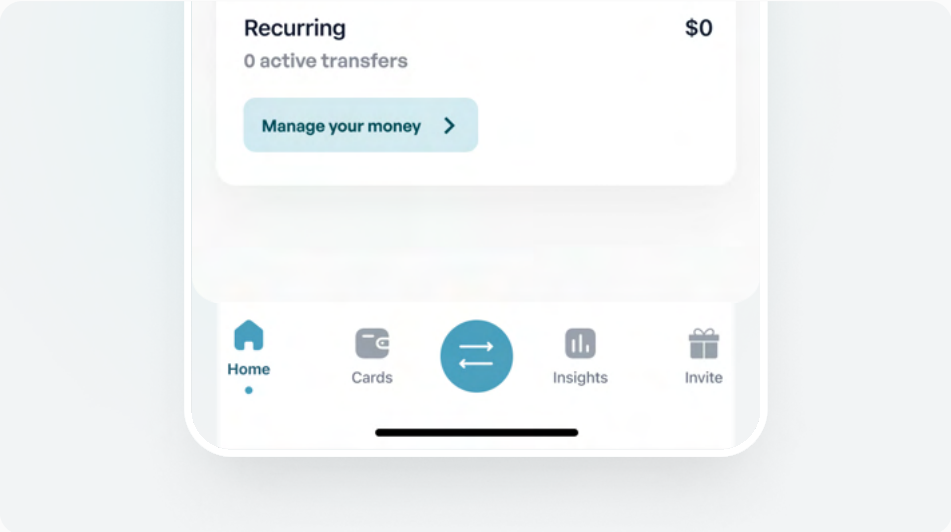
The Mail toolbar uses a background material to distinguish itself from the mailboxes behind it. It disappears only when no mailboxes appear behind the toolbar.

Both toolbars and tab bars appear at the bottom of the screen, each with a different purpose. A toolbar contains different buttons for performing actions relating to the screen, while tab bar lets people navigate between main areas of an app.



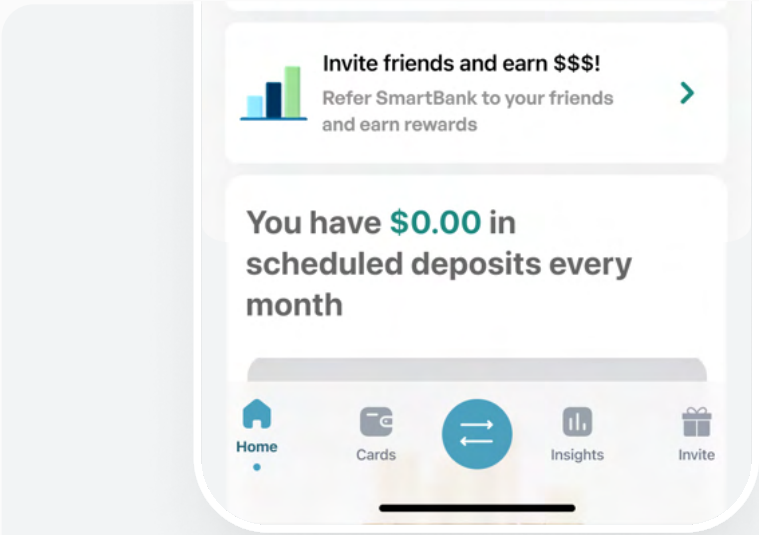
Tab bar

The Tab Bar is the bottommost bar hosting different buttons, usually complemented with labels, acting as a primary method of switching between the main areas of an app. Don't try and hide your navigation in a hamburger menu; obvious always wins, and menus that are always visible will enhance the user experience. It's also good to put labels under your tab bar icons because most people won't recognize symbols right away, especially if they're not universally known.



We can, of course, make our tab bar translucent, as iOS recommends doing by default. If we decide to include that, use background material only when content appears behind it and remove the material when the view scrolls to the bottom. A tab bar hides when a keyboard is onscreen to make space for the rest of the content.

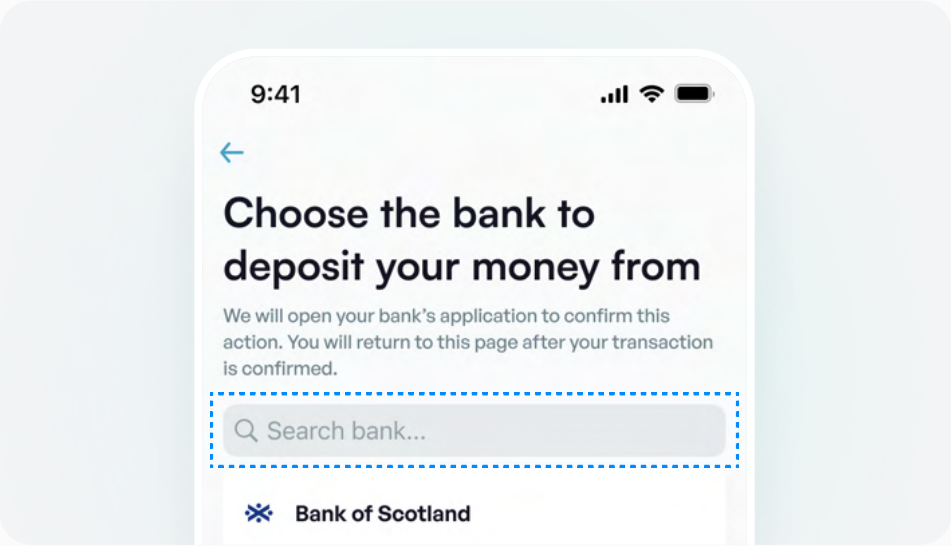
Here's an example of a translucent tab bar used in our designs.



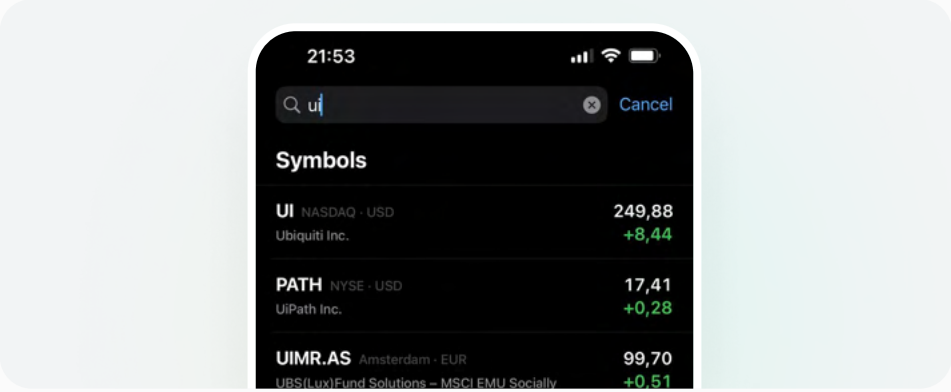
Our home screen tab bar uses a translucent background material to distinguish itself while letting underlapping content show through.

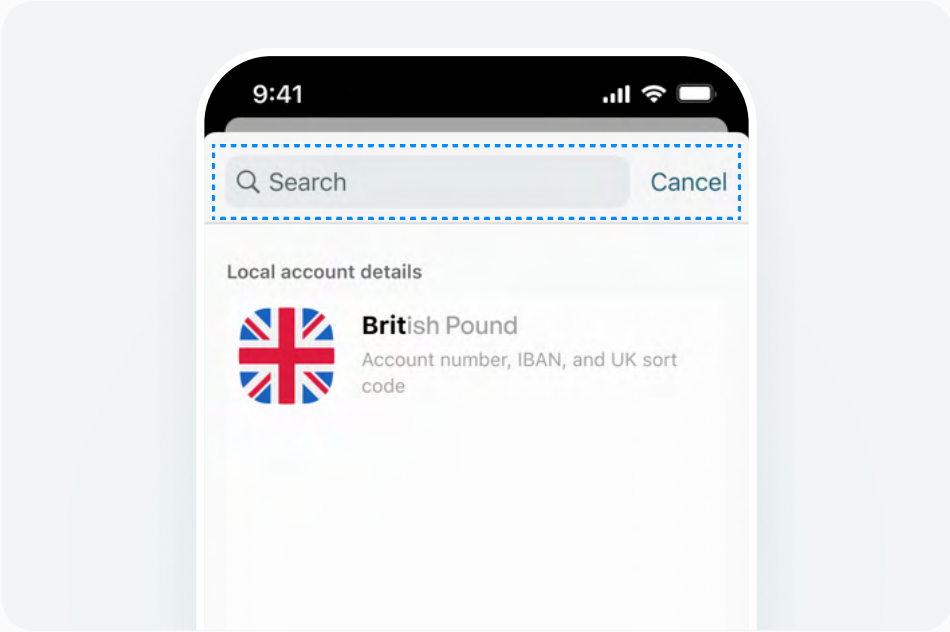
Search bar

A search bar allows people to search through an extensive collection of values by typing text into an input field. It's usually best used for lists containing more than 6-8 items. It can be displayed alone or in a navigation bar or content view. When displayed in a navigation bar, a search bar can be pinned to the navigation bar so it's always accessible, or it can be collapsed until the user swipes down to reveal it. We'll be using both placements for our search bar in our app.



We need to remember a few things when designing our search bars. First, it should be used for more extensive lists of items. Second, you should always enable the Clear button, either a cancel button icon or a Cancel text. Third, you should provide hints and add context to your search inputs. When users start typing a search query, show them live results ans suggest different options.

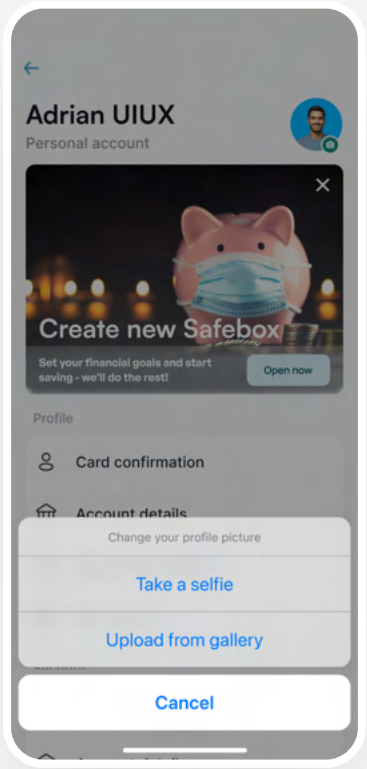




You should consider providing several helpful shortcuts and other content below a search bar. Use the area under a search bar to help people get to content faster. Tap one at any time without typing any more characters.

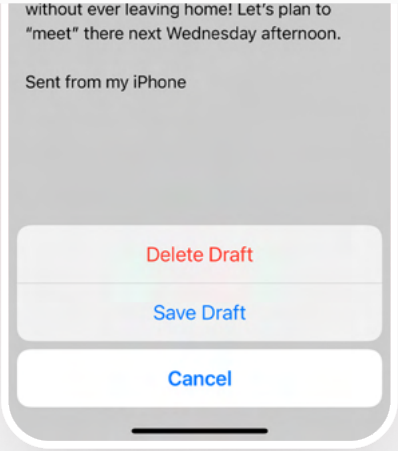
Search autosuggestion is a considerable time saver. While in our app, they don't offer a colossal discoverability boost, they still have their use when a user is looking for balances or languages to select.

Action Sheets



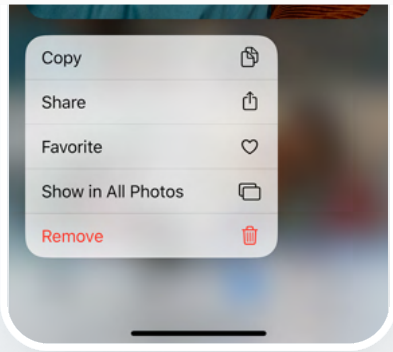
An action sheet presents two or more choices related to an intentional user action. An action sheet slides up from the bottom of the screen on mobile devices.

We use action sheets to provide choices related to people's actions, like uploading a new profile photo in our Fintech App.



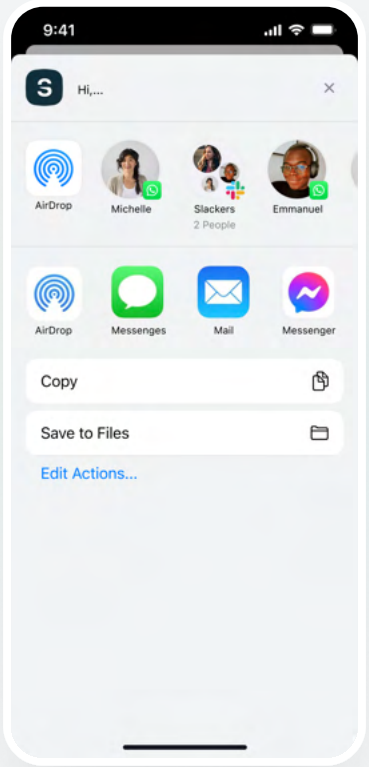
You should accentuate them and make them more prominent by changing their color style for more destructive actions. We also don't want the action sheet to have too many controls. Try to minimize the options to 3 at a time.

Use a context menu if you need to provide items that aren't closely related to the current action. See the example below.



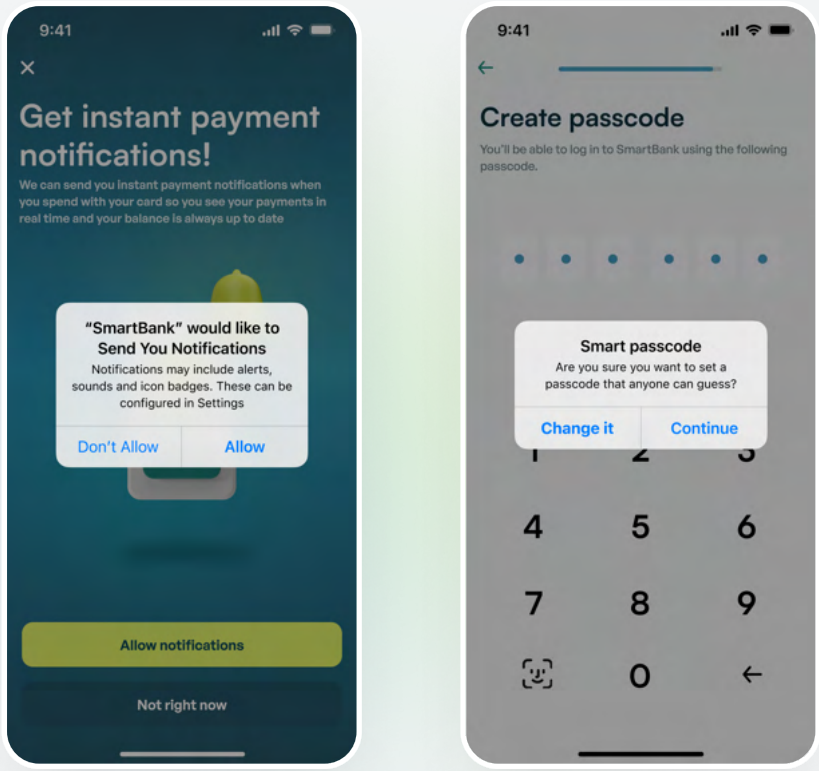
Activity views

By activity, we mean tasks, such as Copy, Favorite, or Find, useful in the current screen. Once started, an action can perform a task immediately or ask to provide additional information before proceeding. All activities are managed within an activity view, which in mobile devices almost always is a Sheet. In our Fintech App, we use activities for sharing information outside our app.



Alerts

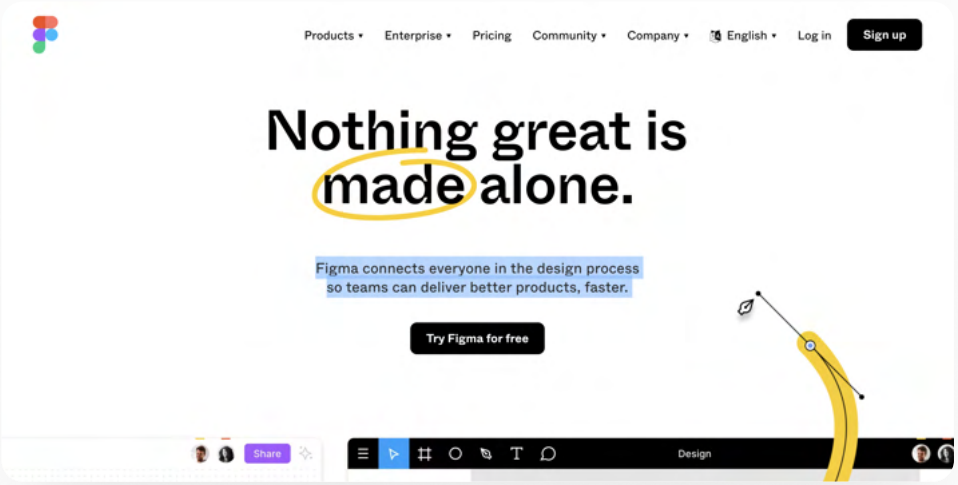
Alerts are used to show critical information related to the device's state or action being taken. An alert consists of a title, an optional descriptive message, one or more buttons, and optional text fields for collecting input. Aside from these configurable elements, you can't customize the visual appearance of an alert, so we ought to use the one provided in the iOS 15 UI Kit. That's all we need to do.



Figma design basics

8. Let’s talk about Figma

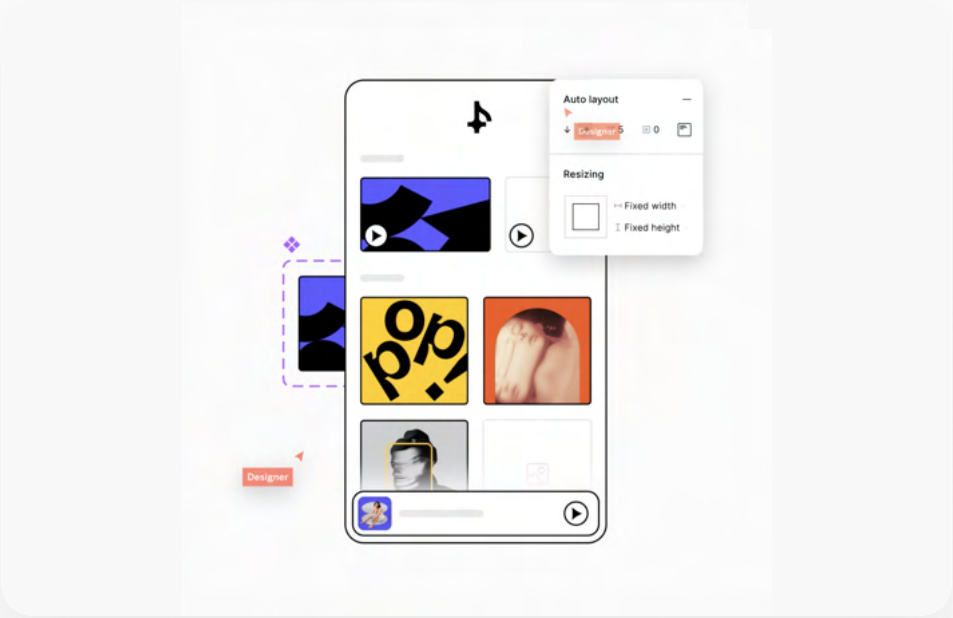
Back in the day, the only design tools we had were Photoshop and Sketch. We’re blessed to have seen more players emerging in our industry, with Figma being our personal number one. This handbook focuses on Figma only, but if you’re really fixed on using Adobe Xd or Sketch, you should be able to replicate what we do here to some extent. We chose Figma for its amazing auto layout and constraints capabilities and a huge collection of community resources, plugins, and developer-friendly interface.



If you know how to operate Figma, feel free to skim through this chapter and look for things that interest you. Otherwise, I will walk you through the whole Figma interface and provide you with step-by-step instructions on how to best use the features created for us by Figma in the most efficient way possible. If you're a Pro in Figma, feel free to jump into the next chapter. Without further ado, let's go!

What exactly is Figma?

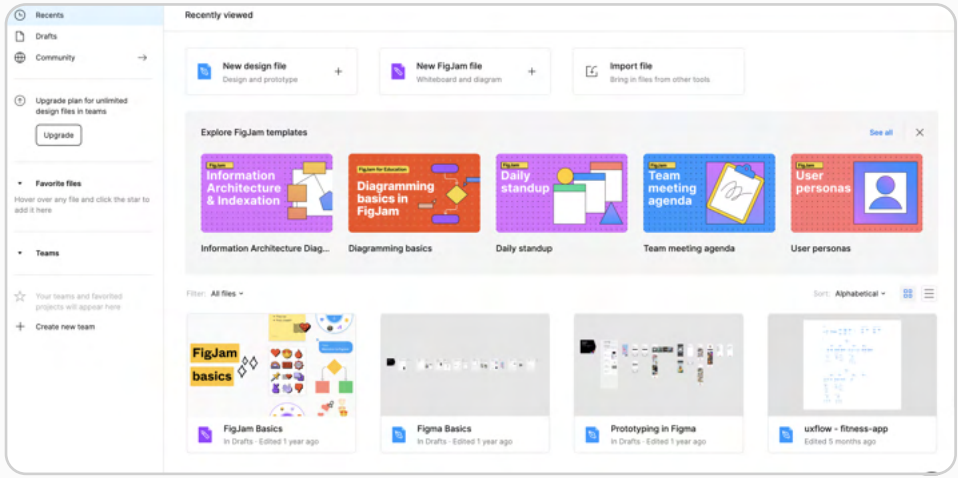
Figma's calling themselves the collaborative interface design tool. And I got to say, it does bring out the collaborative feature to the forefront. It gives designers the ability to share a design file with team members and clients to get instant feedback from each other. This was a game-changer when they first introduced it, as it meant that teams could collaborate on projects in real-time, and the time spent on the back-and-forths was brought down to a minimum.



Figma's collaborative feature allowed designers to make changes live, and seeing these little bubbles flying around, making shapes, and rearranging layouts on the go was like real magic to clients.

Download Figma or fire it up!

First up, if you haven't created an account yet, go to Figma.com and sign up for the free account plan. You can download the installation files and work with Figma on your local device or use the web-browser client. Either way works just fine. Log into your account and let's start exploring what Figma has to offer.



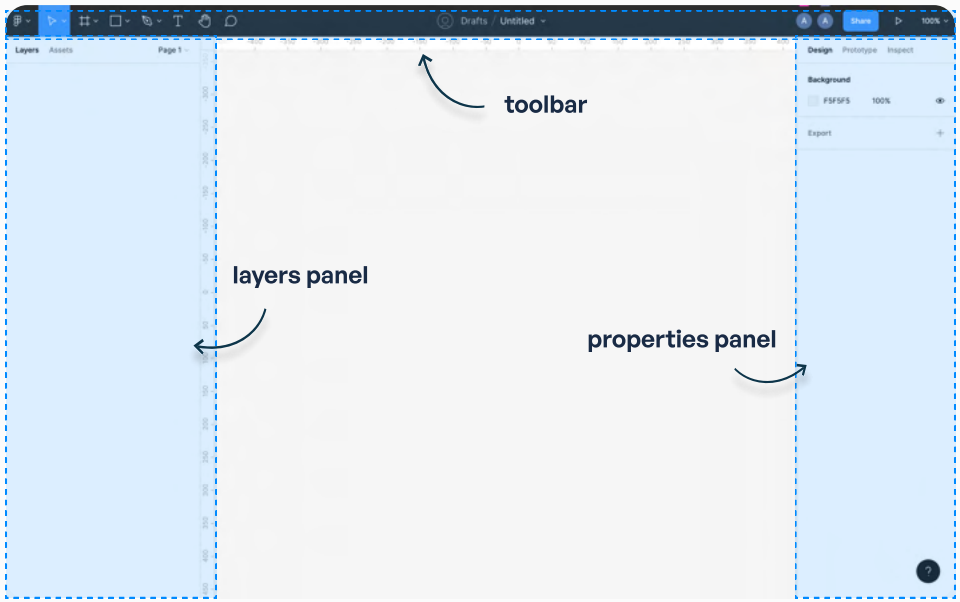
Once you've logged in, you'll be redirected to the Figma client dashboard. This is your design hub. It's where you create new files, import files, create drafts, or search through existing projects.

Important note: From now on, whenever I'm talking about shortcuts. I'll be referring to Macbook keyboard shortcuts. For Windows devices, the equivalent of **Opt is Alt**, and **Cmd is Ctrl**,



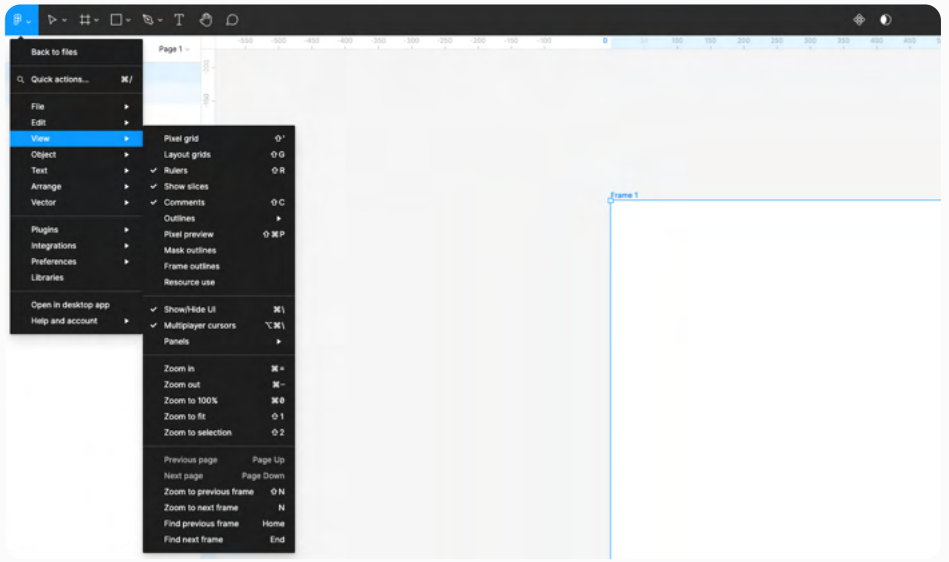
The Interface

Most design tools look somewhat similar. We usually have our “Layers” panel on the left and our “Properties” panel on the right with our “Toolbar” being at the top. After creating a new file, the first screen you end up on is called the “Editor.” You’re probably familiar with that, though.

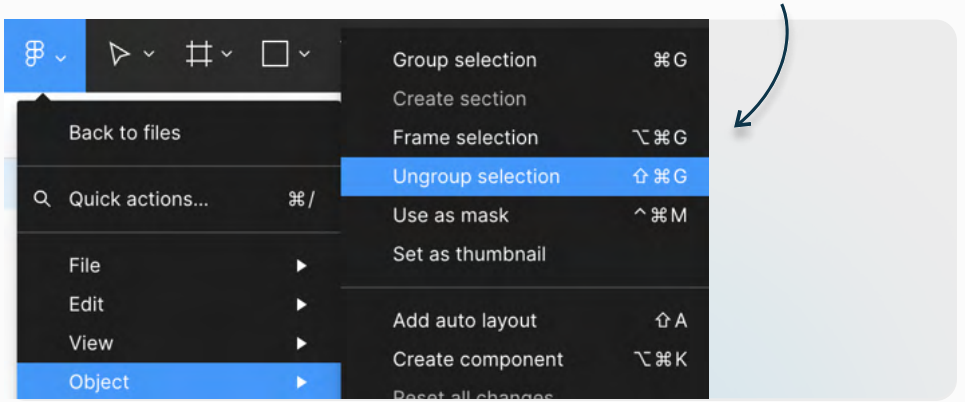


The toolbar is where all your shapes, tools, and menus are hidden. You can set your preferences here, enable grids, and guidelines, save and export files, access plugins, libraries, assets, account settings, shape tools, and customizations. We’ll quickly cover each of the main tools we will use in this workbook. To explore the rest of the toolbar functions - visit the official Figma site or play around with it yourself.

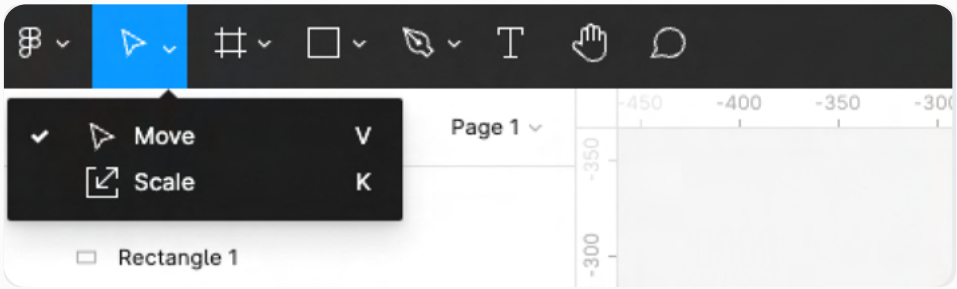
Figma menu



Under the Figma icon, you’ll find all the preferences, plugins, libraries, and customization settings. We’ll be mainly using shortcuts to switch between our views and manage our tools, but if you ever find yourself lost, just open it up and look through different options. You’ll have all the keyboard shortcuts displayed on the label's, to the right.

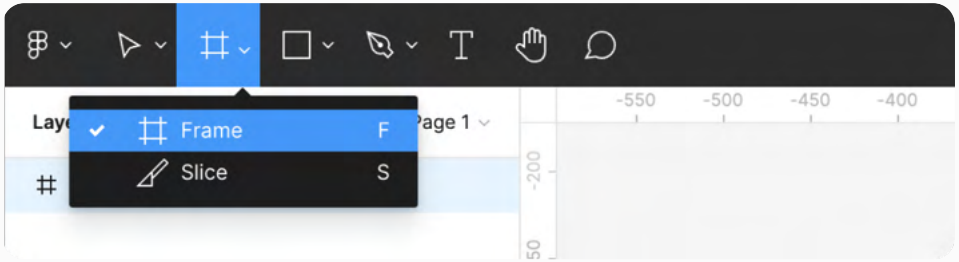


Move and scale



Whenever you just want to move around the editor, click **“V,”** and you’ll pull up your regular viewing mode. We’ll be using **Scale (K)** to edit the size of our components without messing up their ratio.

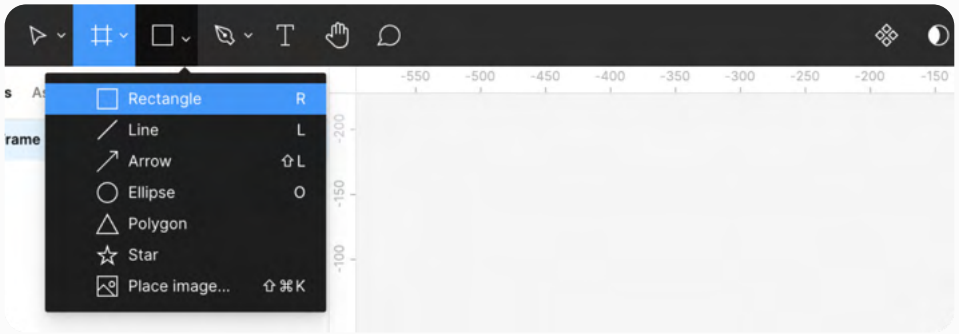
Frame



Frame tool is used to create new artboards. You can choose from a selection of pre-made templates or create one manually. The slice tool is used to specify a specific region of the screen for export, even if they're not organized into a single group.

To create a new frame, we must click **"F."** To frame our selection, we'll have to click: **"Opt + Cmd + G."**

Shape tool



With the help of a shape tool, we can create basic shapes like rectangles, polygons, stars, and lines. Our shortcuts are:

R for rectangle, **L for line**, **O for the eclipse**.

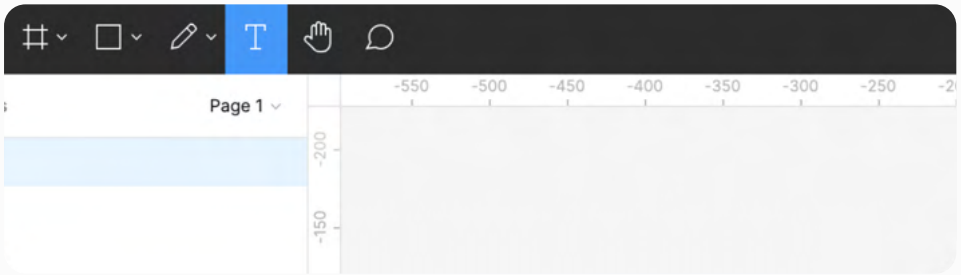
Pen tool



If you’ve ever used Illustrator or Photoshop, you’re probably familiar with a pen tool. With the pen tool, we can create custom shapes using vector anchor points. The pencil tool is used to create your shapes more freely. The cool thing is that Figma will automatically fix your shape and make the lines you’re drawing much smoother.

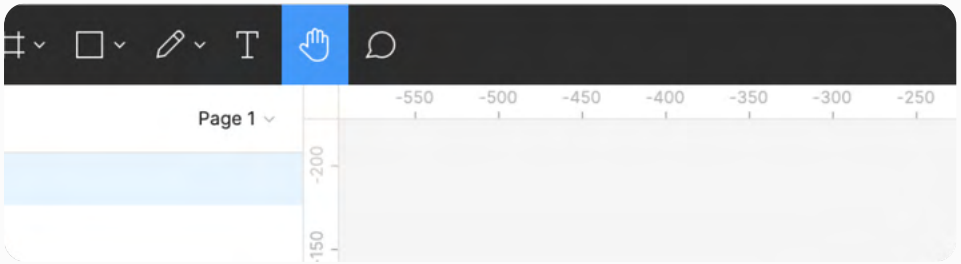
To activate it with a shortcut, click **“P.”**

Type tool



Type tool is used to add text layers to your designs. Our shortcut to pull up the Type tool is **“T.”**

Hand tool

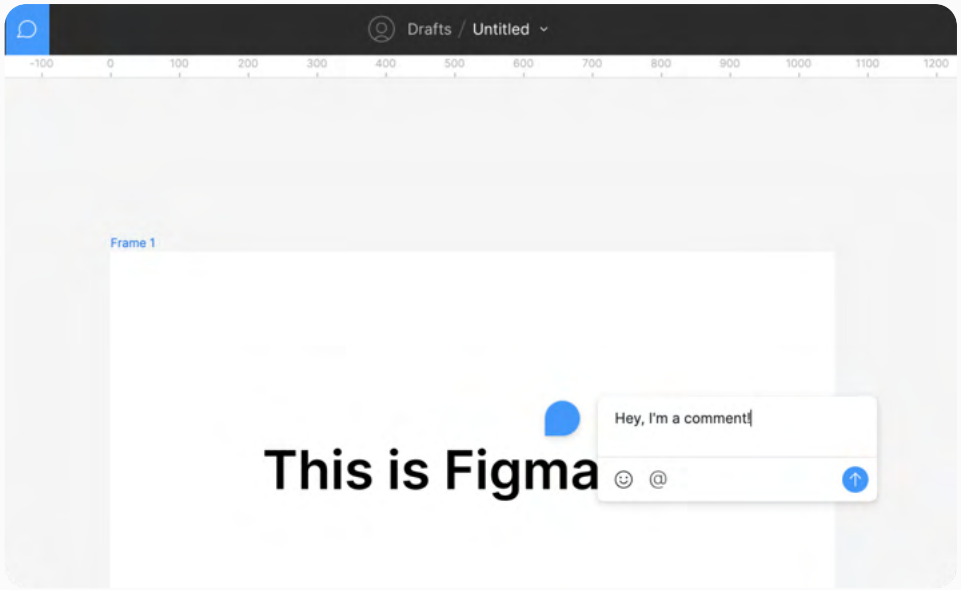


The hand tool is pretty handy when you want to navigate around your design file without accidentally clicking on any layers or dragging up anchor points. However, you can simply hold the spacebar on your keyboard and click and drag your mouse around to do the same.

Comment tool

The comment tool makes it possible for you and other users invited to the file to leave comments on any place on the file and start a new conversation related to them.

To start commenting on a file, simply click **“C”** on your keyboard.

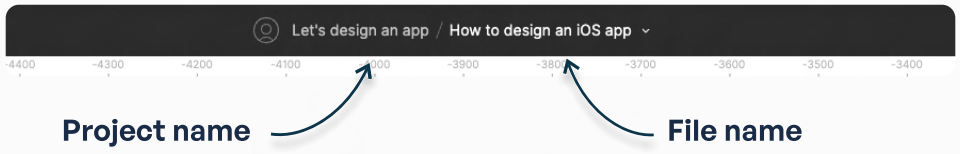


Profile, Share, Preview, Zoom

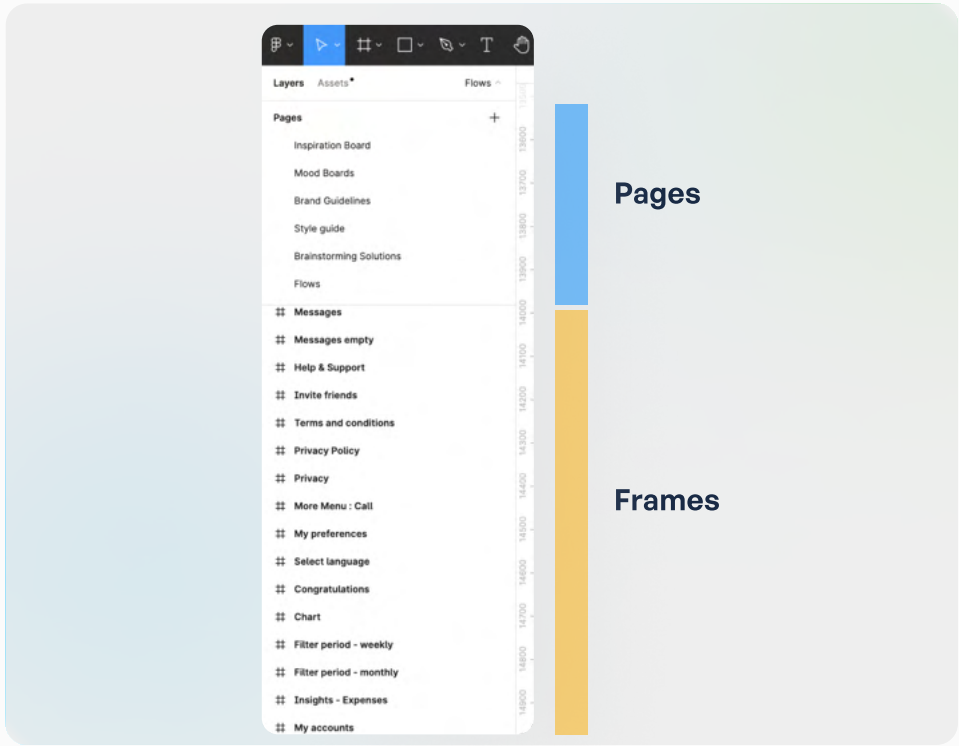


On the right, you can access your profile settings by clicking on the avatar, sharing your file with others, and previewing your designs/ prototypes. You have your zoom, grids, and rules on the far right.

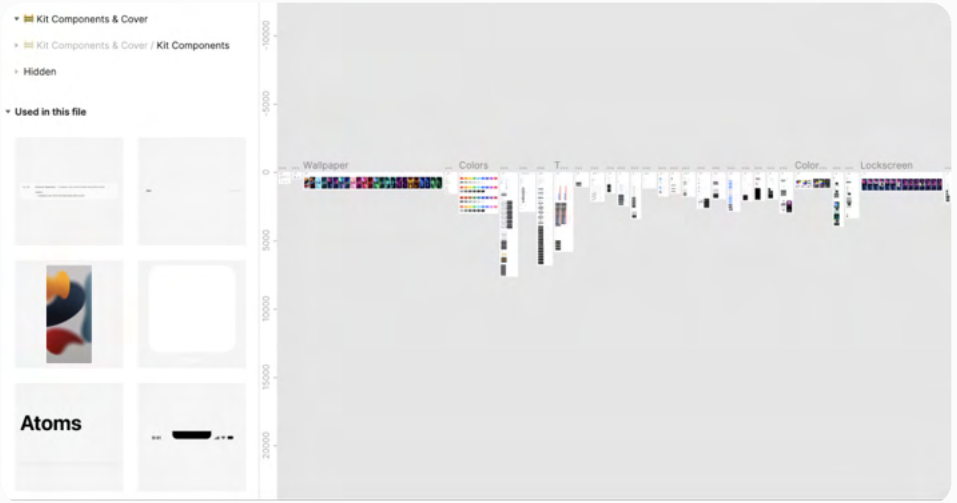
Project names and hierarchy



There's a pretty simple hierarchy of your files. First, you have the project name, in which you're able to have multiple files. Your file is where you're designing. Then inside your file, you can have numerous pages, and within these pages, you can have as many artboards as you want or need.



Assets



On our Layers panel, you can also find a tab called Assets. Assets are the elements you save as Components. Components are reusable UI elements that include buttons, forms, navigations, cards, headers, modals, overlays, etc. You'll learn how to create components and re-use your assets to streamline your work in the Practical part.

That's pretty much it for this part. We'll return to Figma shortcuts, grids, and components in later chapters when working on replicating the designs and components, so keep reading!

It's time to cover auto layout and constraints now :)

Auto layout

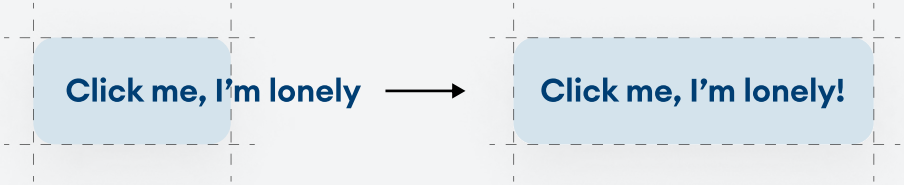
9. Auto layout 4.0

When we had around 50% of this e-book created and our Fintech app was already designed, Figma decided it was about to release a new auto-layout update (😅). This meant that I had to rewrite the entire section again... I couldn't complain because the new features were even better and much simpler to use than before.

Let's talk about auto layout and how to use it correctly in your designs.

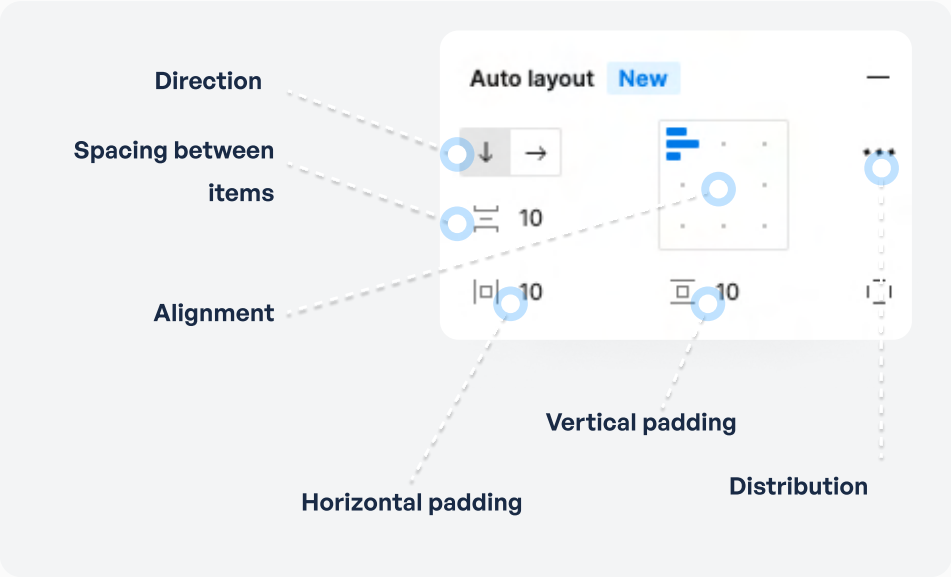
What exactly is auto layout?

Auto layout is a dynamic property you can add to your frames and components. It lets you create designs that grow or shrink to fit your container and readjust their size as their contents change. This is super helpful when you need to add new layers, make room for longer text strings, or maintain one alignment as your designs evolve.



Practice auto layout with Figma

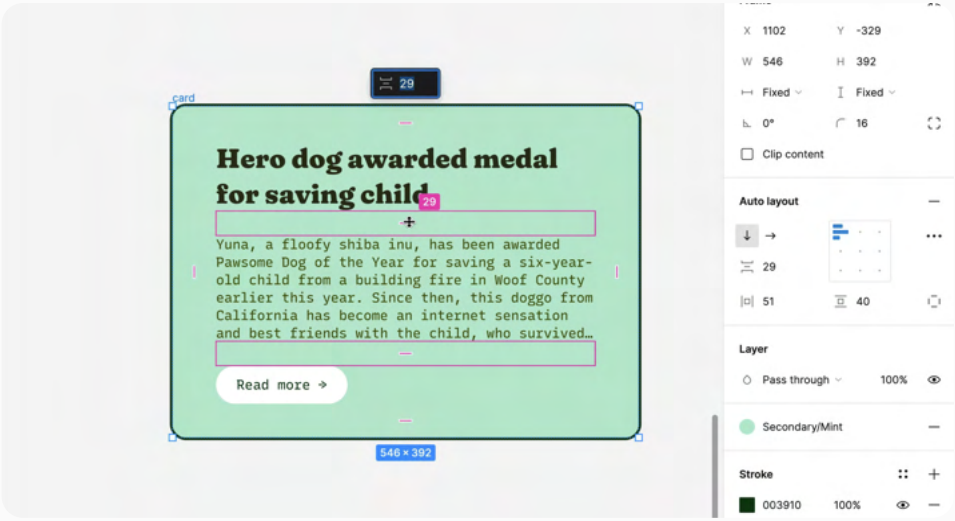
Frames with auto layouts have entirely different properties than regular frames. When you apply an auto layout, changes appear in the right sidebar panel. Let's see what these properties mean.



Direction describes the way auto layout elements align within the flow:

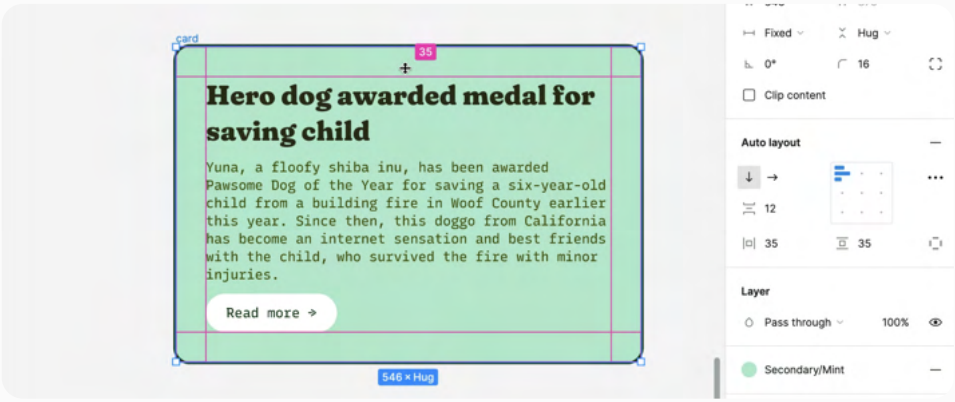
- Choose a **vertical** direction to align your objects within an auto layout container along the y-axis. Example: a list or a timeline.
- Choose a **horizontal** direction to align your objects within an auto layout container along the x-axis. Example, row of buttons, icons, or a tab bar menu.

Spacing between elements dictates what will be the amount of margin left between two elements within the auto layout container.



Source: Figma.com

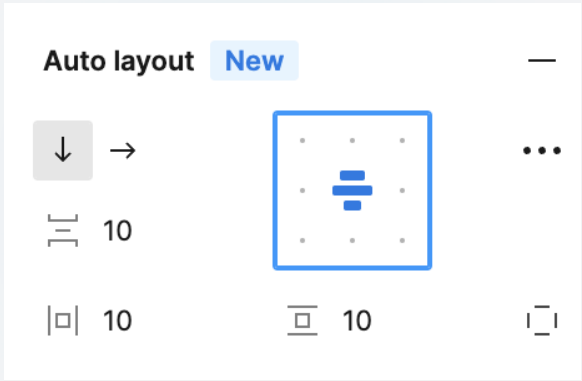
Padding controls the space between the boundary of the auto layout frame and its child containers. You can set padding values all at once or edit their values on each axis, horizontally or vertically.



Source: Figma.com

Alignment lets you choose how to align child objects within an auto layout frame. When you want to align your auto layout selection group to the bottom, top, left, or correct, look at the alignment and padding property on your properties' panel on the right. Unlike objects in a regular frame, you can't control the alignment of the objects individually. For that reason, you set the alignment of the child objects on the parent auto layout frame, similar to how it works in web development. Children inherit their parent's properties.

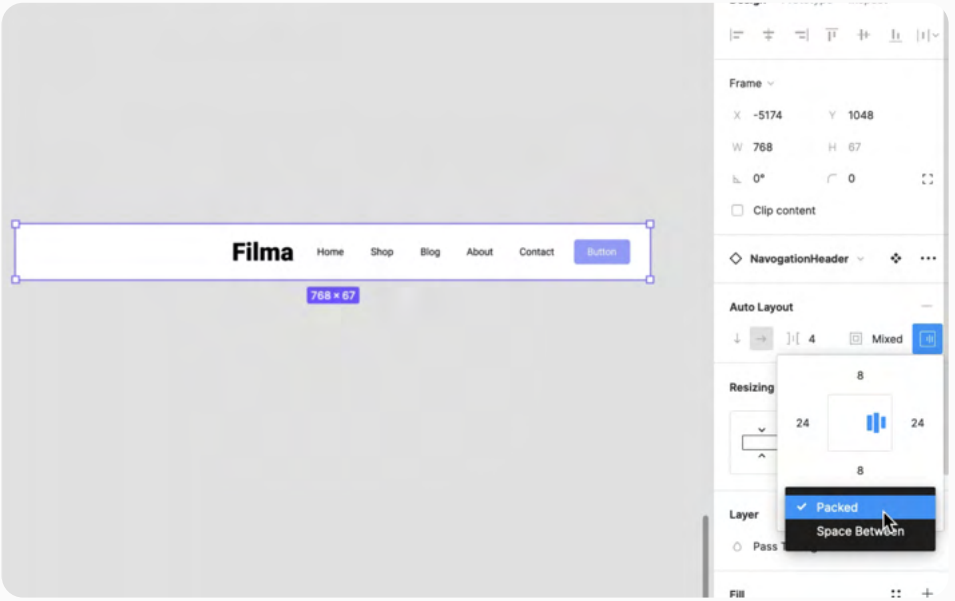
Use the interactive grid to select nine layout options for the children in a frame.



Distribution property let's you set how objects in an auto layout frame will be distributed within that frame.

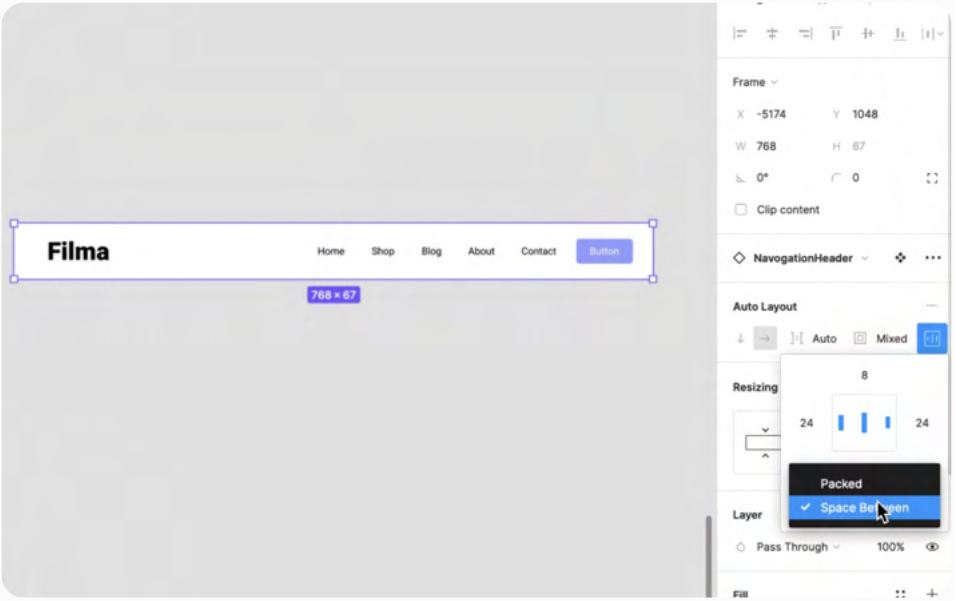
Click the three dots ⋯ to open advanced layout settings. Next, go to Spacing mode, use the dropdown to choose:

Packed: Objects in a frame will be grouped together. Use this option to keep objects in a frame as close as possible and aligned together.



Packed distribution is best used for items that need to align to either side of the container, like web navigation items or a list of icons.

Space between: Objects in a frame have the space between them equally distributed along the direction and alignment set for the frame. Use this option to stretch objects across a frame.



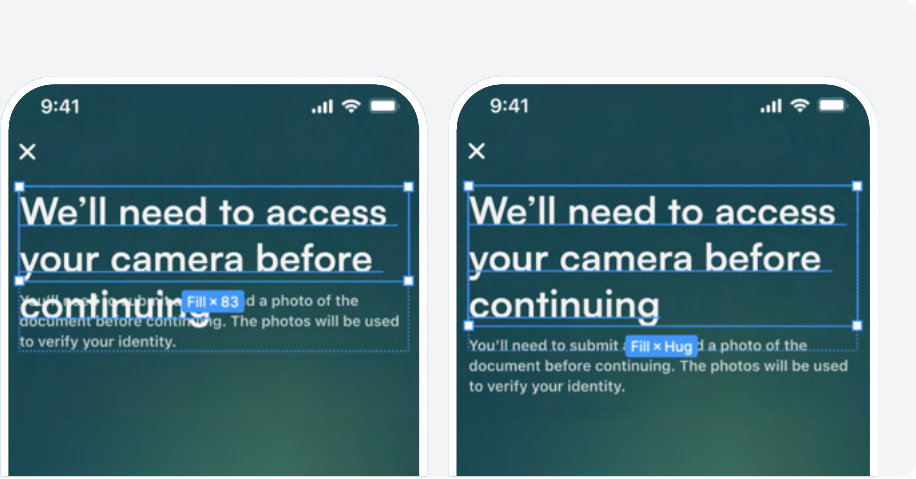
Resizing auto layout containers

One of the biggest strengths of auto layout is its ability to adjust the dimensions of the objects in an Auto layout frame, depending on their resizing properties set for each child element and parent container.

You can choose the resizing behavior for parent auto-layout frames to adapt to any changes made to their children's elements. Resizing settings can be applied for objects on both axes individually using the dropdown menus in the right sidebar.

Resizing text layers

If you want your auto layout frames to be completely fluid, we need to apply the fill width property to our text layers. Fixed-size text layers won't resize to accommodate our text changes, which will cause overlap between layers in an auto layout frame. We'll talk about that later when we're designing our Fintech app.

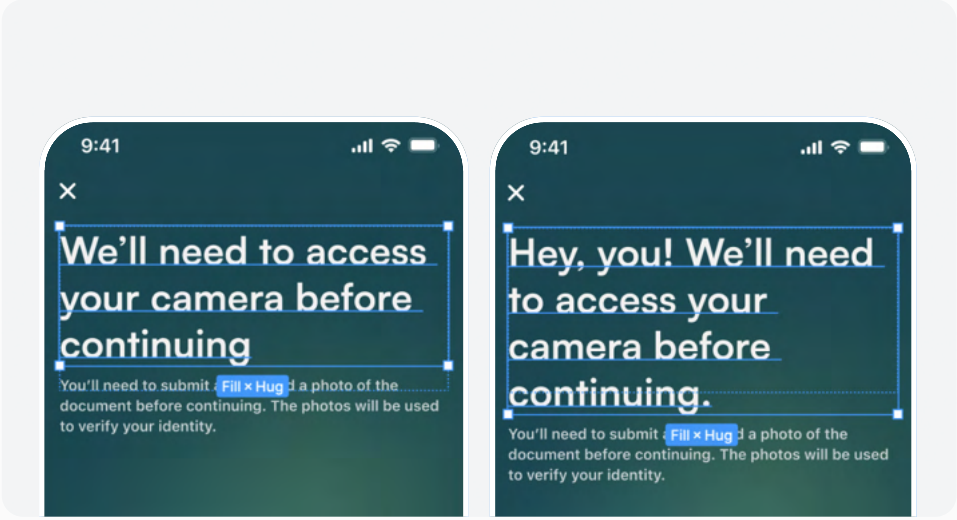


Fixed width and height

If we set our auto layout frame to Fixed width or height, the frame's dimensions will remain the same no matter the size of the content we place within them. The size of the frame won't react to changes in the objects within them. Fixed dimensions are helpful to set constraints for content like images, avatars, and vectors.

Hug content

Set it to hug contents if you want your auto layout frame to resize itself according to its child objects. Our container will keep the smallest possible dimensions to surround the objects within it while respecting the padding values set simultaneously.



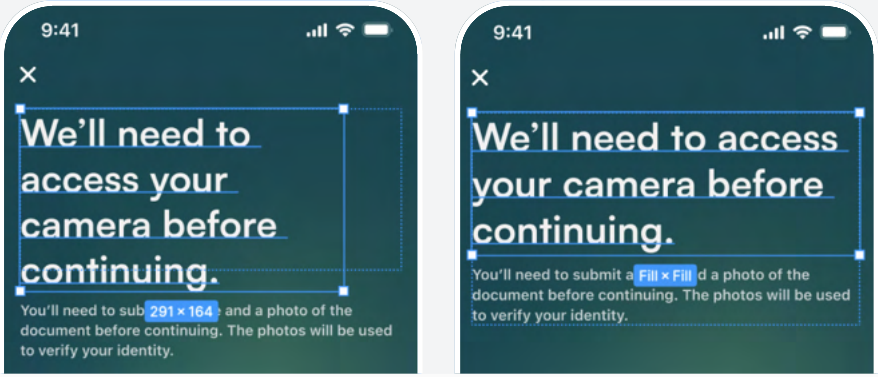
I often use the Hug content property for text layer containers when text layers are put into one auto layout frame, readjust to each other's lengths and align themselves within the container.

If it all sounds confusing at this point, don't worry! It takes a ton of practice to understand auto layout and constraints, and we'll practice a lot in the Practical part of this e-book. You can also check out the official Figma guide on learning auto layout here.



Fill container

Every object in an auto layout container set to Fill container stretches out to the maximum width and/or height of their parent frame.



You should always try to set either fill or hug container properties on your auto layout elements. Why? Because we'll need to work on constraining our designs so they remain responsive whenever we want to change the dimensions of our frame.

You can read more about constraints in the following chapter.

Constraints

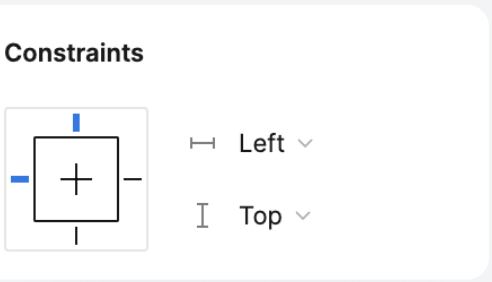
10. What are constraints in Figma?

By applying constraints to your auto layout frames, you decide how your frame will respond when being resized. One important note to mention here - you can't apply constraints to child objects in an auto-layout frame unless the object has absolute position ↓ enabled.

Instead, you can use the resizing property to define how objects respond as the frame or the objects in the frame are being resized.

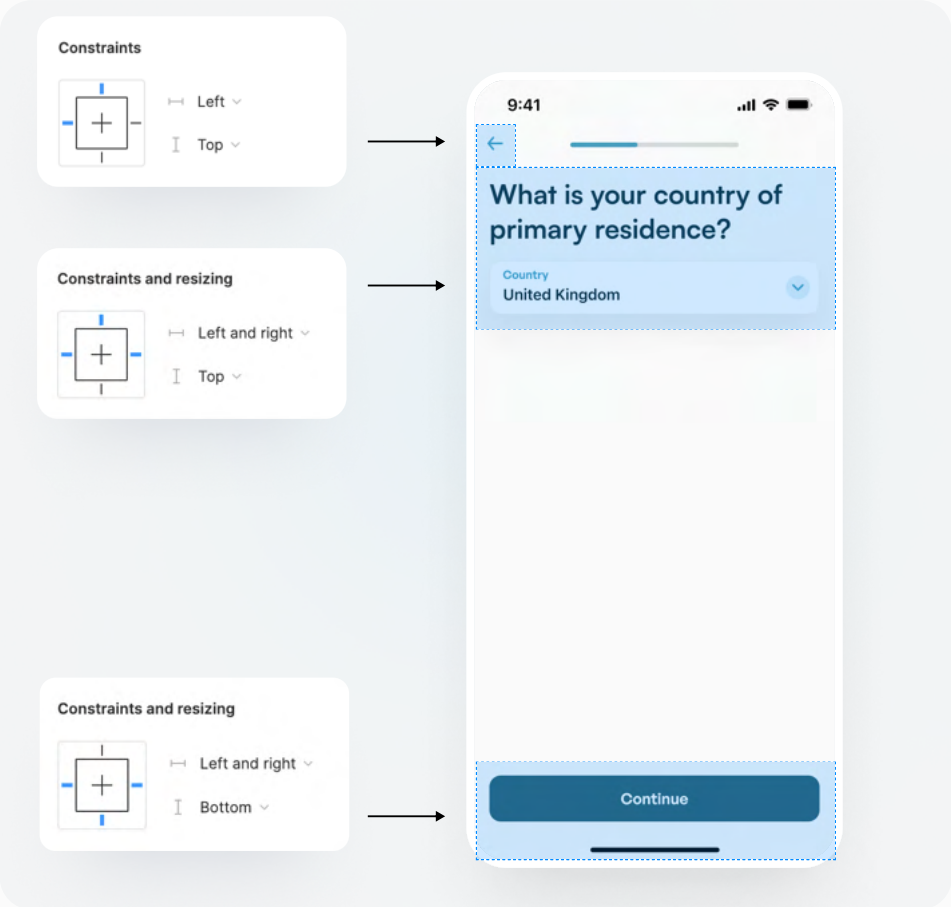
You can still apply constraints to the auto layout frame if it's nested within a regular frame. You sometimes want to do that for complex groups like overlapping vector files you want to keep at one dimension.

The Constraints section and resizing options can be found on your properties panel on the right, above the layer settings, allowing you to set both the constraints for the auto layout frames and the resizing behavior for any objects within it.



Practical use of constraints

There's one thing to mention before we discuss the types of constraints and how they work with our designs. We will be horizontally constraining 90% of our elements by setting the position fixed → **left and right**, or to the **left** or **right**, respectively, for objects that need to remain in one place regardless of the screen size, for example, a back button or a tool icon. See the example below.



Why would you want to constrain it like that? You have to think about how you want to make your content align. Let's take our example.

The back button should always align to the left and top, staying in one position and retaining the same margin from the left and the same distance from the status bar.

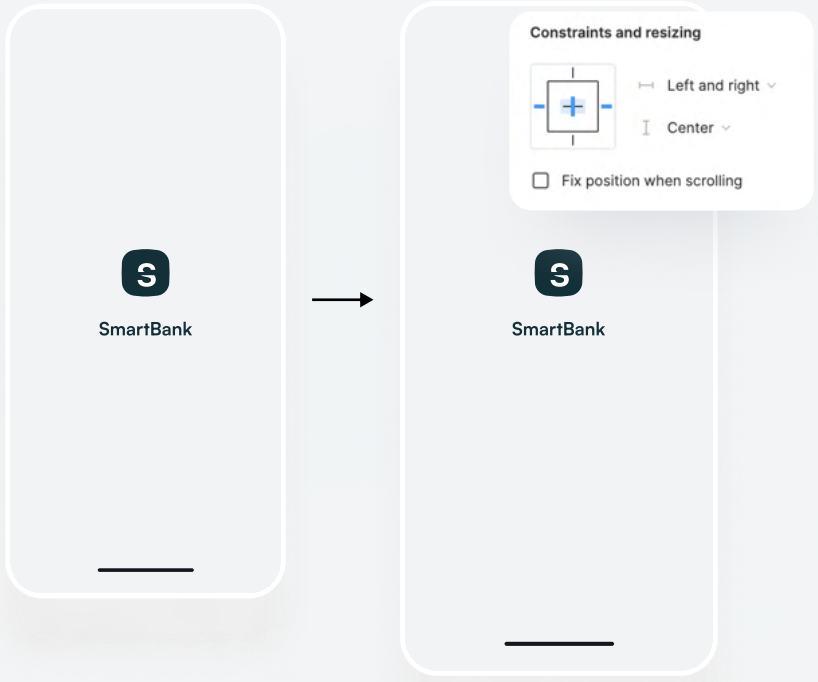
Our main content, the title, description, input field, CTA button, and our link button should always stick to the left and right in a full-width container set to fit our screen and have around 8 px distance from the back button. It should also stick to the top since we want to have our app's same repeatable navigational design pattern.

Lastly, our "Continue" button container should also stick to the left and right, but this time we want to have it aligned to the bottom so it always stays there no matter the screen size.

The same concept should apply to every screen. If you're not sure how to correctly position and constrain your elements, open your iPhone and look at Apple Music or App Store - you won't find better examples of iOS apps than what Apple's doing themselves.

Types of constraints

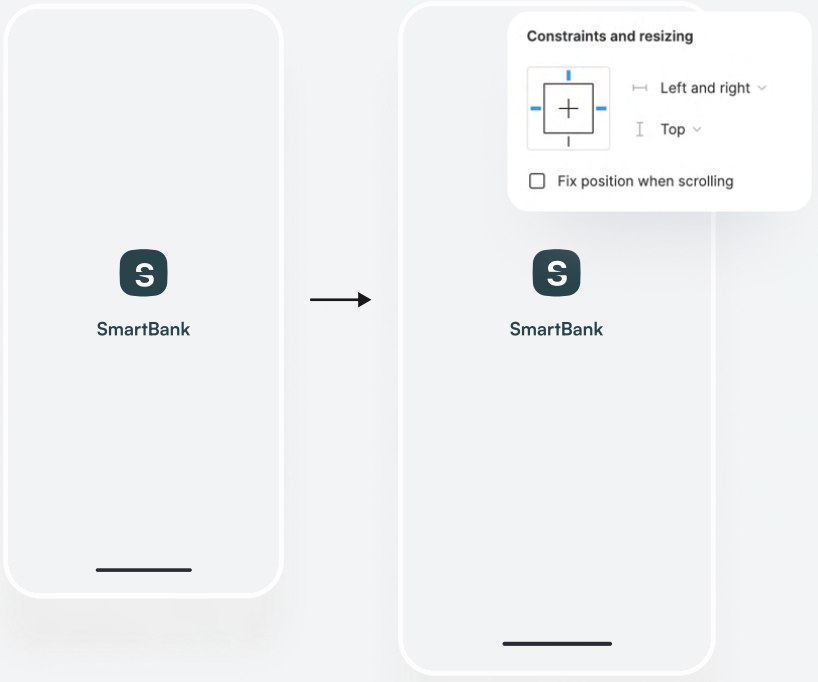
We'll be testing if our constraints are all properly set and working by changing the size of the frame we're designing on - from iPhone 13 Mini to Pro Max. This is how each of the constrained elements would behave. Our middle container is an auto layout frame consisting of a logo mark with the text layer below.



Constraint = Left and right + Center

This constraint will keep your object-centered and retain the same top and bottom spacing between two vertical ends of your frame.

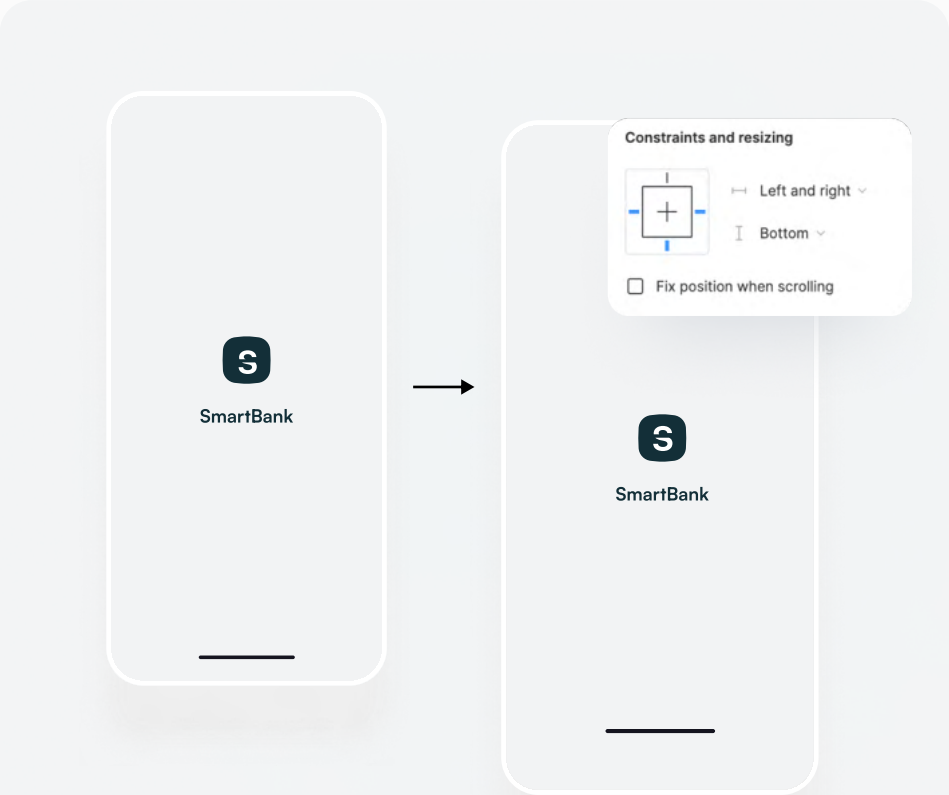
It works well if you want your illustrations and images aligned in the middle of your screen, and the rest of your content sticks to the top and/or bottom.



Constraint = Left and right + Top

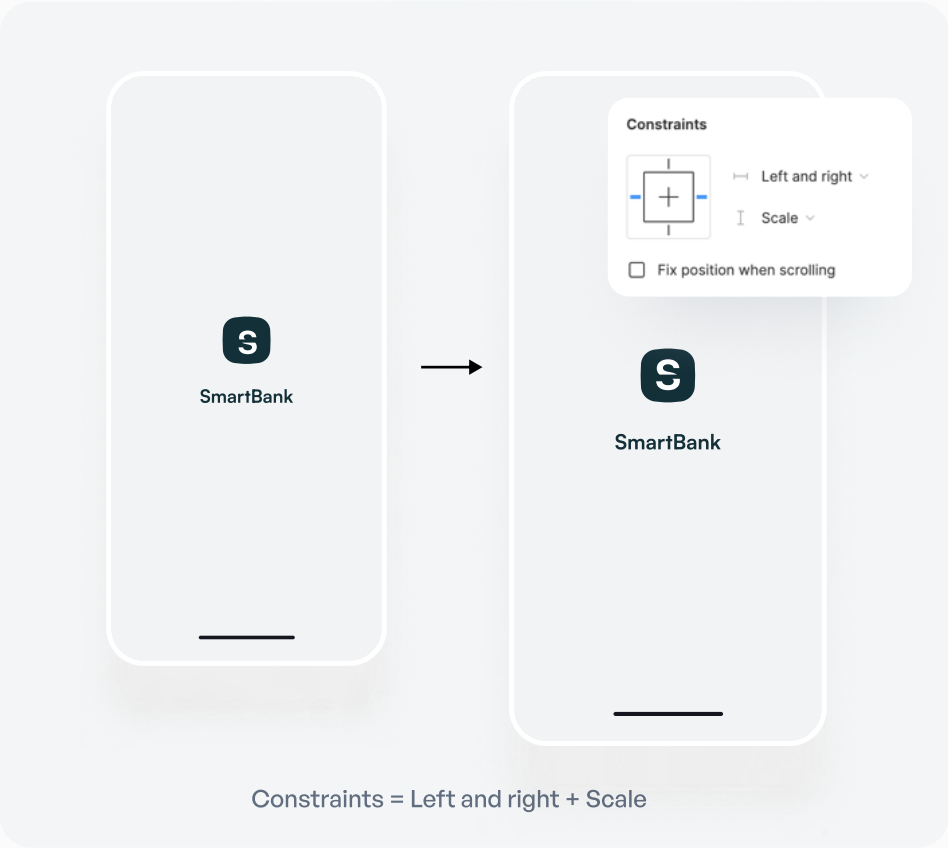
This constraint will keep your selection in one vertical position regardless of the screen size. It's essential to keep your content in one place with text and description containers. Your design remains consistent, and there's a straightforward, repeatable design pattern across all your app screens.

We'll cover that in detail in later chapters when discussing our design.



Constraints = Left and right + Bottom

The bottom constraint will make your components stick to the bottom edge of the screen, and it's best used for elements like buttons, home indicators, and tab bars to retain the same position regardless of the screen size.



Constraints = Left and right + Scale

The last type of constraint scales your content along with your screen size. It might have its use for images and illustration-heavy splash screens but might cause some text and button layers issues. I usually use this with caution, mainly for the images and circular shapes.

Fintech App

Your Fintech UI Kit

11. Fintech UI Kit

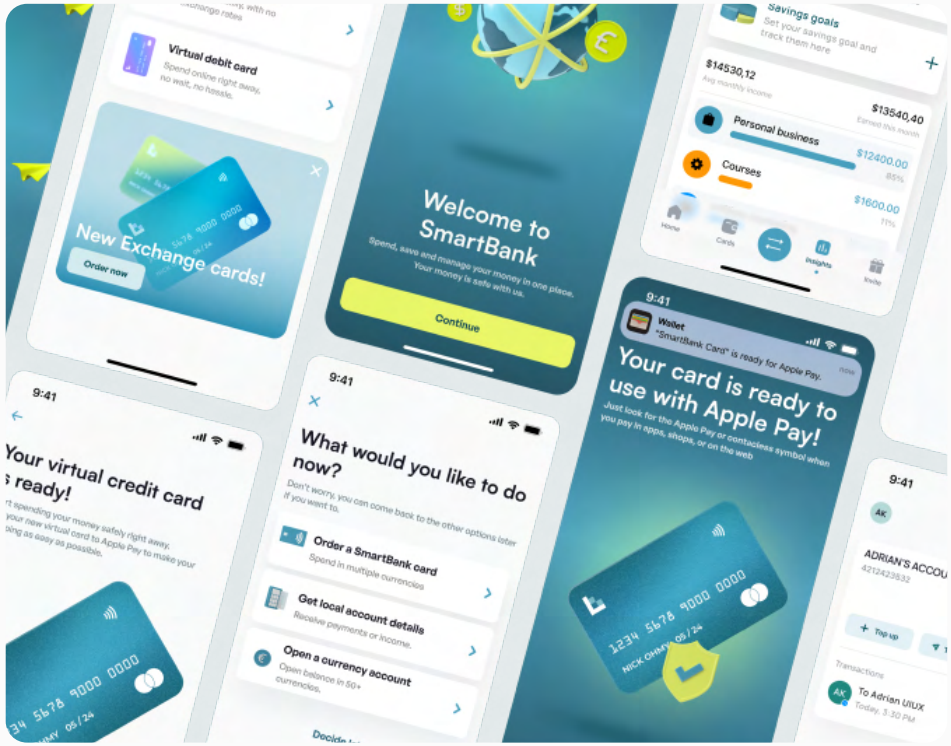
Before we dive in, to show you the exact process we took to design this entire iOS app, let me tell you a bit about the product, how you should approach this UI kit, and what the subsequent releases and updates will be about.

We (Natalie and I) designed this app to show you first-hand what a standard iOS app design process could look like if we approached it as a design agency. There will be certain stages that we put less focus on, and that's because it would be practically impossible to fit the entire UX and UI process into one book and describe it without you getting lost. Soon, we will release a separate product on UX alone, or we might even include entirely new chapters to this e-book, so you get the complete picture.

While designing this UI Kit, we assumed many things, and there are by no means the best practices when creating the user experience of a fintech app. We took a lot of inspiration from the existing fintech apps like Wise, Monzo, Revolut, and a few big commercial banks, as well as my recent work on UK's biggest government-owned savings bank app. We will show you all the UX stages and best practices we follow when working on a new product and share with you all the templates we use to work through these stages even though we're not going to cover them in much detail.

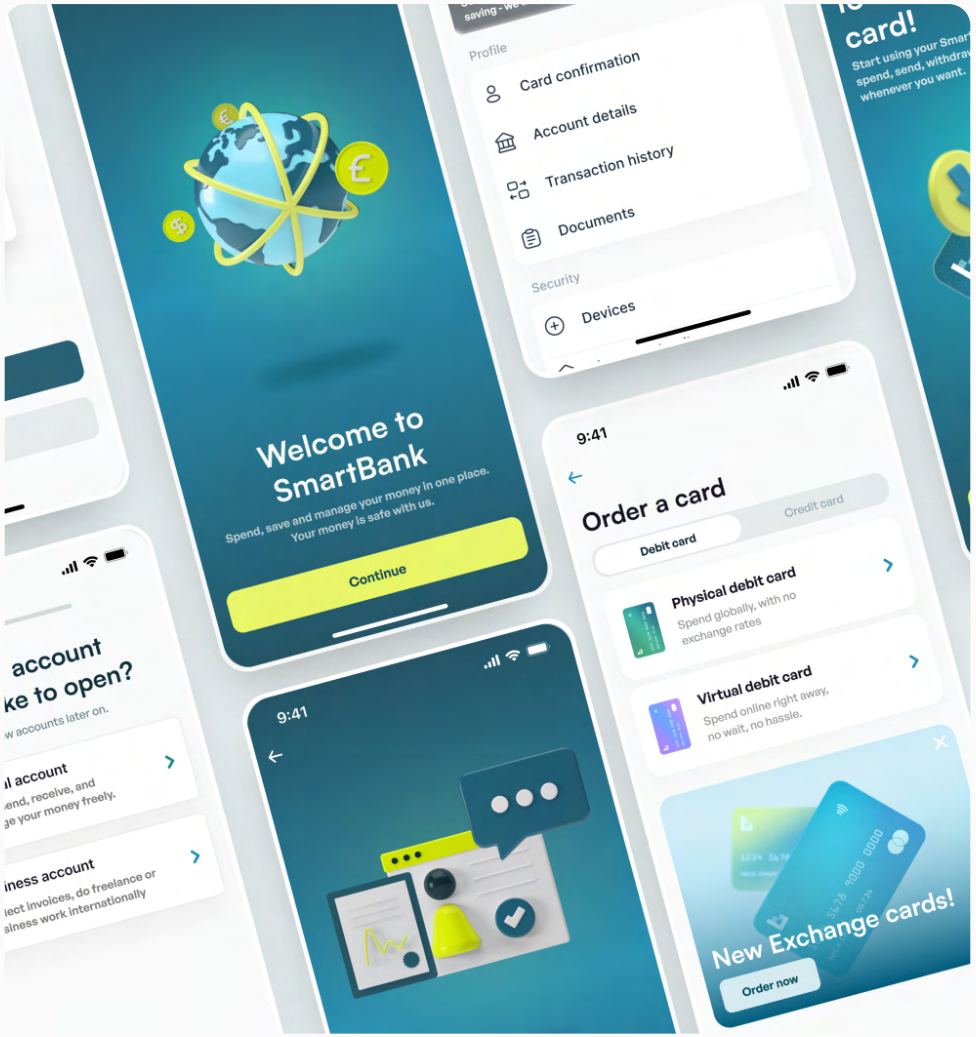
The main goal of this e-book is to show you the best UI practices when designing apps for iOS, but we just couldn't leave out UX entirely.

We also want you to know that along with the new updates this book will be getting, we will also keep on updating the UI kit. We plan to include at least 9 more chapters, refresh and fix the designs for each release, add a working prototype and include a brand new and modern dark mode design. It's all coming, and you'll get free updates whenever we release them, so make sure you stay subscribed to the Gumroad newsletter and follow my daily stories on IG.



To sum up, right now, the UI Kit consists of 100+ screens, 20 flows, UX templates, UI templates, and a practical guide on how to design 15 out of 24 existing flows. New updates are coming soon.

Without further ado, let's dive into the UX behind our Fintech App.



The Fintech project

12. Intro to the project

In this book, we will design a fintech (banking) app. When choosing a project, I wanted to give you the best example of what a standard app would include. A while ago, I posted a story on my Instagram asking about what kind of apps you'd like me to design, and out of 4 suggested, Fintech emerged as a winner, with more than 30% of you voting for it.

Luckily, with Fintech, we will cover almost every design pattern from onboarding through account setup, verifications, notifications, search, analytics, accounts, cards, profile settings, splash screens, intro screens, and welcome screens.

We will design all the above flows (and more!) for a finance app. What that means for us exactly is - the welcome and onboarding screens, home screen, transactions, managing cards, recent activities, insights, languages, notifications, currency exchange, profile, and a lot more. The remaining flows will be discussed in later iterations. For now, though, we'll cover 15 out of 24 planned.

In later releases, we plan to launch a dark mode UI and a working animated prototype, so stay put; it's coming soon!

For now, though, let's discuss our project in detail.

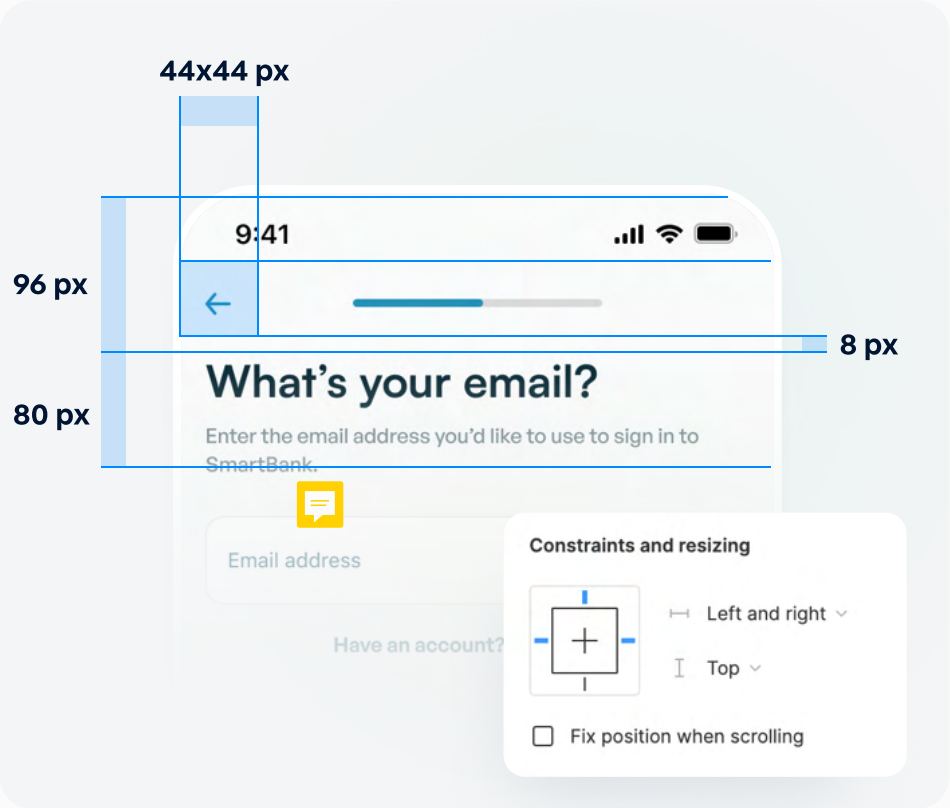
Fintech is a massive market with huge design potential. We're confident that this app project will serve you best as a reference work to practice your design skills and visual pattern recognition. When you finish the practical part of this e-book and start applying the knowledge you're going to obtain to your future concepts, you'll improve your designs in no time. You'll be able to take all the iOS design patterns we'll create in this e-book and apply them to apps from all niches to make something great yourself.

One important thing I have to highlight again - this is an e-book that, by default, should teach you how to design better user interfaces by applying the best design practices and following the most straightforward and standard design process. We will not focus that much on UX research, analysis, and ideation. While UX design typically consists of around 60-70% of the whole product design process, it is too big to fit in the same e-book, and it will be a topic for a different release or a separate video course.

However, we will go over the primary user research to understand what needs to be done and show you a general overview of how you should approach the UX stages on a real-life commercial project. We'll assume many things to streamline our design process.

We'll create a list of functionalities a finance app should have and go through simplified competitors' research and market analysis.

We'll analyze what's working for them and what are our hypothetical client wants and needs when using our app. We'll follow our UI design framework and start creating high-fidelity wireframes and analyzing the layout and structure of each screen.



It's time to learn what our project is about. Scroll down to the next page to read our project brief. See you there!

Project brief

13. Project brief

When designing our app, security should be our top priority. Financial data is susceptible; therefore, we need to think about two-factor authentication, PIN numbers, security questions, and biometrics as additional forms of validating user personality and allowing access to sensitive information.

We'll also need to design a live currency converter. We're building this app for active young people. Having an automatic currency conversion built-in app would be a fantastic addition when traveling abroad and paying for expenses in foreign currency without being penalized with high conversion rates.

To give you more content to work and practice with, we decided we want to show you how to design the core functionalities of a fintech app first and design our virtual currency function, and all the supplementary flows in our second release.

Our MVP Objectives are as follows:

- Users can top up, receive and send money in different currencies.
- Users can create and order new credit/debit cards and use them for payments
- Users can create currency accounts, manage them, and view their previous transactions

Our core user stories:

- As a user, I want to be able to receive and send money in multiple currencies.
- As a user, I need to be able to create new cards and pay for my expenses in multiple currencies.
- As a user, I want to be able to look at my current month's spending and savings.
- As a user, I want to top up my account using my credit card or copying the bank account details and making a manual deposit.
- As a user, I want to set reminders for my upcoming spending and set automatic transfers to my savings account every 10th day of the month.
- As a user, I want to be able to quickly send money to myself, my friends, or my family members.
- As a user, I want to be able to choose the credit card I want and order a virtual one.
- As a user, I want to be able to edit my profile settings, add a profile picture, and edit my app preferences.
- As a user, I want to open currency accounts and move my money freely between them.

These are our MVP's core objectives that our theoretical stakeholders expect us to include in our designs. Our goal is to create an iOS personal finance app that enables our users to do the above.

Lean UX

our approach

14. Our UX approach

Our approach to UX will be based on a Lean UX methodology where small changes are made upfront and tested with users right away, using low-high fidelity prototypes until an optimal solution is obtained. We'll focus less on traditional UX deliverables to streamline our work process and achieve business objectives in the least amount of time.

We'll use problem statements from our market audit and create assumptions that will help us develop hypotheses.

These assumptions are usually generated in a UX workshop as a team with the client's stakeholders, where the problem is shown, and ideas are brainstormed to solve that problem.

Lean UX

The Lean UX Cycle contains three phases: Thinking, Making, and Checking/Testing.

1. **Thinking:** Instead of knowing precisely what you want to design, you start with something that could work, so you create a hypothesis and then figure out how to test it quickly, then you build on top of that.

2. **Building:** The smallest element that can be built will help us determine whether our theory is correct. It could be Wireframes, mockups, or working prototypes.

3. **Testing:** During this phase, we test what we've built (our MVP) with real-life consumers, observe them while they use our products, and get feedback from them.

This phase aims to determine whether a feature/product is likely to succeed before investing time and money to fully develop it.

Why Lean UX and not Design Thinking approach?

If given more time and resources, we'd probably use the Design-Thinking approach, but it can be quite time-consuming. It focuses more on the deliverables after each step. It can also be more expensive because of the exercises performed in the research and testing phase, such as usability testing, which can cost tens of thousands of dollars and take several weeks to complete.

What are the differences and similarities between the Design Thinking and Lean UX approaches?

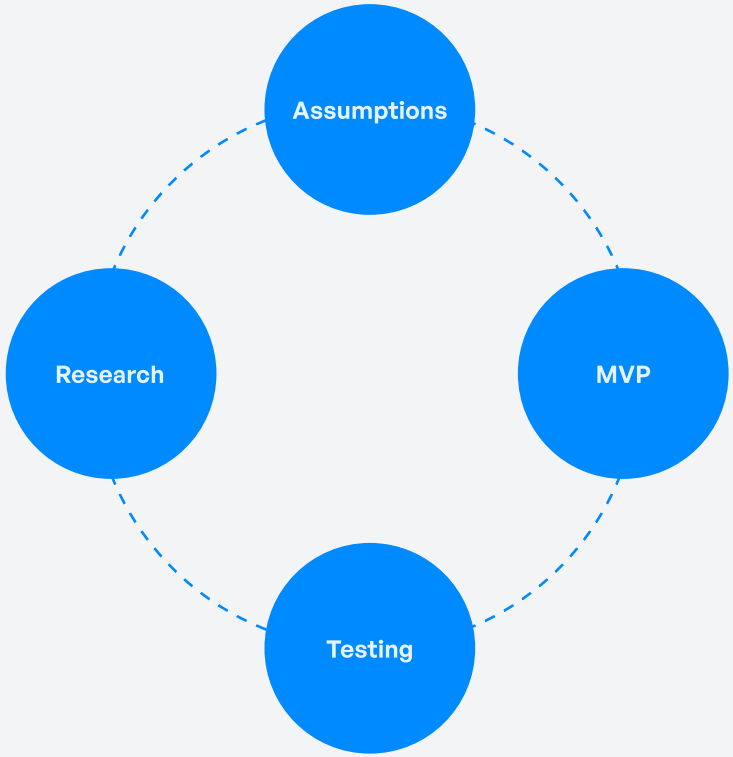
Similarities: Both processes are focused on the user with the user-centric approach and are Iterative. Continuous evaluation of findings at each decision point is what makes them similar.

Differences: The most notable distinction between these two processes is that the lean process focuses on delivering maximum value with the least amount of waste of resources, such as time, money, and people, by focusing on short and iterative cycles.

How is Lean UX different from traditional UX

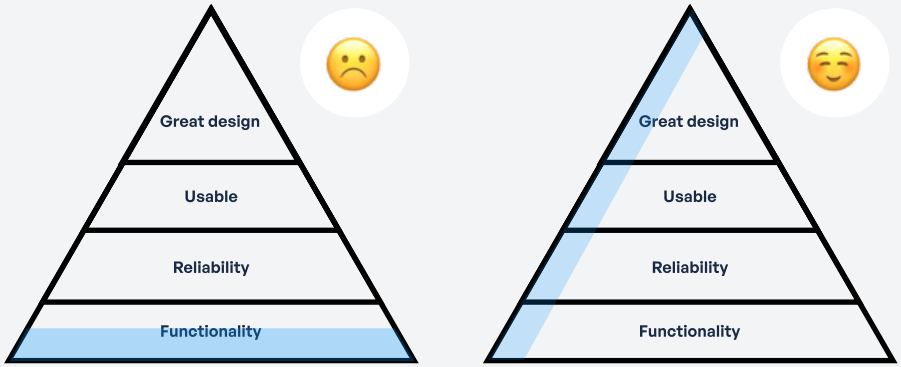
In traditional UX our project would be focused on fully documenting the deliverables at each stages to be as detailed as possible based on the initial requirements.

In Lean UX however, we aren't focused as much on the deliverables as much as getting the product released to the market as soon as possible and starting our chain of improvements here and now, so we can iterate and improve based on real user feedback.



We are not fixed on the requirements rather on creating a "problem statement" that should lead us to make a set of assumptions that we'll then use to create hypothesis.These hypothesis are usually created during a UX workshop with the stakeholders. As a team, you're makign assumptions and coming up with problem statements, then the team starts the brainstorm session focused on solving the problems stated.

Once we define all problems, we can look for suitable solutions. Based on the assumptions, and a couple of internal sprints, we design our first MVP. The Minimum Viable Product (MVP) is a fundamental concept in Lean UX. The idea of a MVP is to build the most basic version of the concept as possible as soon as possible, test it and if there are no valuable results, drop the idea, or iterate and adjust the product for the next round of testing.



Wrong way to do the MVP

Right way to do the MVP

Design Framework

overview

15. Our design framework overview

We know what market needs we're trying to fill now. We know what features we're going to release. It's time to start our design process and move through the UX phases of creating an ideal user persona. We will apply it to our user journeys and map them all out, build our information architecture, create our user flows, and then audit the UX of our competitor's screens. We'll brainstorm the solutions, design our first low-fidelity sketches, and suppose we end our UX phase with mid-fidelity wireframes. When we have our wireframes built out, we'll search the web for visual inspirations, create an inspiration board, organize our designs into groups to create mood boards, start designing our brand identity guidelines, and develop our style guide with a basic design system. Then we're on to apply our brand visuals, typography, and imagery to our wireframes, componentizing our elements to start building our congruent, reusable, and organized components library. That's the part we'll land on. We skip the low-fidelity wireframing process to save you time and show you the end results, as this is more UX than UI.

We'll create our first prototype and test it with our user base when all that is over. Get a round of feedback, iterate, improve, run it once more, test it, get feedback, iterate, and, if ready, release our Fintech product to market.

So, let's go back and start from step one, shall we?

UX Research

& market analysis

16. Market research

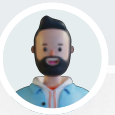
Market research can be described as gathering information about the specific target market and potential customers to verify the success of a new product or service. This method allows businesses to discover their target market, understand brand perception, collect and document opinions and make informed decisions about the product.

Market research can be conducted inside the company or outsourced to a company specializing in that field. The analysis investigates several areas in the market. This kind of practice allows tailoring the product or service to the needs of a specific user/buyer.

Knowing the user's needs and understanding their problems, pain points, and desires helps provide the best possible solution.

Market research allows collecting information on what's trending in a specific industry, the challenges and demands, and the prices of particular products and services. This includes conducting surveys, interviews, product and service research, competitive analysis, persona research, etc.

To carry out an effective market research we need to define our objectives, determine users we want to survey, choose our research method (online surveys, user interviews, usability testing), ask good questions, gather the data and analyze the results.



As this process phase is very extensive, what we're going to do is research products available on the market, find similar apps to what we are aiming for, and we're going to conduct our UX analysis by directly comparing three main competitors. After several google searches researching the fintech space, we came up with these three apps as our best-matched competitors.

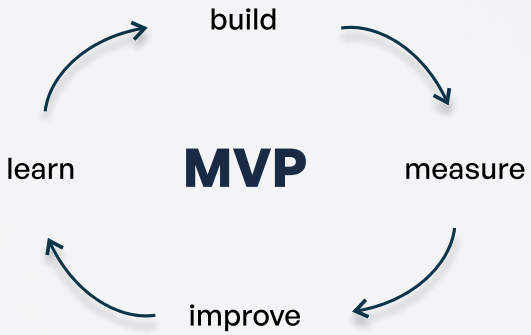
App name	Wise	Revolut	Monzo
Relese date	1/2011	1/06/2015	13/05/2016
Price	Free	Free	Free
Target User	Millenials	Millenials	Millenials
Downloads	+10 mln	+10m	+1 mln
Average User Rating	4,2/5	4,5/5	4,5/5
Features	Money transfer, sending receiving money abroad, exchanging money	Money transfer, sending receiving money abroad, exchanging money, crypro trading	Money transfer, pots, manage overdrafts, virtual and physical cards, monzo plus or busienss lite

With our product in mind, ask yourself these questions. What are our potential user groups missing in their lives that this app could fill, and how is our app different from the competition?

The first part of our UX design process is finding the answer to these questions via research, interviews, and data analysis. Let's measure our market expectations now.

The market expectations

Let's think about the list of features for a second. We could go all in and start building out our app by looking directly at what our competition is currently running with, but that's never a good idea. They've probably spent millions of dollars researching, designing, testing, and iterating their product and most definitely started with something small and straightforward - an MVP; and the goal of a successful MVP launch is to start with something simple and start building on top of it. It should serve as a solid foundation for your future app an won't let you fixate on too many details at once.



Let's assume that we've done our market research and decided we're building our app to launch it for a small group of beta testers. The functionalities we'll be making for our app should cover the most essential needs that we found and documented during our app research. We will build our product iteratively.

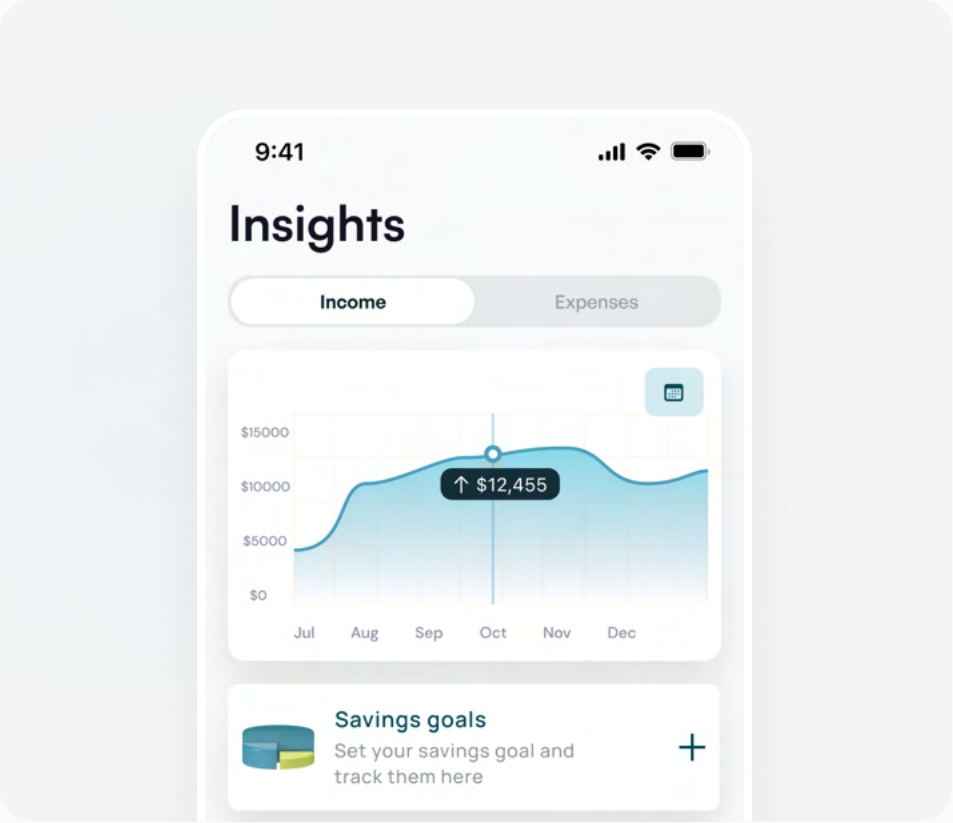
Keeping that in mind, here are the fundamental features that we've found an average user might expect from a finance app.

- Ability to connect to external bank accounts
- The visualization of data
- Expense tracking and ability to generate reports
- Bank/Credit card integration
- Top-notch security
- A currency converter
- Payment reminders
- Easy top-ups

Most importantly, we need to connect our finance app with bank accounts or credit cards to track our finances, so a seamless integration flow is necessary. We're launching our product in the UK first, then branching out to Europe, and lastly, we'll release it on the US market.

When designing our app product, we'll use the methods of banking integration available for UK users - open banking.

Visualization of data should help us present finances so that users can understand them much quicker and get an overall idea of what the average performance looks like for them in each given month.

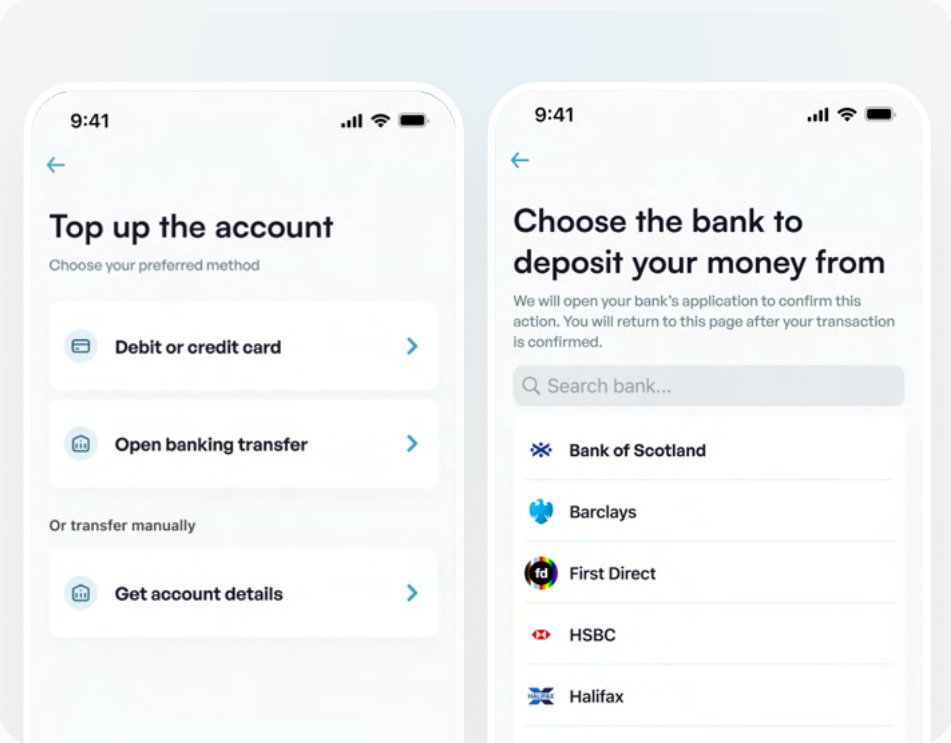


The ability to generate reports and invoices could come in handy when tracking our business spending and expenses.

Payment reminders should help us reach our savings goals much easier when ensuring we're hitting our expected benchmarks.

Bank integrations should help us move our money freely between different accounts.

Credit card top-up integration should help us top up our account with ease.



Now that we know our project goals and objectives, we did our market research and analyzed the competition. We know the expectations set for a product by looking at other successful apps on the market.

While doing our research, we shouldn't forget about our Internal Stakeholders. Carrying out a round of research or requirements gathering will help us establish the following:

- What our fintech app needs to achieve for our user base
- What our fintech app needs to achieve for our organization
- What are the potential roadblocks we might face
- What are the compliance and financial regulatory requirements
- What are the functionalities and features required to make our fintech app a success

We must ensure that our app works as much for our business as it serves our user base.

Now I get what some of you might be saying now. If this is an MVP, it doesn't make sense to build out the entire app with its features. Instead, we should start with limited functionality, offering one main functionality that's improved greatly compared to our competition, and build it out to include the rest of the functions.

I agree this would be the right way to do it if this project was real. However, as this e-book is designed to help you create better user interfaces and iOS apps, we wanted to give you as many screens and flows as possible. Having just the basic functionality included in our designs would be pretty dull. Don't treat this as a UX book; we'll cover that in the future.

UX process overview

User personas, flows, user journeys, IA

17. The overview of our UX process

Based on the data we've gathered from the fintech market and competitors, having interviews with potential users and stakeholders and then spent some time researching more about our product, we now need to determine what types of people may use our product. We've audited competitors of similar apps, brainstormed solutions to generate new ideas, and decided what our competitors did well and where their products looked poorly.

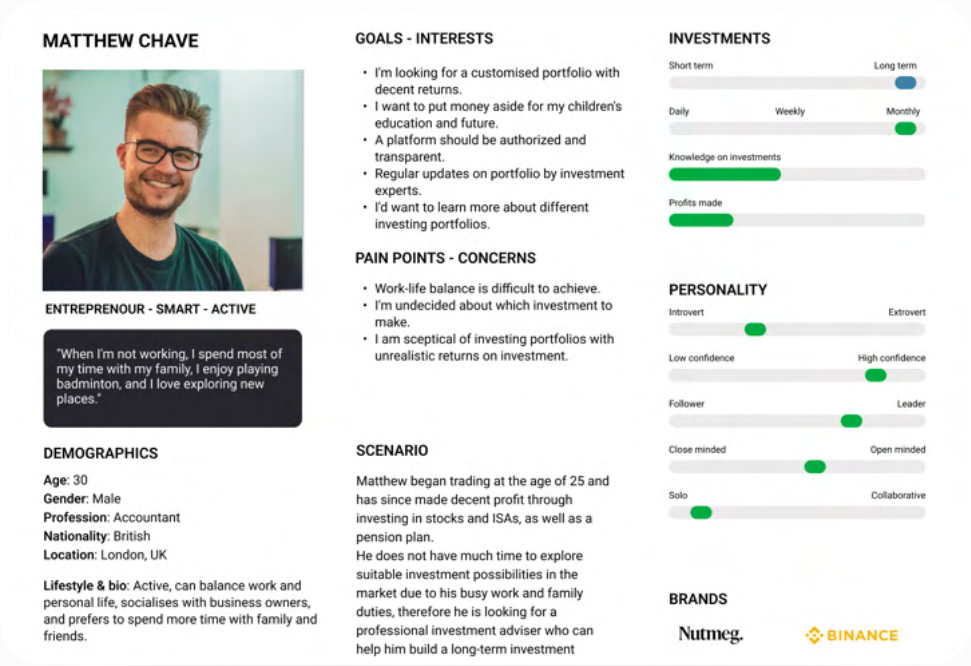
Now it's time to understand who our ideal customers are. Enriched with this knowledge, we can establish a hypothetical client avatar and encapsulate our excellent customer traits, goals, and expectations into a user persona.

User persona

“User personas” are archetypal and theoretical avatars of the product’s users. A person who exists in the theoretical space to help everyone involved in the product development process to better understand who they are trying to build the product for. Personas are tools that UX designers use to communicate their research findings.

We can, of course, build as many personas as we need; as with most products, there will be a vast spectrum of users accessing your app, and you will need to find multiple solutions to accommodate their specific and unique needs.

Let's assume we've already created a user persona for our finance app design. Remember that a persona is based on (very often) even weeks or months of research, including stakeholder interviews, observing support calls, running multiple usability tests, interviewing customers, analyzing usage data, and administering surveys.



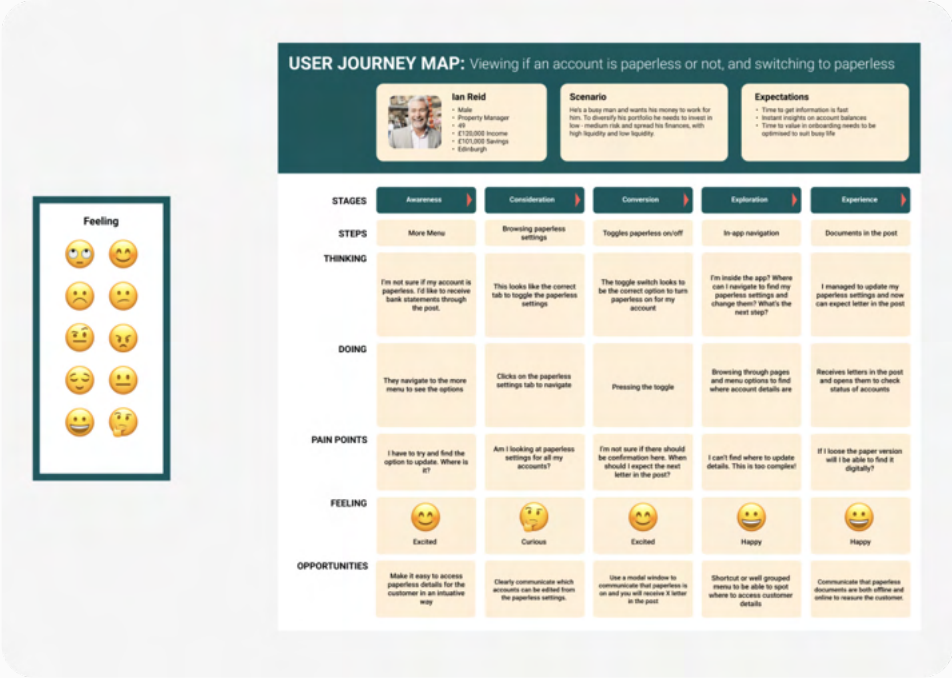
Credit: Solace Digital - our UX work on an investment app

Remember: you can find all the UX templates I use for all my commercial projects in the "Design framework" folder attached to this e-book. They're all named accordingly so you can grab anything you need and start designing!



User journeys

A user journey map visualizes an individual's relationships with a product/brand over time and across different channels.



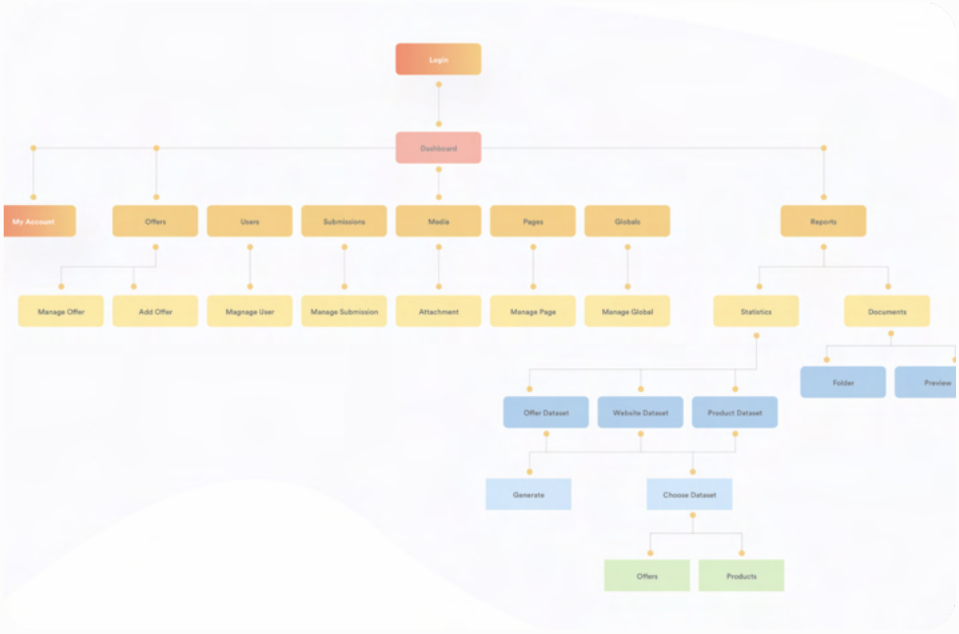
Credit: Solace Digital - our UX work on a finance app

While user journey maps come in all shapes and formats, commonly, it's represented as a timeline of all touchpoints between a user and a product. This timeline contains information about all channels users use to interact with a product.

What I like to do for my journeys is include an estimated time to completion for each journey to highlight areas for improvement.

Information Architecture

IA is truly a backbone for the design. Its job is to create a seamless experience that allows users to complete their tasks, not on finding their way around the app. UX designers' job is to structure the content, so it's easy for users to see what they might be looking for. The more content the product has, the more significant the role of IA is in the UX design process.



Credit: Solace Digital - our UX work on a student app

Hierarchy and navigation are essential when designing your IA. Hierarchy defines the structure of the content and its order of importance; navigation defines how the user navigates through it.

To create a successful user experience, we have to consider what the user expects to see and how we want to show that information as a business. We have to think about our users' **mental models** when creating our products. Add information in places where your user expects to find it. That's why researching popular competitors could help create an experience that your users are familiar with.

That's also where the user persona comes into play. It's crucial to follow the logical flow of the UX stages, not to get confused or stuck at any step of the process.

User flows

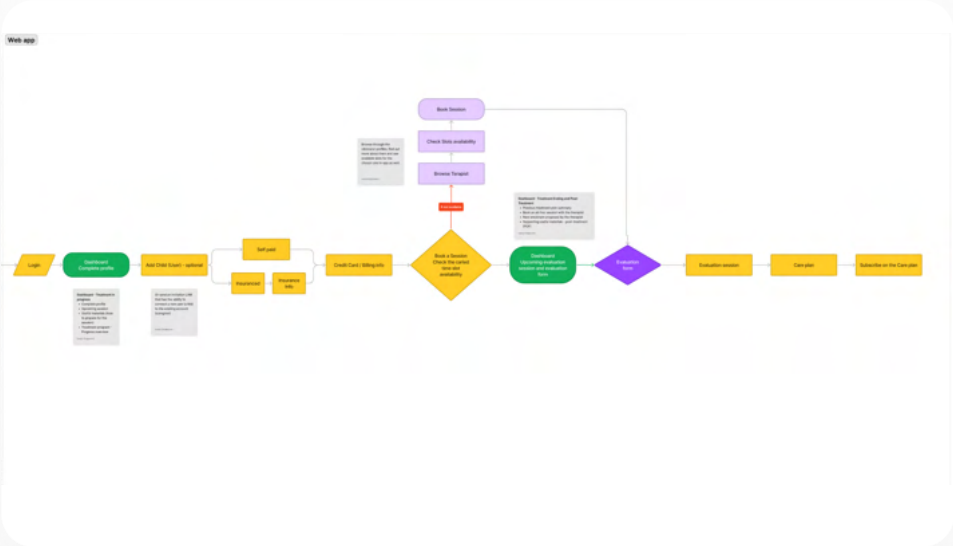
User flow - a map or diagram of the steps or screens that a user will go through while using a digital product. It can also be an illustration of someone moving through the system.

Why should we use user flows?

They help understand the logical flow of steps that the user may take while using the product.

How it is different from Information Architecture?

User flows are based on the IA. IA maps the whole application, which is the foundation for the user flow that shows the micro-interactions and the specific steps a user takes to achieve their goal.



Credit: Ivana Stojanovic

User flows serve as a great base to start building your low-fidelity wireframes. But before we do, let's conduct a UX audit of our competitors.

We're not going to show any user flows in this release of an e-book, but I've still included a Figjam template example that we would typically use. If you guys are interested, we'll include a separate UX chapter where we cover everything from start to finish going along with this project in the following e-book update.

Would you like to see the entire UX process broken down? Send me a DM on IG or an email to hello@uiadrian.com and with enough interest, I'll try to include that in the future releases!



Competitor's UX Audit

Whenever you're looking for a source of inspiration to develop your new product solutions and improve your customer's experience - analyze your competitors! UX Competitive analysis will help us define what features we should add to our product, what things we could improve, what patterns to use for the best interaction, and how to measure the success of our product.

Supposing is good, but finding out is better.

The UX Competitive analysis will help us prevent many problems, understand our users' needs better, and determine which product features we should try and implement in our product.

What exactly can we achieve by a competitive analysis?

- We get to explore all the weak and strong spots of our competitors
- When exploring the weak spots, we get to think of better solutions
- We can define what's missing in the existing product
- We get to understand the current market expectations and needs
- We can work on our advertising angle and our Unique Selling Proposition to develop a better go-to-market strategy.



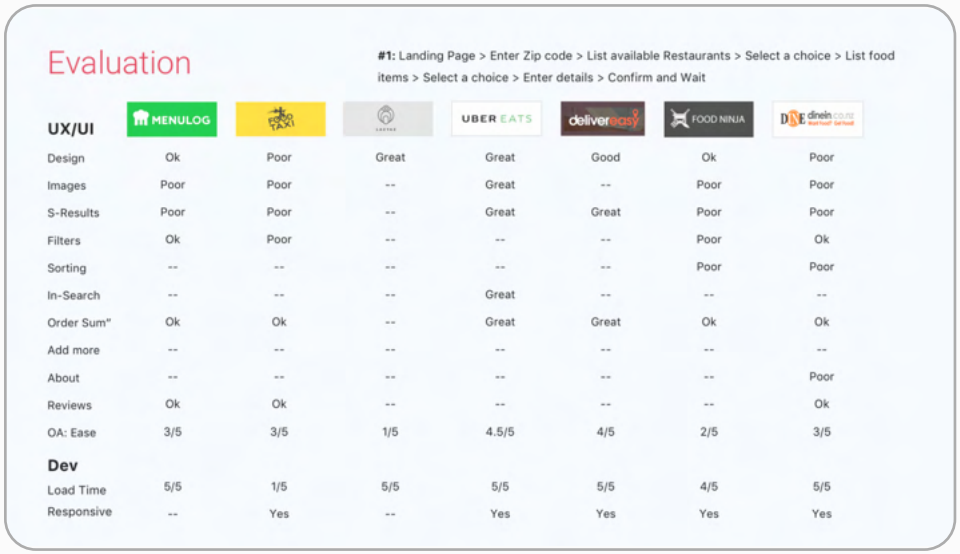
By analyzing our competitors, we should be able to gather beneficial feedback and notice visual patterns that are being repeated across most of your competitor's products. We went the simplified route and audited each of the main areas of the competitors' products. If this was an actual project, we'd dive much deeper, creating a comparison table to group all our findings and directly compare them across our selected products.

We should test at least 3-5 competitors, 2-3 direct and 1-3 indirect, track their user journeys, and measure the product's usability to get the best results. We'd also download their apps, go through their onboarding process, and explore the entire app ourselves.

After documenting all the collected data, we'd then group all our findings into a comparison table, analyze them, and draw conclusions.

We should also analyze their copy and how they address the users. This should give us a better understanding of their target audience and decide if we should use a similar tone of voice for our product.

A more in-depth and thorough competitor's analysis would look something like this 👇



Evaluation

#1: Landing Page > Enter Zip code > List available Restaurants > Select a choice > List food items > Select a choice > Enter details > Confirm and Wait

UX/UI	MENULOG	[redacted]	[redacted]	UBER EATS	deliveroo easy	FOOD NINJA	dinein
Design	Ok	Poor	Great	Great	Good	Ok	Poor
Images	Poor	Poor	--	Great	--	Poor	Poor
S-Results	Poor	Poor	--	Great	Great	Poor	Poor
Filters	Ok	Poor	--	--	--	Poor	Ok
Sorting	--	--	--	--	--	Poor	Poor
In-Search	--	--	--	Great	--	--	--
Order Sum"	Ok	Ok	--	Great	Great	Ok	Ok
Add more	--	--	--	--	--	--	--
About	--	--	--	--	--	--	Poor
Reviews	Ok	Ok	--	--	--	--	Ok
OA: Ease	3/5	3/5	1/5	4.5/5	4/5	2/5	3/5
Dev							
Load Time	5/5	1/5	5/5	5/5	5/5	4/5	5/5
Responsive	--	Yes	--	Yes	Yes	Yes	Yes

Source: UXplanet.org

Paper sketches

When it comes to paper prototyping, it's not about being a good artist but about good thinking and quick idea generation. Paper prototyping can be extremely helpful during the early conceptualizing stage, allowing one to quickly visualize and test various ideas.

We can either use an iPad or, traditionally, a pen and paper for sketching. Paper prototyping allows us to quickly design and test internally if the idea makes sense, and it can help discover fundamental problems before diving right into wireframing.

It's unnecessary to draw too many details, it's not required to do that at this stage. Paper prototyping helps you sketch the most essential phases of the work. I use it mainly to sketch out the most important parts of the app to see what they could look like.

Many designers don't take paper prototyping seriously, and it's often underestimated. This is a quick, fast and efficient method that can be done at the very early stages of the design process. It doesn't require any special skills. It's easy to modify and change because everything is done manually.

Should you use paper prototyping?

It really depends on the context of the work and your preferences.

If you're designing a similar solution for the nth time, you could live without paper prototyping. But if you are looking for a novel solution, it is a proper way to start your ideation process.

You can find printed mockups online that represent the shape of a phone screen that you can then begin sketching on. To save even more time, boot up your tablet and use software like Marvel or Balsamiq for a more digital and automated approach to "paper" prototyping.



readingwritingandgreentea.blog

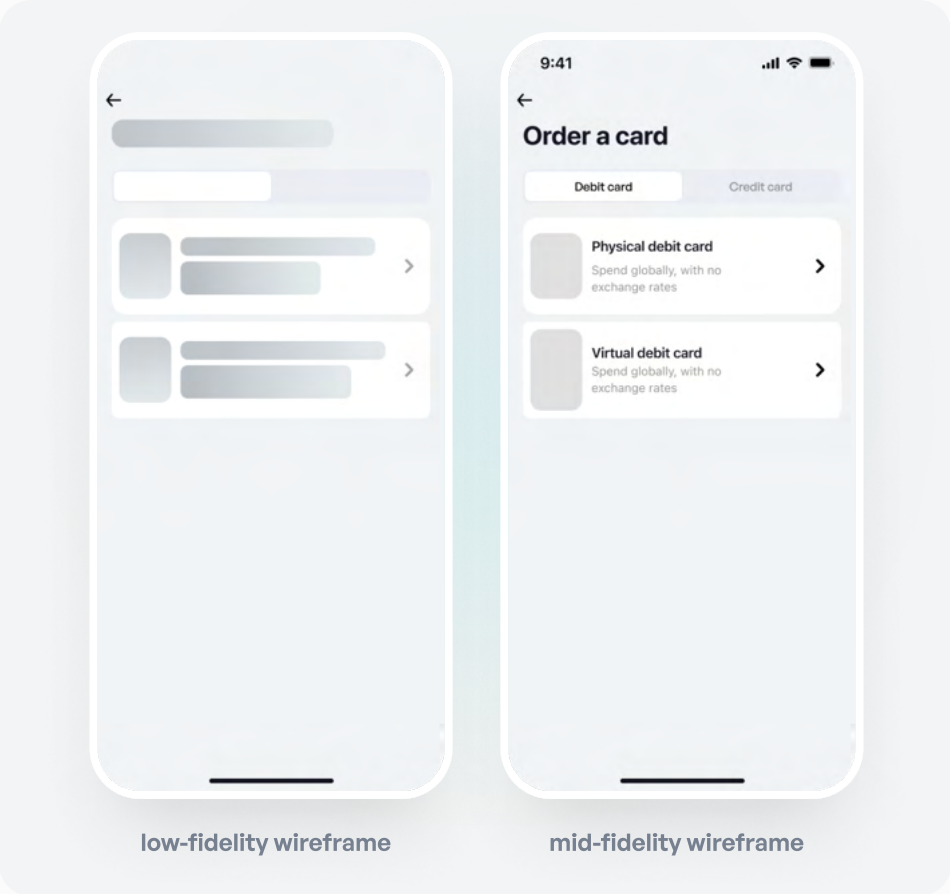
Wireframes

Wireframes are an essential element of the product design process. A wireframe is a skeleton mockup of a web or an app concept. Wireframes can come in a wide range of visuals, from low to high-fidelity, depending on your product design approach. Wireframes are part of the physical design process, which comes after conducting research and discovering the product that's being designed. Once we have our market research and analysis completed, a couple of interviews carried out, we know our market needs and wants, we've created our user persona, we've run through a couple of the most important user journeys, we've designed our information architecture, and created our user flows for each segment of the app, we've audited our competitors' products, and brainstormed solutions sketched out a few prototypes, now, we're finally ready to go into digital wireframes.

There are two types of wireframes, actually three to be exact.

Low fidelity wireframes

They usually don’t involve any details or include any colors, just the boxes and lines that act as image containers, text containers, cards, etc., just to get the general structure and layout of each screen the right way. They give an overview of the product and its functionalities without getting into detail.

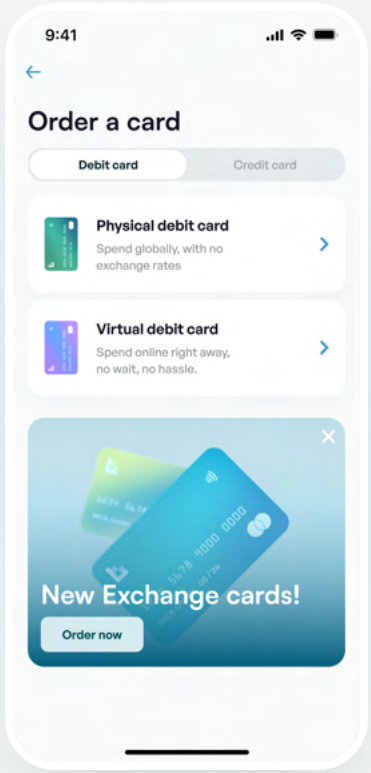


Mid fidelity wireframe

They have a much more accurate depiction of the layout with more detail given to specific components like buttons, text, and links. It usually has the correct spacing and font size and is colored either in black and white, grayscale, or follows the color scheme of your style guide, all depending on your approach and client’s expectations.

It’s the closest iteration to your high-fidelity prototype, which usually avoids final images, and illustrations and doesn’t strictly follow the same font styles and colors, but it might. They are great if you want to develop and present an idea before getting too bogged down in the fine details - perfect for our Lean UX approach.

High fidelity wireframes

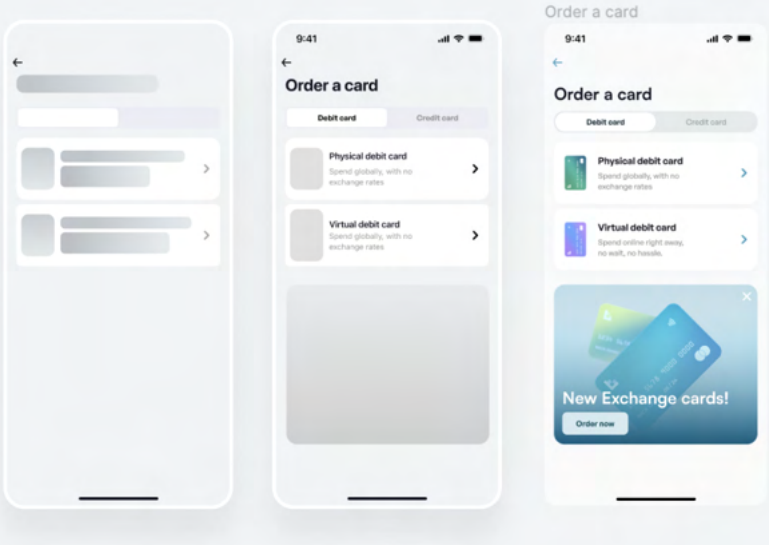


high-fidelity wireframe

And lastly, we have high-fidelity wireframes, which are the closest you can get before your final product.

They use the correct font styles and all colors are applied, the imagery is on-brand with the brand’s visual identity, and the placeholder text is replaced with the proper product copy. They’re built on a design system and are dynamic, componentized, and responsive.

For our Fintech UI Kit, we’ve skipped the low-fidelity preparation and started with mid-fidelity wireframes - you’ll find the mid-fidelity drafts inside the file on a separate page.



Mood boards

18. Inspiration vs. mood boards

An inspiration board is a collection of reference work, images, and color blocks that provide a starting point for UI designers to gather visual feedback and inspiration for the project. Inspiration boards are more than just a pretty collection of photos, shapes, and interfaces. They serve as a foundation for creating curated mood boards that are a way to find the correct brand's visual language.

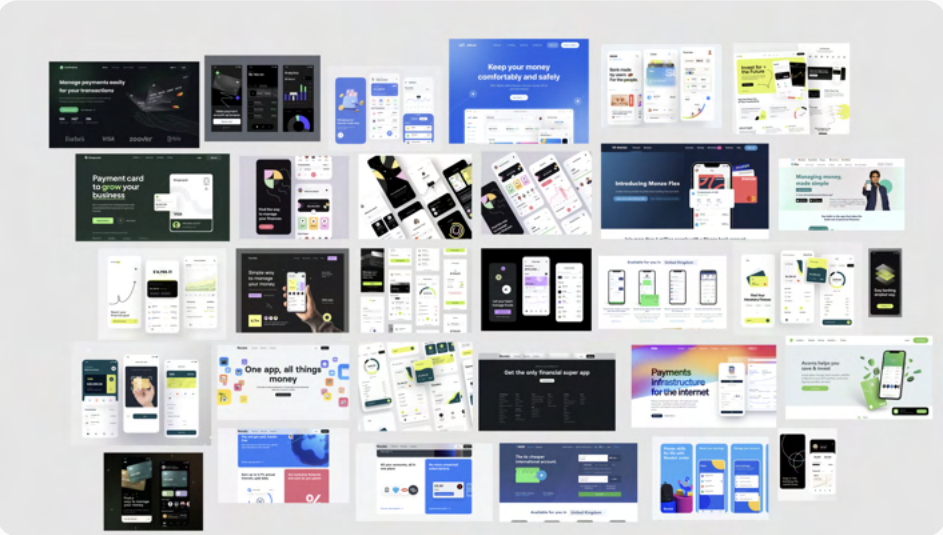
Inspiration board

Before designing and preparing a mood board for my projects, I start with an inspiration board consisting of different ideas, interfaces, colors, and design patterns. I go over the internet and find designs I like, screenshot them, and put them all together without any specific, organized order or logical way. Whatever you find helpful for your project, snap a screenshot and save it to your Figma.

Where to find inspiration for your project?

We've already covered a lot of these, but the best source of visual inspiration is Dribbble. The sheer number of designs you can find is astonishing, and that's precisely what we need here. We don't need outstanding UX and logical layouts. We just need visual inspiration, so go crazy and save whatever you find inspiring. Spend 15-20 minutes collecting inspiration and placing them all in your Figma.

You should end up with a board of inspirational images, interfaces, colors, and art that should look something like this (next page).



Once you have a satisfying amount of inspiration collected, take some time and just look at it to find similar design patterns.

Group them up into matching color schemes. You should end up with an inspiration board that looks like this 👇



Mood board

Mood boards are immensely helpful when creating a visual language. They are a significant step in a design process that helps eliminate the guesswork from the visual output our stakeholders expect from the project. Mood boards are a visual presentation of the style and theme of a given concept, a collection of colors, images, words, textures, and similar materials that illustrate or describe a proposed style or theme.

Mood boards help narrow down options and save time. They will make your stakeholders feel more in control and a part of the design process, which allows to manage the expectations much better, and progress through the design process much easier and without going back and forth to change the style and start all over again.

How to create a mood board?

I usually start by making the inspiration board and splitting up all the inspirational references into separate groups that all use a similar color scheme and similar design patterns.

After they’re all grouped up, it’s time to put them into a presentation template, add a title and a description to explain your choice of color and the overall visual direction and present these new findings to the stakeholders to gather the initial feedback.

Mood boards are undoubtedly super valuable assets in the design process. They help start defining a general visual language and help get an idea of what a client likes and dislikes. I'm saying this from experience, it pays to give this stage of the process more attention than it often receives. Before implementing mood boards into my design process, I can't say how many times I had to go back to a project and change colors or give another round of visual adjustments just because a client wanted to "test out" a different color or imagery.

By taking the extra effort to make your mood boards more detailed and polished, you'll build a stronger foundation for your product and gather much better and more precise feedback.

After one or two rounds of mood board presentations, you'll agree on the visuals, and you'll be able to proceed to the next step of the UI design process, which is creating our style guide.

Let's talk about our choices for the Fintech app for a second now to see what visual direction we should take with our product.

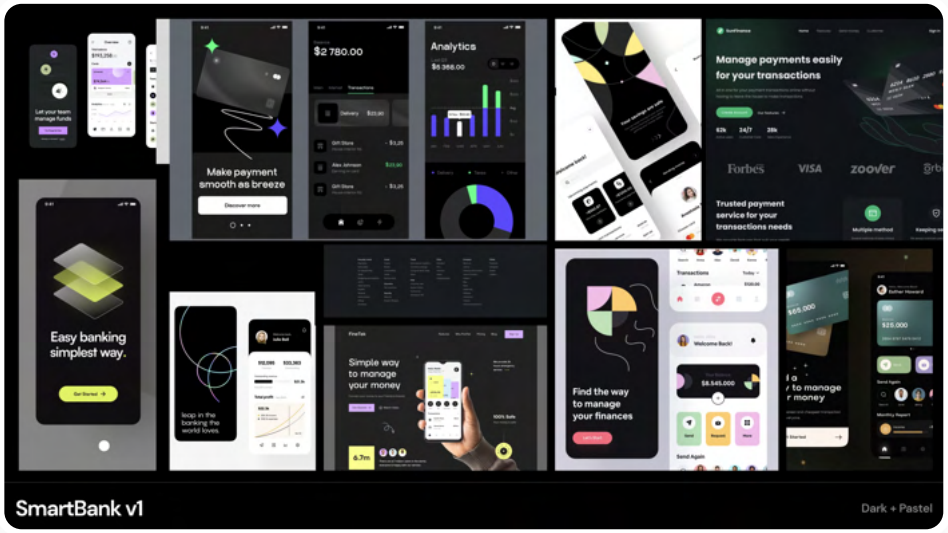
You'll find the templates we used for this project in our worksheet file - on a mood board page. **All credit goes to Mixpanel**, who shared their fantastic template with our Figma community!



Mood board no. 1 - Dark Pastels

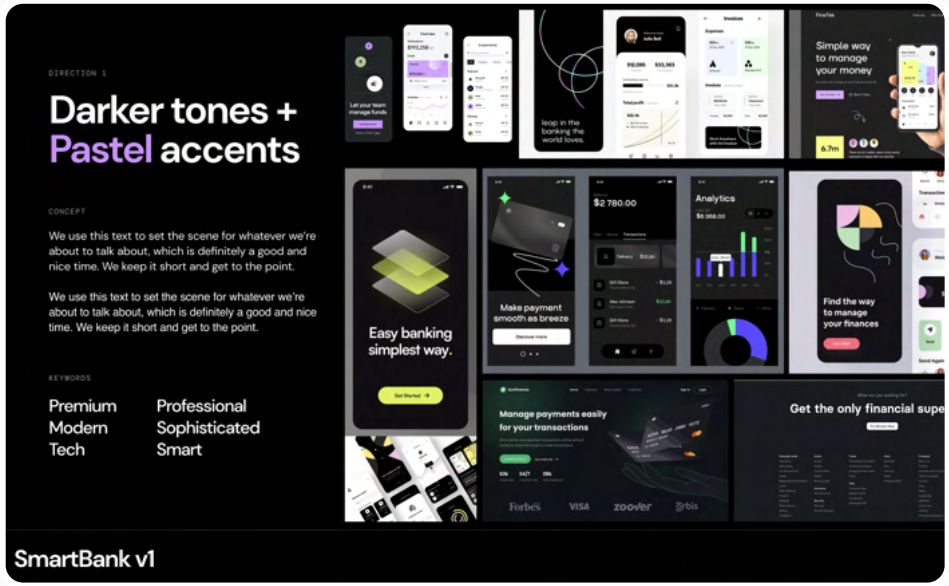
The first mood board consists mainly of dark tones and pastel accent colors. While dark-themed UIs can be stylish and elegant, they often appear dramatic and too sophisticated, especially for an app that should bring positive emotions and a sense of security and calmness. We still wanted to include that variant for a potential dark mode design we could implement in a later release.

We need to be careful designing dark UIs, though. All kinds of usability issues are coming into play here — mostly related to readability, scannability, accessibility, and contrast.



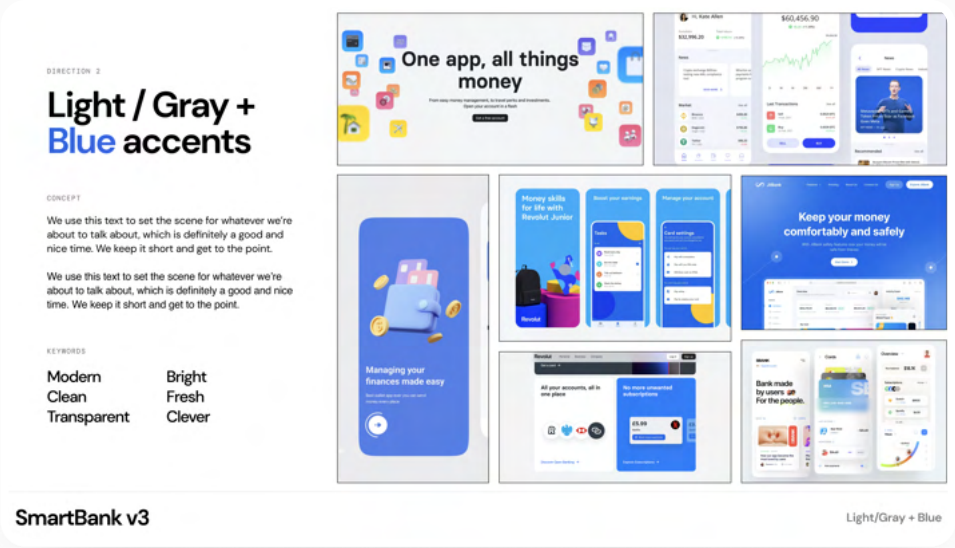


We also have to consider what time of day and the type of environment our users will be using our app. Darker UIs usually go well with gaming and movie apps. The gaming/movie-watching context and the environment where users are gaming or watching movies fit much better with a black color palette. A black background design reinforces the visuals, introduces a sense of mystery, has better contrast, supports visual hierarchy, and is easier on the eye strain. Let's pitch this idea to the stakeholders.



We shouldn't go full dark UI with this concept; instead, we should offer a toggle switch between light and dark modes at the user's preference. We could create a more dynamic, striking experience with a sense of luxury and prestige that could provide improved user experience, which correlates to enhanced customer engagement.

Mood board no. 2 - Blue

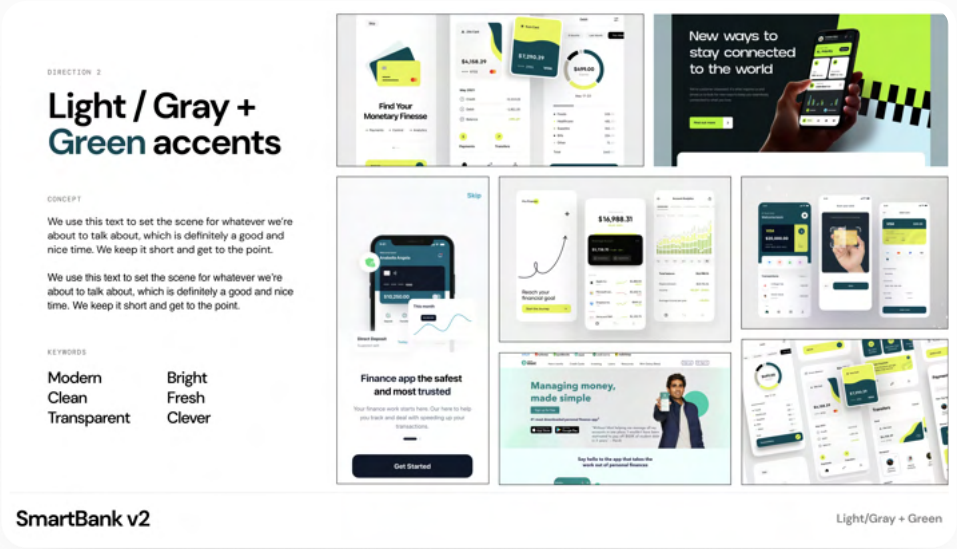




Blue is a color of trust and confidence. It's the most popular color choice for fintech apps and tech brands in general. It's a universal color that anyone can identify with. The color blue is gradually being associated with financial products, and new startups seem like they need to jump on the trend train before it leaves, so it feels like everyone and their mom is using blue in their apps now. I wouldn't use the exact words to describe it to the stakeholders, of course, but you get the gist!

We want to be a bit different with our product, so let's try something else with our third mood board design.

Mood board no. 3 - Green



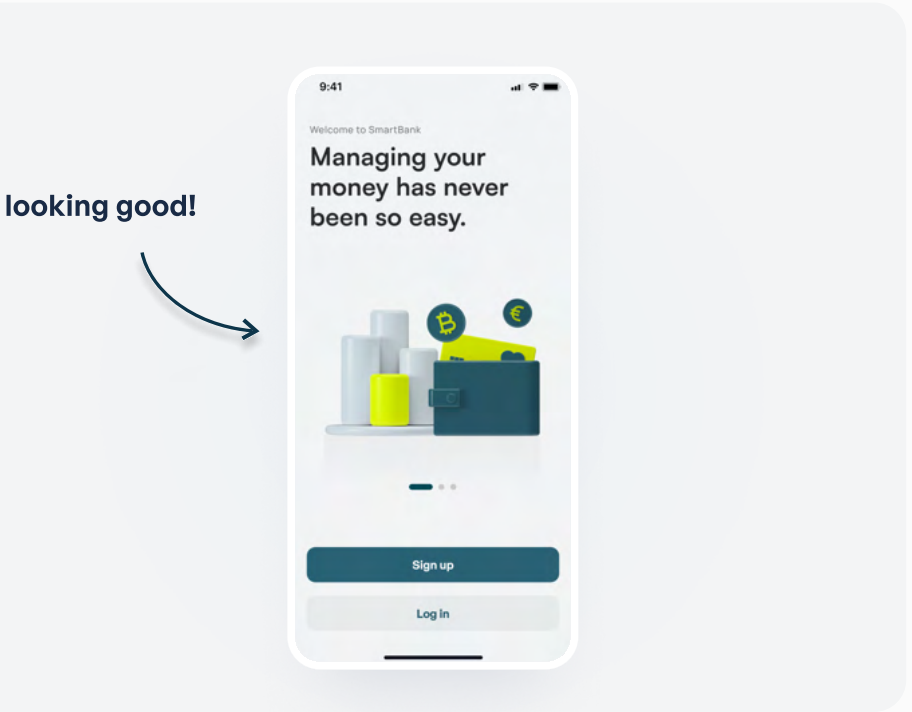


Green subconsciously creates a feeling of prosperity, wealth, harmony, and freshness. It's naturally associated with money, finances, banking, and ambition. It is also the second most visually pleasing color, as it is so common in nature.

Users are naturally pulled towards green, as we tend to see and associate green with our environment. It's also considered the most restful and relaxing color for the human eye to see. Brands with colors tend to appear as humble and accessible to their users.

We have to be careful with our greens, though. Too much green will make our app look bland, lethargic, and depressing. Making our green too green will make it look too generic. We have to find the right balance, and the following exercise can help with that. We went into a different route and added a hint of blue to create our primary color.

Let's say our client **chose the last option**. We don't see any major competition utilizing these colors, and they feel fresh, modern, and somewhat futuristic, but toned down. There's also one more thing I like to do. Using our brand colors I quickly mockup a welcome screen to visualize what our end product might end up looking like.



Style guide

& brand identity guidelines

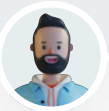
19. Let’s create our style guide

A style guide is a collection of pre-design elements, graphics, and rules to follow to help make sure each page of the website or every screen app will look and feel consistent with the rest. While designing for web or apps, style guides serve as a primary reference point.

Style guides usually include typography, colors, grids, and spacing. A style guide is essential, especially when working on a project with multiple designers and developers at many stages. The focus will help to ensure that people aren't just making stuff up as they go but follow a very strict and precise style that's congruent across all platforms and media.

Suppose a business or a design team doesn't utilize a shared file that documents all the intended look and feel of a product. In that case, visual inconsistencies will arise quickly, leaving users confused and frustrated. No one likes chaos. We are drawn to coherent and congruent visuals, and that's what the style guide is for - to create consistency across all our digital products.

Most user interface style guides include documentation for typography, iconography, layouts, and grids, color palette and components. More detailed documentation of all your components in all their states falls under a design system definition.



Color palette

By picking colors from our mood board and presenting the sample work to our stakeholders, we decided that our main primary color would be deep sea green, with our accents being bright and green. Let’s create our entire color palette now.

Primary color



A primary color will be the color displayed most frequently across our app's screens and components. We want to make it more neutral, but at the same time make it have enough weight to be distinguishable from our background and stand on its own.

Accent (brand) colors

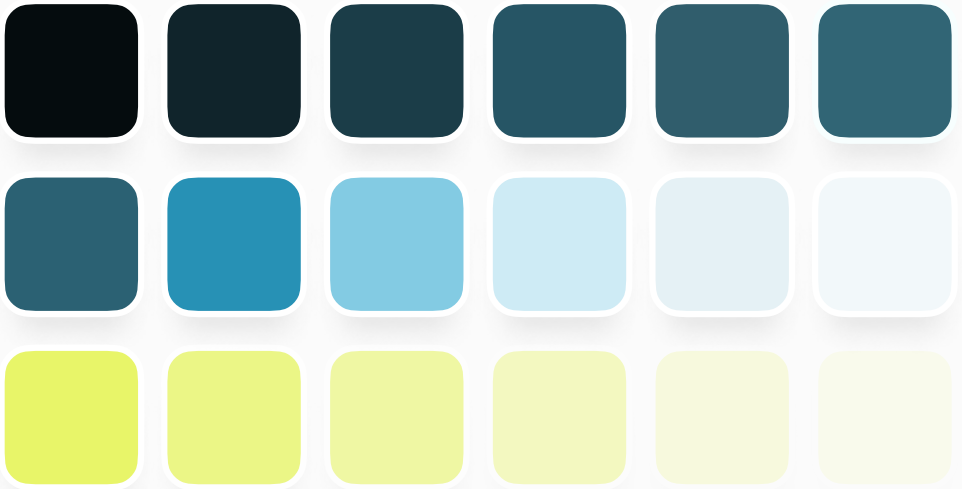
Our bright yellow-green accent color will be mainly used for CTAs and illustrations, and our ocean blue will be used to bring out more color in our gradient backgrounds, and we’ll use it to highlight navigational elements, icons, and controls.

Grays



To create our grays, we’ll pick the darkest green tone of our primary color and slightly go down on the saturation scale until we arrive at a shade as close to white as possible. This will be the color used for our background, while a shade darker will serve as a background for our iOS controllers and other interface elements.

Our entire color palette

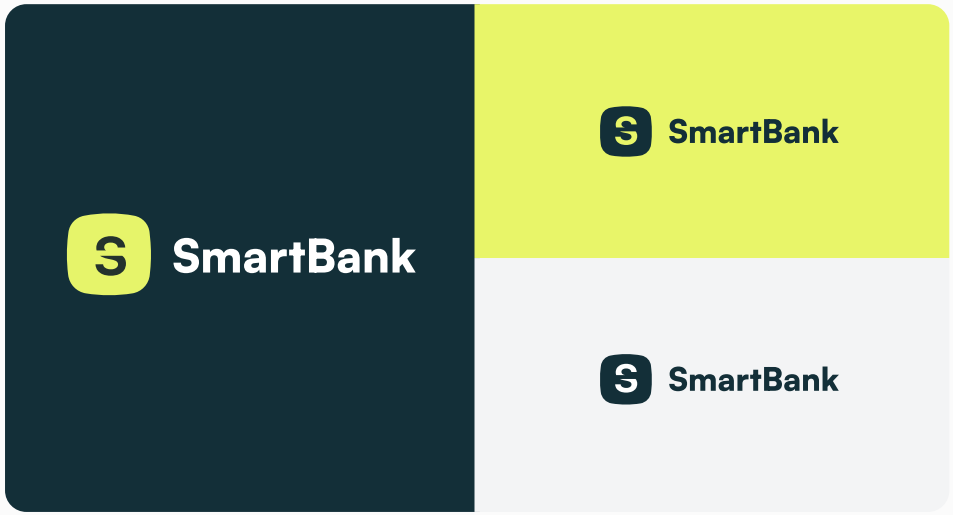


Logo

When including our logo in the style guide, we want to explain the proper use of color combinations in any digital composition.

SmartBank

We want to make sure there are no misunderstandings when presenting our logos on different backgrounds, compositions, and digital media. We want our brand to be associated with a set palette of colors, and we don’t want any visual mix-ups and surprises.



System colors (iOS)

We’ll take the system colors from the recommended iOS color palette mentioned in the iOS design basics section. We’ll only need red for errors, blue for notification messages, orange for status icons, and green for success messages and status icons.



Typography

Typography must speak for the brand. It’s a critical part of the design process that can single-handedly change your app's entire look and feel. We chose the following fonts for their modernist, Grotesk-style, geometric feel that’s fresh, light, and easy on the eyes.

Satoshi — Display & Headlines

General Sans — Callouts & Body text

Font sizes

Style	Typeface	Weight	Size	Line height (leading)
Display - Large	**Satoshi**	**Bold**	34px	41px
Heading 1	**Satoshi**	**Bold**	28px	34px
Heading 2	**Satoshi**	**Bold**	22px	28px
Heading 3	Satoshi	Semibold	20px	25px
Headline	General Sans	Medium	17px	22px
Subheadline	General Sans	Medium	15px	20px
Body	General Sans	Medium	17px	22px
Callout	General Sans	Medium	16px	21px
Footnote	General Sans	Medium	13px	18px

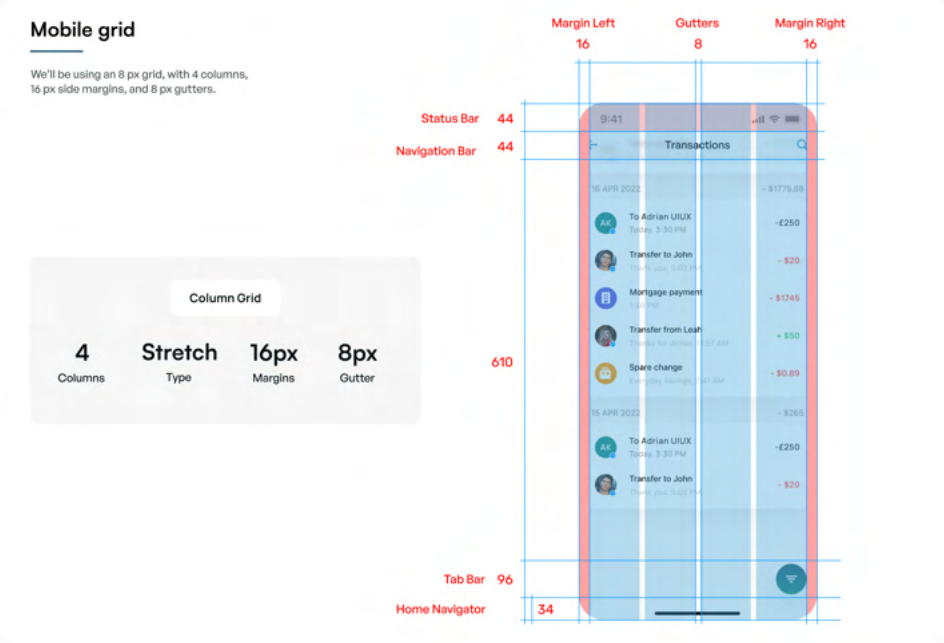
We used the font sizes recommended by the Human Interface Guidelines, so there are no changes here. We've only created two variants - one regular and one bold, to keep things organized and our choices flexible.

Line-height and letter spacing values differ from font to font, and you're free to use the values suggested by iOS. Keep in mind that they were designed specifically for San Francisco font, and they might not be perfect suited for your font.



Layout grid

We'll build our app on a standard 8 px grid with 16 px side margins and safe space left for the status bar and the home navigator. We'll set a grid of 4 columns with 8 px gutters. To toggle it on and off, you'll have to click "Control + G" on Mac and "Control + Shift + 4" on Win.



Instead of aligning everything strictly on a grid, we'll also use the "Soft Grid" method, as it's much easier for development. We cover the layout grids in the next chapter so that's all you need to know for now.

Imagery and visuals

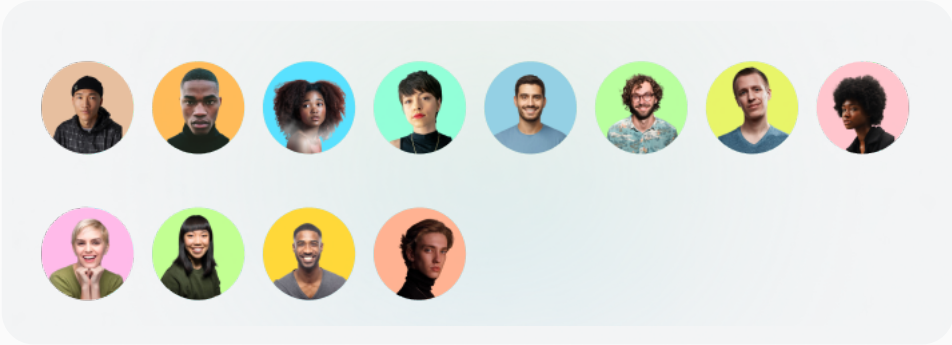
With our imagery, we decided to go for the 3D look. 3D illustrations are definitely more eye-catching and will help us create our brand's modern and youthful look. Our 3D renders will look vibrant and clay-like with visible roundness and playfulness matching our choice of fonts and colors.

The entire library of 3D components we used for this project can be found on the ui8 website called Feenancy. I've included a few of the elements in our files, but due to copyright reasons, I'm not allowed to give you to the source files, guys; sorry about that!



Avatars

We'll use a colorful library of avatars from our amazing Figma community. You'll find all the assets on the components page.



Credit: abhidesigns__

Icon pack

We've used the Huge Icons icon set from ui8. A fantastic collection of over 3000 icons grouped into 22 different categories.



Brand identity guidelines (simplified)

We probably wouldn't be tasked with creating brand identity guidelines as UI designers, but it's worth mentioning here anyway. Besides, knowing how to build even a simplified version of the brand identity guidelines for digital or app use will always be valuable for your future product work. A documented guide on using colors, logos, patterns, and thumbnails is a super helpful resource to have.

Mono color logo

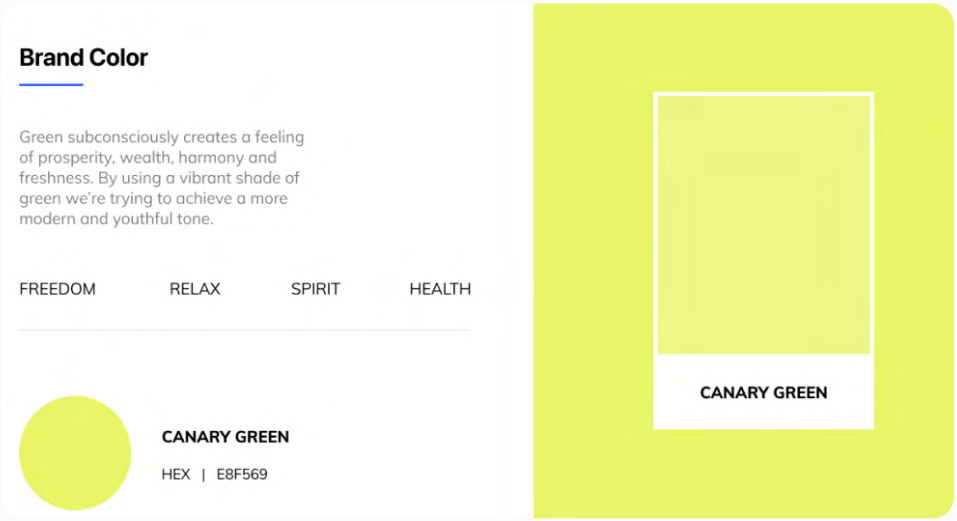


Monochrome means using a single color throughout the whole logo design. It does not contain any other effects, shadows, or shades other than the single color selected.

Brand colors



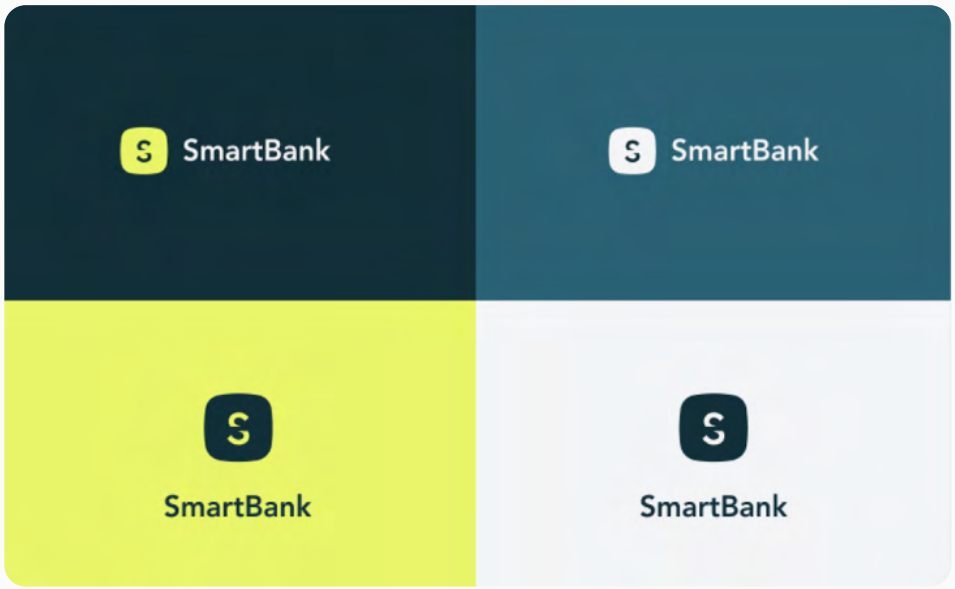
Like the style guide, we've listed out our color palette colors with their names and highlighted the main brand accent, the Canary Green - a fresh, vibrant, and modern color.



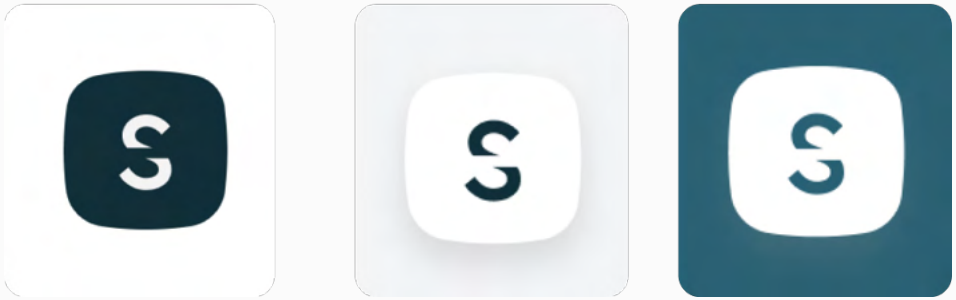
Backgrounds and use of color



We want to show alternative color combos so that there is 100% clarity on what we can do with the logo and what we should avoid. We aim to achieve maximum legibility with our logo stamps.



Brand mark



A brandmark is a symbol, element, or visual image that helps immediately recognize a particular company. It's crucial for developing and maintaining our brand's image - we would also use it to create our iOS app icon.

Brand Patterns



Brand patterns are used as an additional branding element. They play a significant role in creating strong brand recognition. They bring more depth to the brand identity and create a more memorable brand experience. Furthermore, they often incorporate an element from a logo or a shape consistent with the whole brand image.

Grids & Layout

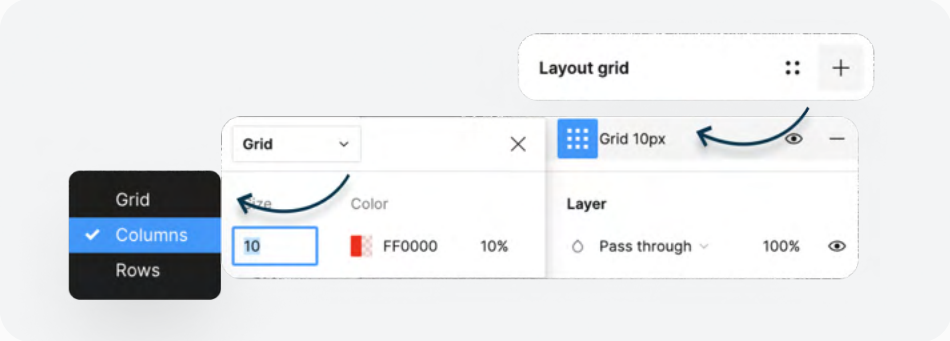
20. The mobile grid

Having a good grid system in place helps organize our designs and make much faster design decisions by giving us more control over alignments and spacing between elements.

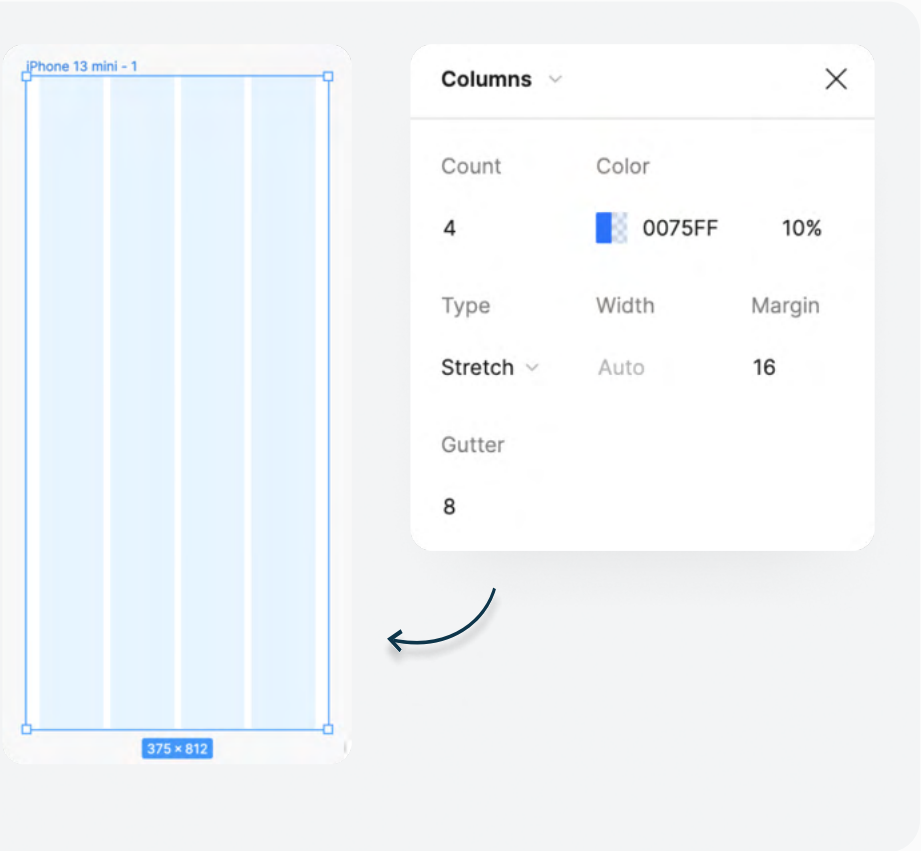
In Figma, layout grids can be applied only to frames, but you can enable multiple grids for multiple frames nested within other frames or frames inside your components. This gives you total freedom in creating as many grids as you need to space out the elements in your designs. Before explaining the different types of grids, let's quickly cover the basics.

How to create grids?

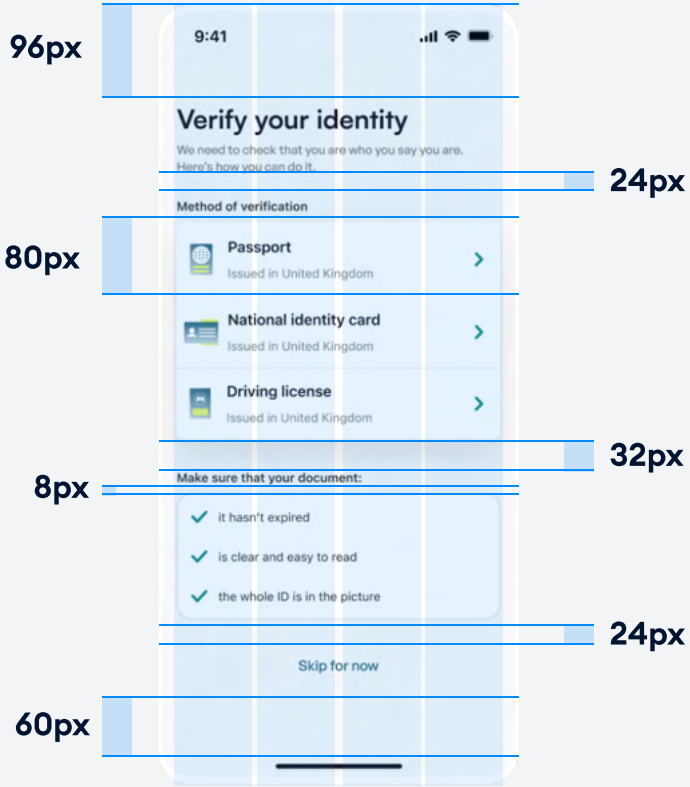
Select your frame, navigate to your properties' panel on the right, and click on the "Layout grid" plus button to create a grid layout. It will automatically create a 10 px grid for you. We don't want that, though. We'll need to create a column grid and build our designs on a "Soft Grid" instead of using the baseline grids.



Now that we have our column-type grid selected, let's set them up correctly. **Change the Count to 4**. Choose a color you'd easily distinguish from your designs - usually **blue**, **purple**, or **red** is good. Leave the **Type** at "**Stretch**" - we want it to stretch the content inside but retain the same margin values on the side. Set the **Margin** to **16 px** - it is a value recommended by iOS and leaves enough safe space to work on. Lastly, change the **Gutter** to **8 px**. We'll use increments of 4 and 8 to space out our elements on the artboard.

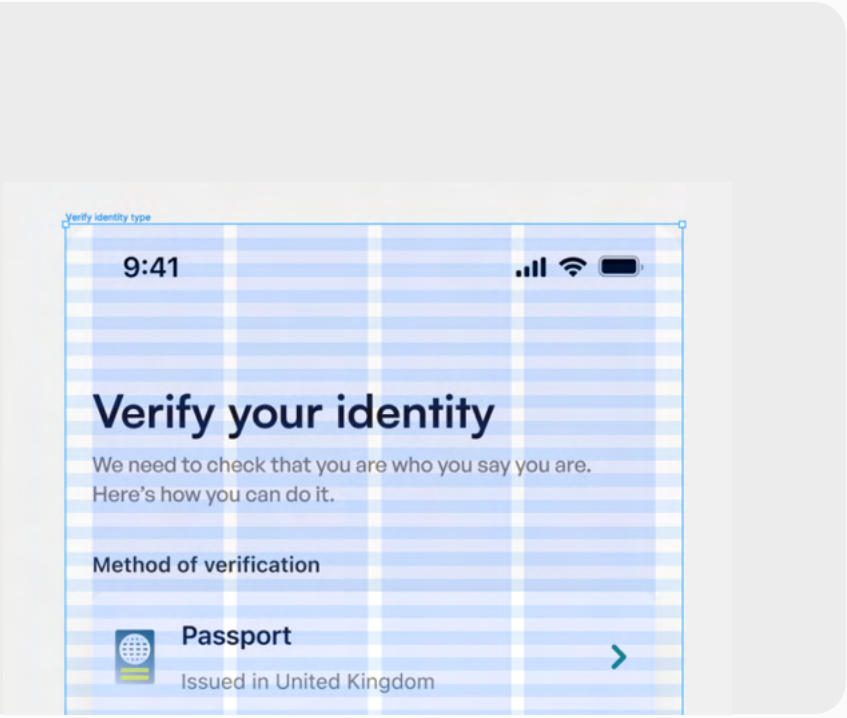


We've just created a 4-column grid, and it's considered a "Soft Grid" method based solely on a column alignment with vertical spacing being increments of 4 or 8. Let's take a sample screen from our app and analyze the spacing based on our "Soft Grid."



In the example above, everything is perfectly spaced out in increments of 4 pixels. That's really the only thing we should look at when designing with a "Soft Grid". What about "Hard Grid" though?

With “Hard Grid,” we would have to ensure that all elements, whether text layers or shapes and containers, sit on a perfectly flat baseline grid. Let’s take a look.



As you can see in the above example - our text layers are somewhat of a baseline grid, although not every element is perfectly aligned there. This creates additional problems with development as there is no such thing as a baseline grid in mobile app development, and it’s much easier to just design with 4-pixel increments between elements instead of perfectly aligning every element on a baseline.

That’s pretty much all you need to know about grids for now. We’re almost ready to jump into our Practical part.

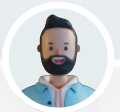
Next up, you’ll see me record a short video explaining how you should approach the Practical workbook part, where you should find all the assets and files, and how you should set up your Figma files for maximum efficiency when working on improving your designs.

I’m super excited that you have arrived at this point in the e-book! This means you’ve already gone through the basics of iOS design and understand what different tools in Figma are used for. You know how you should approach the UX of a product, you know what UI design framework to use to avoid roadblocks in your design process, and you know the basics of auto layout, constraints, and grids.

Now it’s time to play that video and set up your Figma. We’re finally ready to get our hands dirty and practice design!

I hope this introduction to our practical workbook gave you the foundation to understand the very basics of iOS design. It’s time to put this knowledge into practice.

See you on the following pages!



Please, play the video first

wireframes

21. A few words before we start designing

This is a big project, and it would have been tough for me to explain everything in written form. You've just gone through over 200 pages of introduction with another 350 pages of practical design knowledge about all the essential mobile flows like Onboarding, account setup, Verifications, Settings, Insights, Home, Search, and so on in the next 15 practical chapters.

I owe you proper instruction on approaching this next part to maximize the learning experience and work through each screen as effectively and efficiently as possible. **So, please, click on the image below and play this video first!**



Prepare your practice files

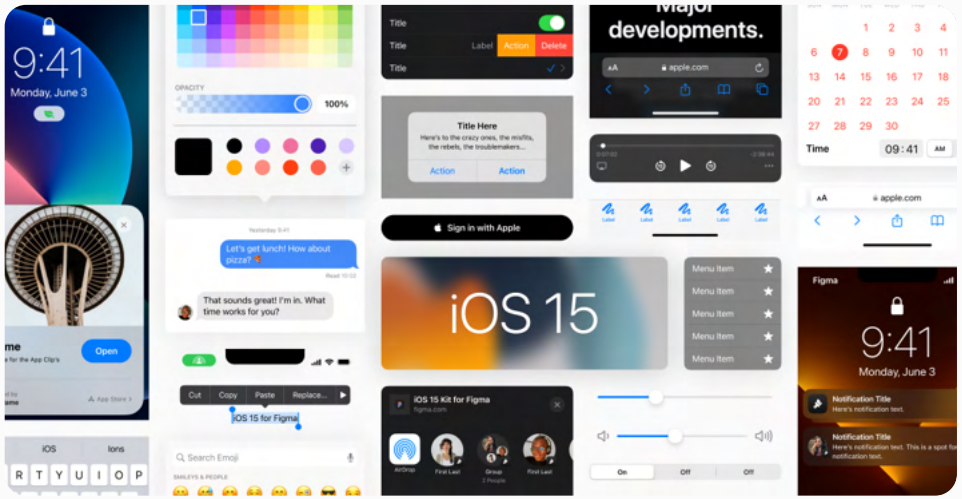
22. Your practice files setup

This is one more reminder to prepare all your project files - if you've followed the instructions from the video, you should have everything set up the right way.

I just wanted to briefly go over everything you should have opened by now to make your learning experience as easy to follow as possible.

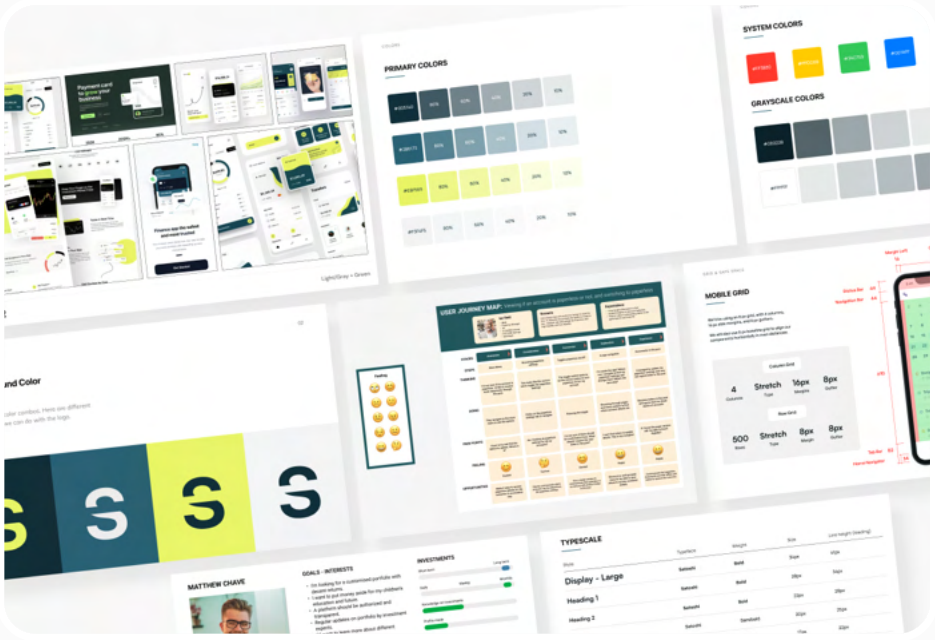
iOS 15 UI Kit

This is your best friend and one of the most important assets we'll use for this next part. You will also need this UI kit when working on future iOS designs, so make sure you have it ready and easily accessible. The file is in the iOS 15 UI Kit folder, and if you prefer to work in a web browser, you'll find the link in the "Resources and Assets" document attached to this e-book.



UX and UI design templates

All the UX and UI templates we've used for this project can be found in the "UX/UI Design Framework" file in the main Fintech UI Kit folder.



Auto layout training file

If you need to practice your auto layout, I've also included .fig files and links to the Community practice file created by Figma themselves. We will keep on explaining the auto layout basics and best practices, but I'd recommend you at least give it a try and scan through the content to see if you need to practice it more.



Fintech UI Kit

This workbook comes with two Fintech UI Kit files. One is a source file with no edits made whatsoever, with all the original components and assets and all pages following our UI Design framework. You can do whatever you need to with this file, use it as inspiration, borrow some ideas for your own projects, change the colors, play around with the fonts, and build something new out of the components you have in the library. Go crazy with it!

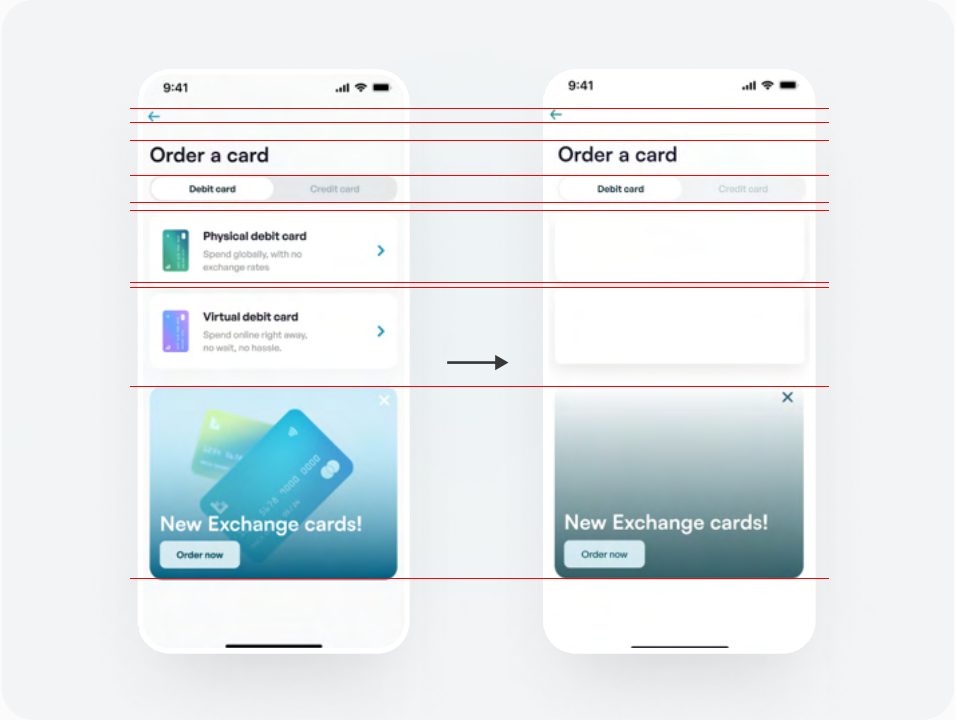
The second file is our design practice workbook, where we'll be working on replicating our original designs by using best practices like auto layout and constraints, making sure our spacing, margins, font sizes, and styles are consistent and standardized.

Fintech practice files

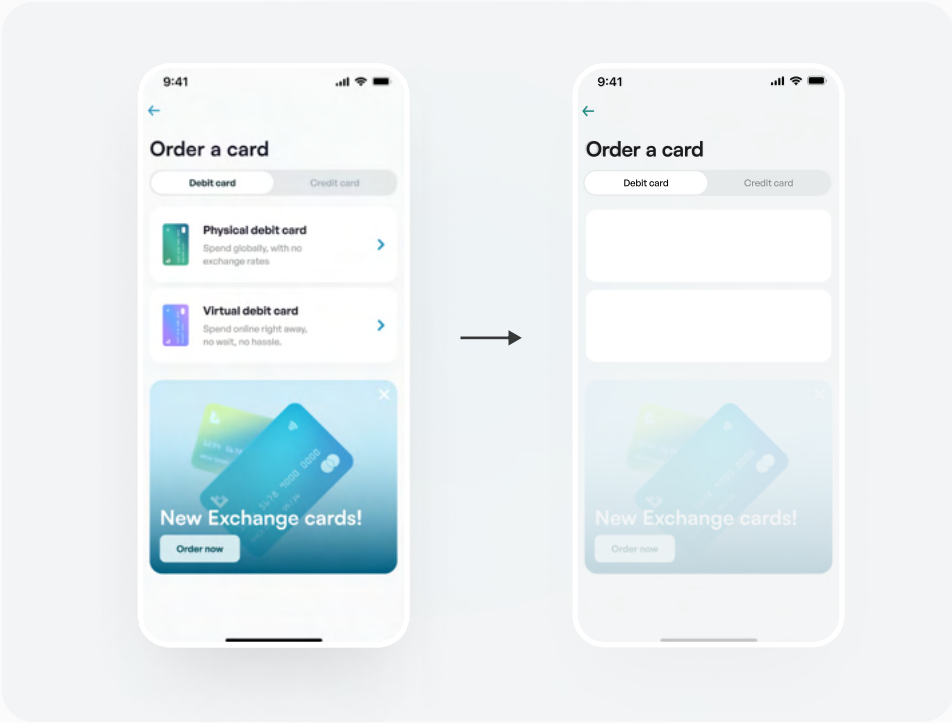
If you've watched the video, then you have already seen how to approach this file, but I'll repeat it one more time, just in case.

When you open up your Fintech practice files, you'll see 100+ original and 100+ empty frames. The empty frames are where you should be working on replicating the designs I've created. The original frames should serve you as a reference point.

Depending on how comfortable you feel with the design, you should **copy my designs** to learn the proper spacing and font sizing.



Or you should trace them. To trace the design, simply export the original frame as a PNG, place it on your training frame, reduce the copy layer opacity to around 20% and lock the layer in. Start creating shapes and adding text layers in the same position as the original. Look at the left original frame for reference when you need to look up the spacing between elements. To check the spacing between elements, click on an element and press "Option" on Mac or "Alt" on Windows. You will see values appear in red.



For more information, make sure to play the video! There's one more thing we need to cover in the next chapter.

Ready, set, Figma!

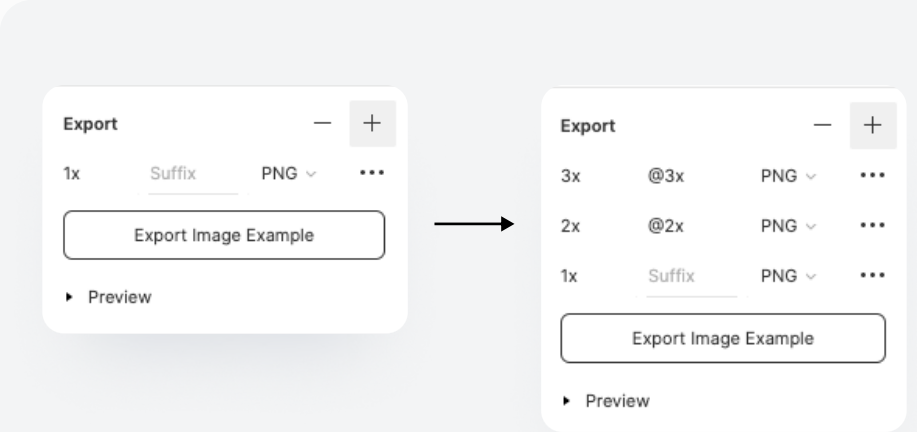
22. A few words before we start designing

Ok, I promise you, this is the last chapter before we start designing! :D

Here are the last few things I want to mention - Image resolution, image resizing properties, and creating components.

Image Resolution

The same thing applies to all of your artwork in an app. You need to export it at @1x, @2x, and @3x versions to support all devices your app will be designed for. With Figma, you don't have to manually calculate the dimensions. Just make sure to utilize the properties panel when clicking on "Export". At the same time you have your image layer selected, make sure to click the "+" three times to prepare three different image resolutions.



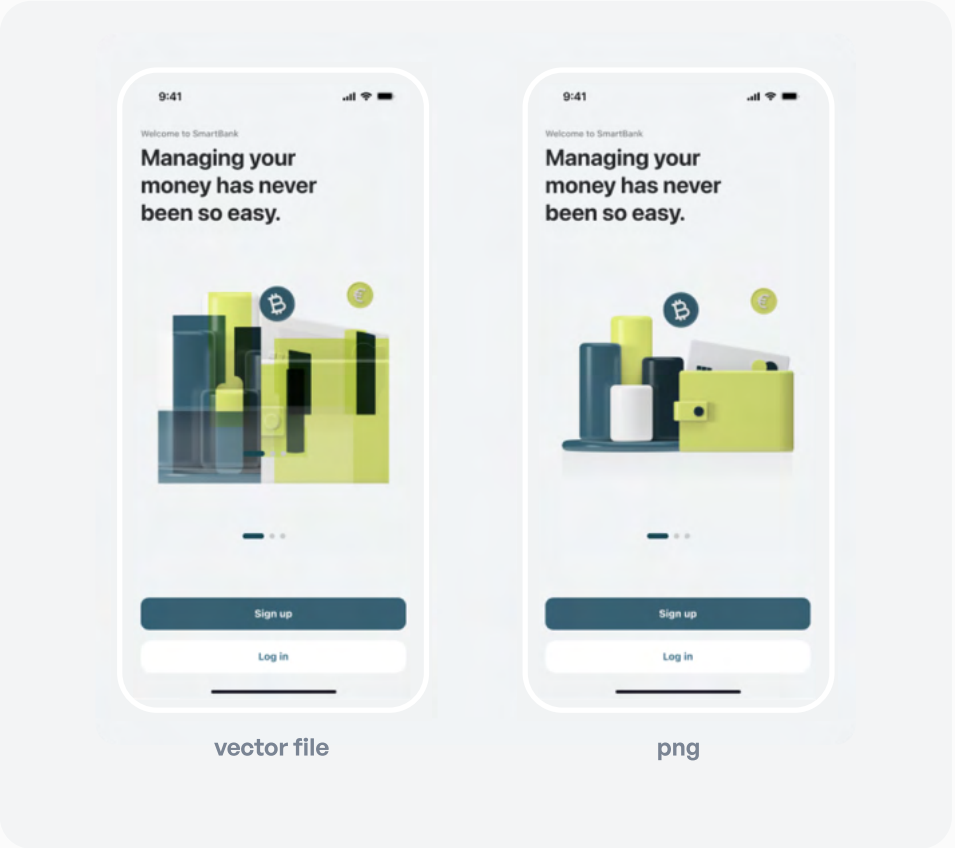
Put them into your Project folder under “Image assets” and have them ready for your hand-off. Although, if your developer is not familiar with how Figma works, take some time and explain to him how to export these assets. Just make sure to name your layers appropriately.

Images and illustrations

Most people scan content by looking at images to understand the content much faster than words. This makes illustrations a logical and rational choice for app tutorials that have to be short and concise.

When designing your illustrations and combining vectors to create a composition, like I did in our example, you should either group all vectors first and export them as svg, or make sure all the vectors in the group have a “Scale” property applied. Otherwise they will stretch out and lose the right dimensions. If you want to use a single vector file, export the group, upload it back on our frame and replace our group selection quickly by clicking “Shift + Option + Cmd + V.” You will ensure that your images stay responsive according to your constraints. If you did it with a complex composition consisting of vectors, your design would break.

Look at the example on the below. Both frames have the same constraints set, with one image being a group of vectors placed on a frame and the other one being a single exported vector group.



Remember when we’re moving further into our designs, as I will briefly mention this but not explain it every time in the same detail.

Buttons

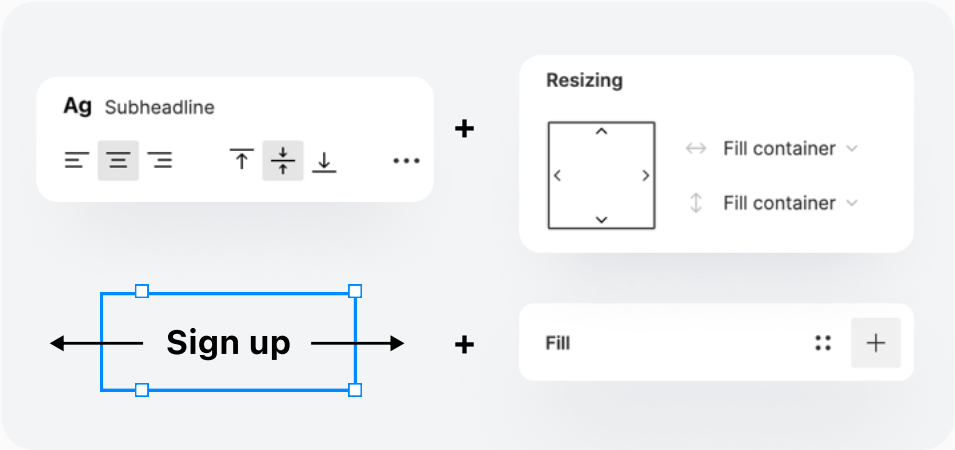
We'll be using components for our buttons so that, if needed, we can change the style and size once and update it everywhere.

Before we jump to the Design Workbook, let's use this example to create one and add it as a component to our library.

Let's create one together using the auto-layout function.

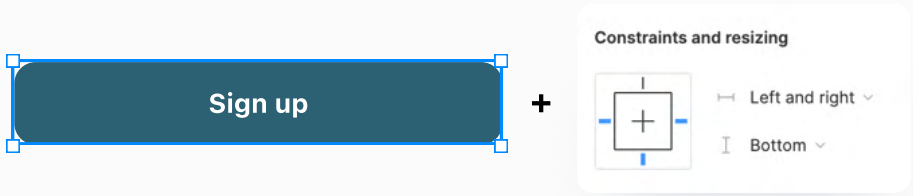


Add a text layer by clicking "T" and start typing "Sign up." Click "Shift + A" to apply auto-layout with your text layer selected. Select your text layer by hovering over your text and clicking "Cmd + mouse click." Change the resizing to "fill container" for both height and width. Change the font style to "Subheadline" and align it center.



Drag your button to resize it manually to 341 px width and 48 px height. Apply fill and change it to your primary color.

Select your button and set constraints to Left, Right and Bottom, since in 90% of cases, our button will be aligned to the bottom of the screen.



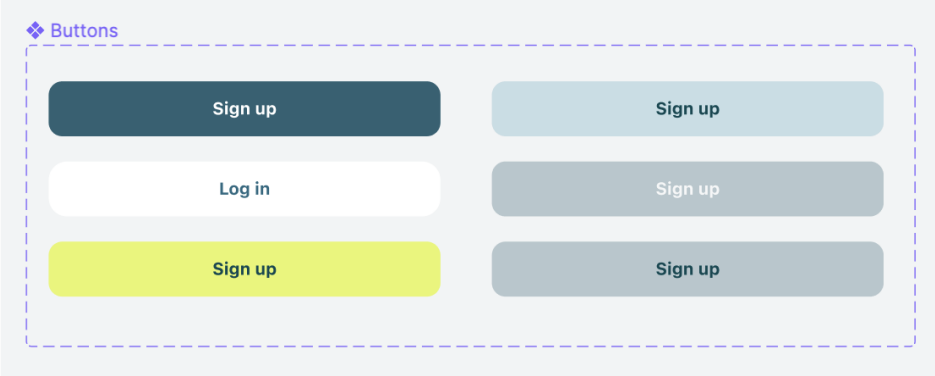
There's one more thing to do here. Let's create a component out of our button so we can reuse it everywhere later on.



Move it out of the frame and select it. Click on the plus next to Variants on the right Properties panel and let's create some variants out of our button.



We'll need buttons for different situations, like primary, secondary, disabled, one for darker backgrounds, one for lighter, and one for tertiary actions. Our standard size for the button will be 48px tall and 341 px wide.



Look at the Figma file and navigate to the "Components Library" page to see how I've designed these button states. You should replicate my actions and apply them to your practice file, as I previously mentioned.

The best way to learn design is by replicating work and analyzing it from every angle possible. This e-book should serve you as a guide for whenever you feel lost at any screen in our app flows. First, find the screen you're stuck on in the original Figma file, check every setting, and if you're still unsure how to proceed, refer to this e-book for an explanation.

Having covered the basics, we are now ready to proceed to our app and protect every flow, from firing up our app and landing on the launch screen to changing language in our app settings.

So, one more time, open up the original Figma file, open your practice file with blank screens, follow the steps outlined in this e-book **and let's start designing!** 🙌



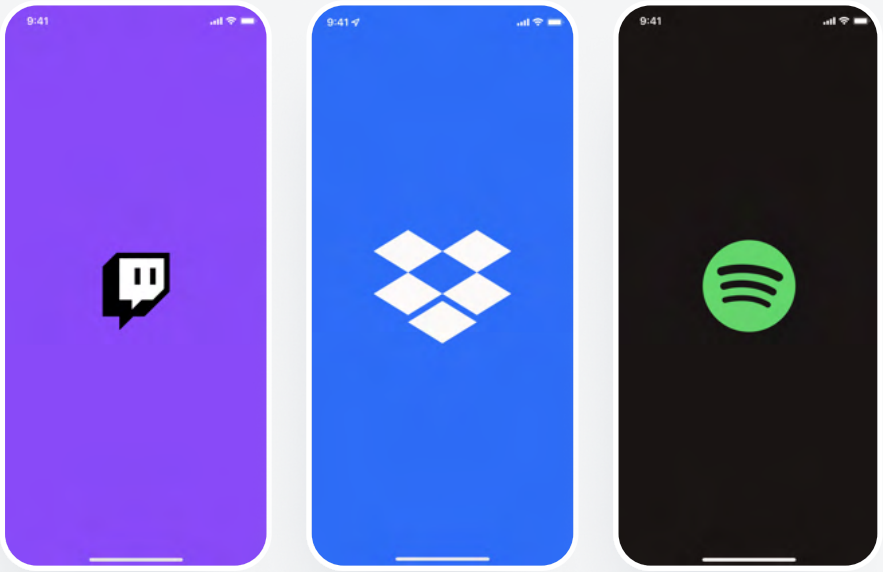
Launch Screen

THE INTRO

Launch Screens - the theory

A launch screen is a short introductory screen that's meant to set the tone for the rest of the user experience, smooth out loading delays and ensure that users' early experience is a positive one - no one wants to be stuck on a blank screen while the page is loading in the background.

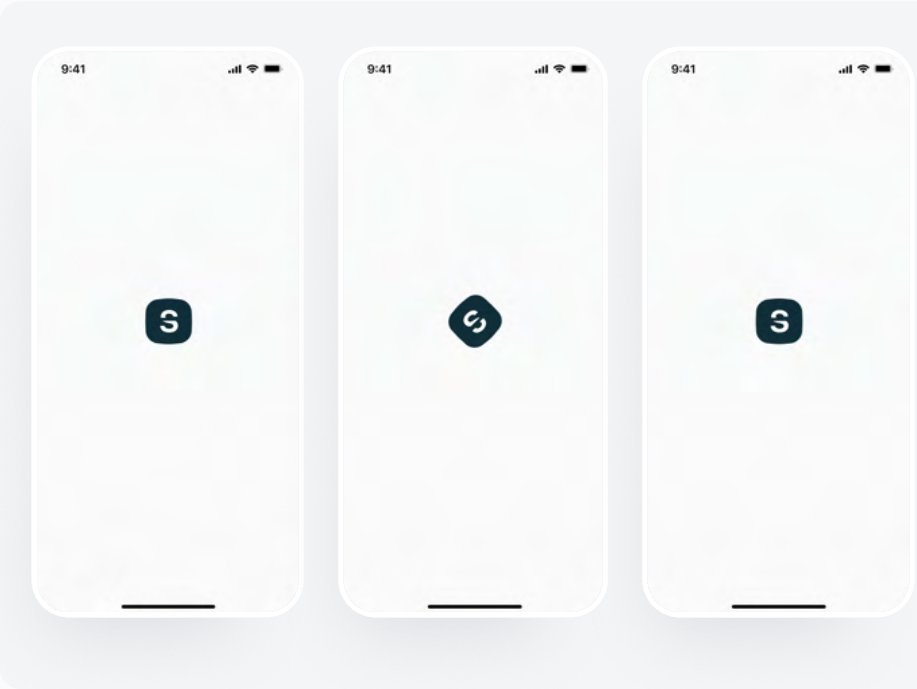
This screen can be an image, graphic, logo, or animation sequence, sometimes coupled with a progress bar. It's important to note that not all mobile apps need to have launch/splash screens.



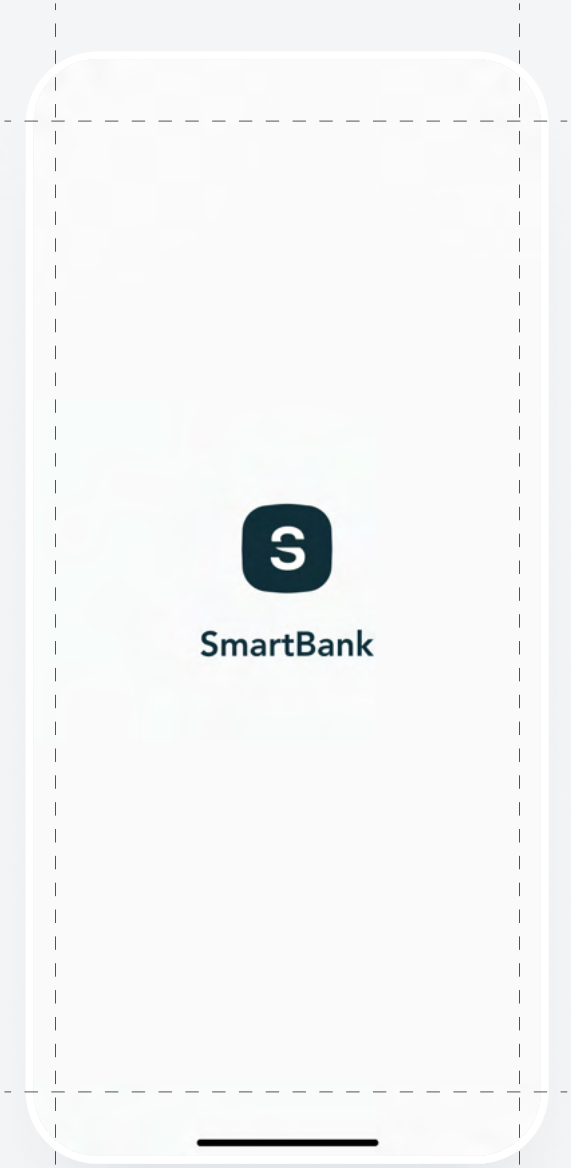
Popular app launch screens

Sometimes launch screens create unnecessary friction, primarily when an app is used daily, and seeing the same launch screen every time you launch an app might be irritating.

In our case, we've decided to use one just to set the tone and welcome our users with a nice logo animation that's designed to play in less than a second.



There's a lot of debate about the necessity of a launch screen these days since operating systems have advanced and apps can launch almost instantly. Launch screens are still present in nearly all popular apps and offer an excellent chance for us to do some prep work before the main screen is shown.



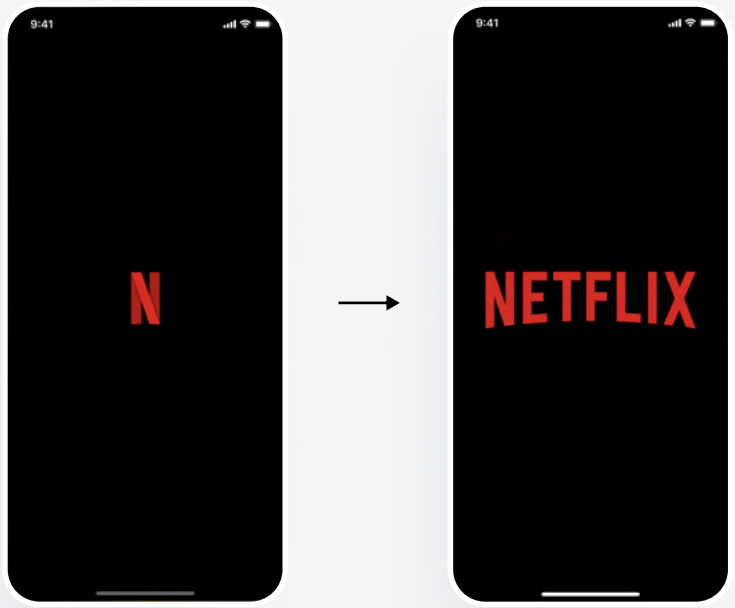
1.1.1. - Welcome Screen

The launch screen essentials

When designing a launch screen, there are a few things you might want to consider. Let's break them all down below.

Make it short and concise

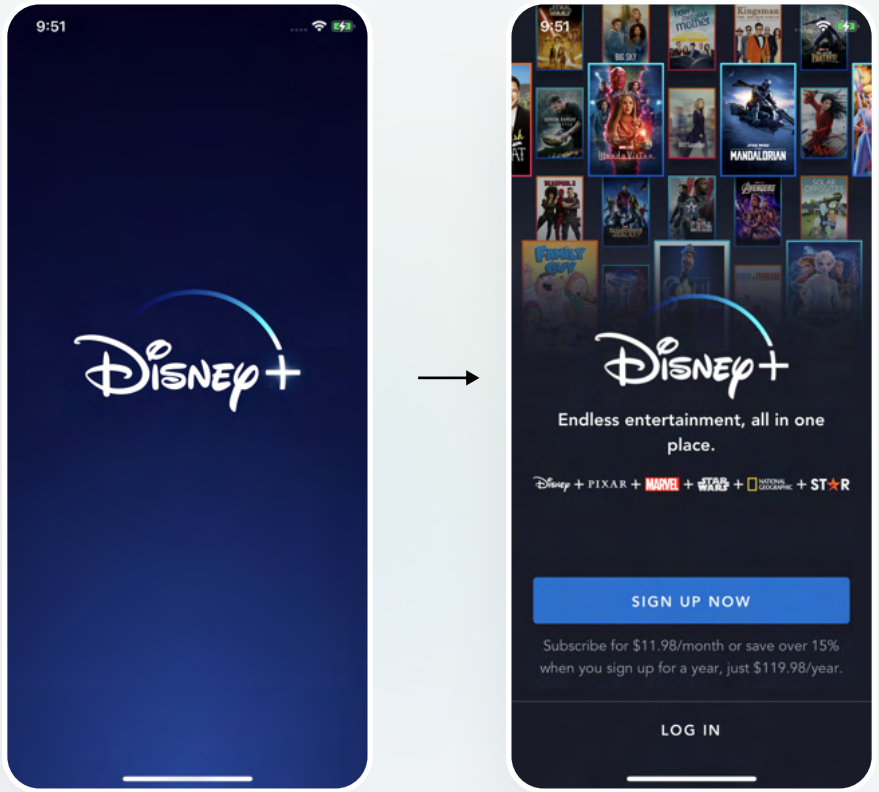
Launch screens should follow the 1-3-second rule; they shouldn't take longer than that to load. Otherwise, you'll leave your users frustrated and annoyed, stuck at a sequence of animation that's impossible to skip. Let's take a look at Netflix. Their app takes a while to load, but their launch screen and their signature Ba-Dum! sound puts you in the right mood from the start, and when the animation is over you're taken right into the app, ready to watch.



Take it up a notch

If you have enough time and budget on your hands, you can add some extra motion with your launch animation, transitioning into your onboarding experiences to awe your users. Don't go too crazy with it, and repeat the same pattern every time a user launches an app.

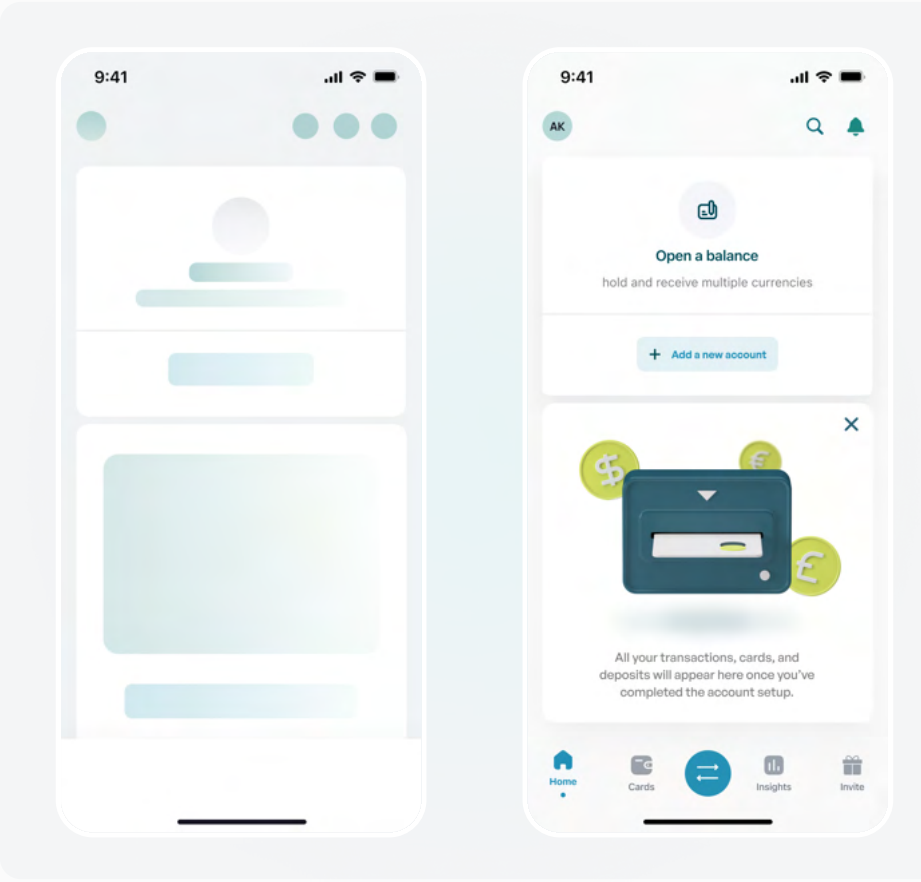
Make it a one-time experience when users first boot up your app. Disney+ is an excellent example of that.



Use a skeleton

If your application takes longer than 3 seconds to load, use a skeleton screen instead of a launch screen.

Skeleton screens show a skeleton of the final user interface, which informs the user that something is actually loading in the background and he's not just experiencing a frozen application.





Launch Screen

THE STRUCTURE

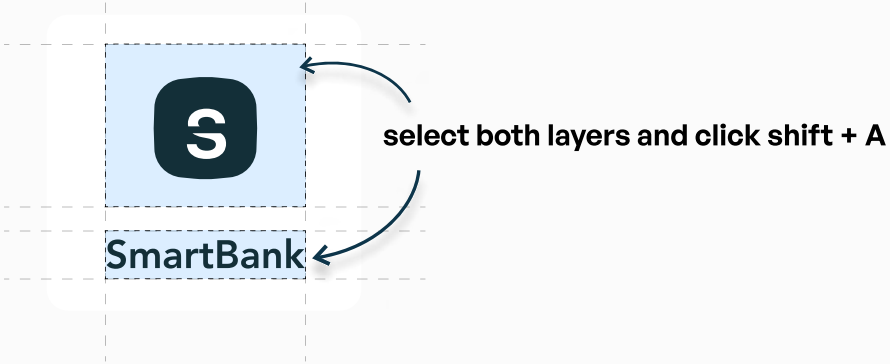
Launch Screen - the structure

The structure of this screen is relatively simple, and it's the perfect starting point for us. Let's talk about how you can replicate it yourself.

As mentioned previously, you should have your Practice files and your iOS 15 UI kit open at all times at this point. You'll be working on your empty frame and using the original on your left as a reference point.

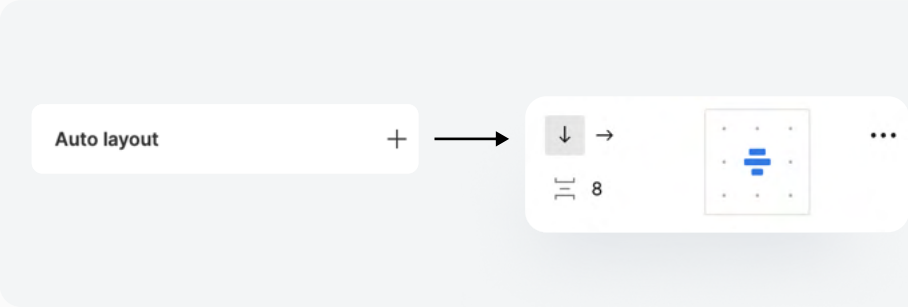
The resolution we'll be designing on is the 375x812 px (1px=1pt).





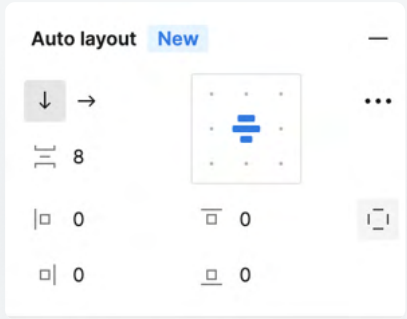
Select your logo and text layers and click "shift" + "A" to add auto-layout. We'll be using this shortcut a lot so memorize it. From now on, I will be referring to this action as "apply auto layout."

When your selection has auto layout applied, you will see a change on the right-hand side properties panel. Here's what it should look like 👇



If you've skipped the theory on adaptive layout, scroll back up a few pages and read up on iOS design basics. This is a piece of knowledge you should obtain when designing in Figma and being a better designer overall.

With auto layout applied to our selection, make sure everything is aligned to the middle, and the distance between our group elements is set to 8 pixels.



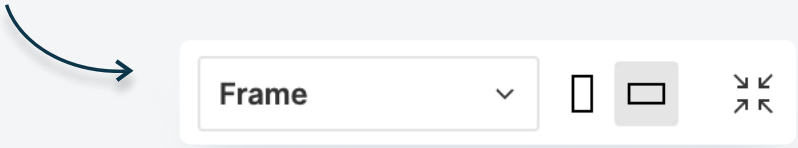
Let’s set our constraints now so that our design remains responsive when switching to different screen sizes.

There are a couple of ways we can approach this, so let’s discuss our options and how the design would differ if we chose each one.

I will present you with different options, but from there, moving forward, we’ll pretty much stick to one constraint setting for each of the interface element type.

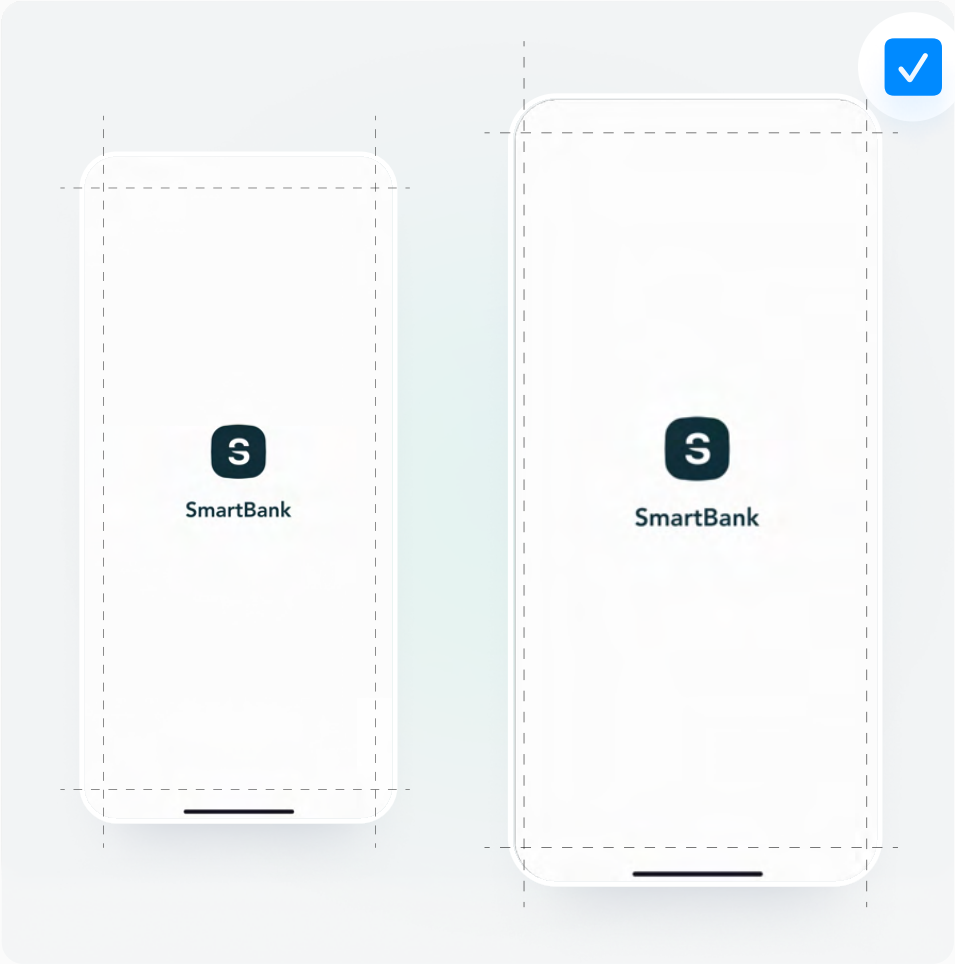
Design checklist you should follow

- [x] Turn on your layout grids and have your iOS UI Kit at the ready
- [x] Take your logo, image, and illustration files either from the components library or copy them straight from the screen next to your practice frame. Don’t copy entire designs though!
- [x] Select your parent elements and add constraints
- [x] Make sure your selection inside the text and image containers are set to fill-container
- [x] Check responsiveness by selecting the frame and clicking on the dropdown with the "Frame" label in the right Properties panel and changing it from our iPhone 13 Mini to iPhone 13 Pro Max.



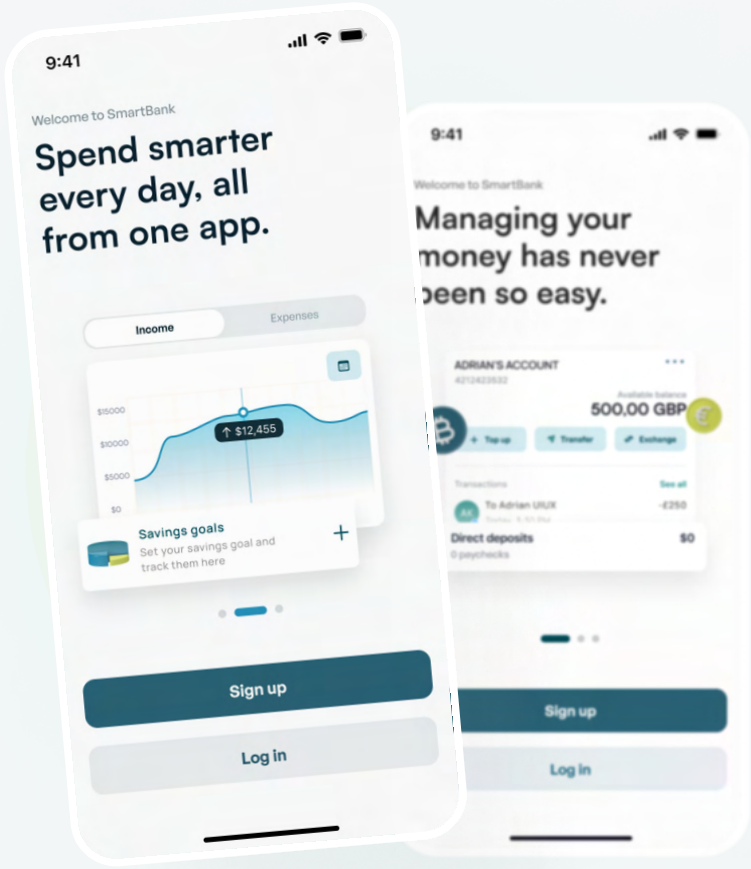
If you've successfully resized the frame and it's all working great, let's move on to another screen. If not, double-check the design checklist from the previous page and see if you haven't missed anything.

Ok, great, let's continue!



Chapter recap

- A launch screen can either be an image, graphic, logo, or an animation
- Launch screens should follow the 1-3-second rule; they shouldn't take longer than that load.
- Don't overload your users with crazy animations, but at the same time, don't make them look at a static empty screen.
- If you have enough time and budget on your hands, you can add some extra motion and add beautiful transitions.
- You should remember that at the resolution we are designing for, 1 pt equals 1 px. Points are a measure used for responsive scaling which you'll get to understand better when we talk about design hand-off at the end of this Design Workbook.
- Make sure you've completed the design checklist before moving forward!



Onboarding

THE INTRO

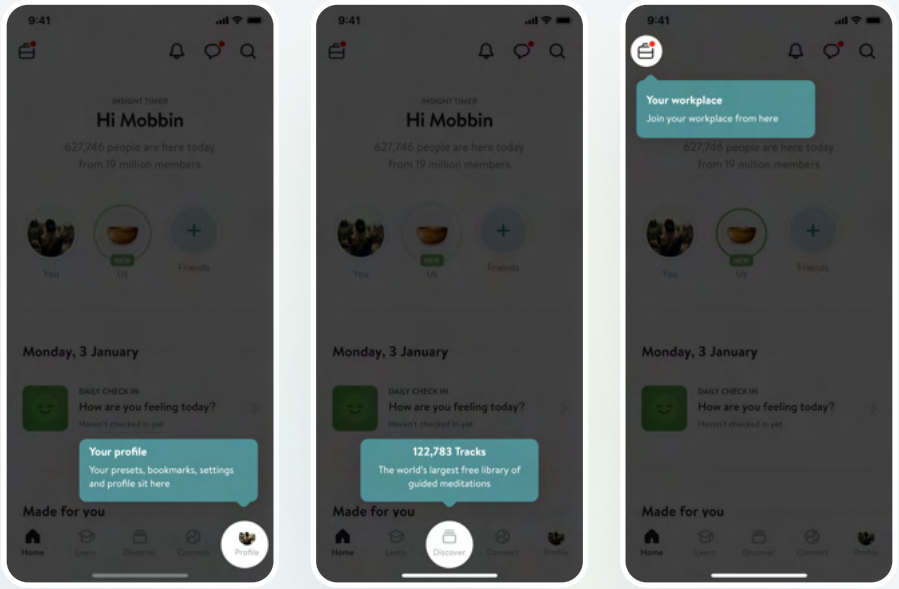
Onboarding - the theory

There are a few types of patterns used for the onboarding experience. Each of them has different uses and depends on various factors and a given context. You should pick the one that suits your app the best.

Here are the most common onboarding experience patterns:

Guided Tours

Guided walkthrough is a quick tutorial on the app's essentials to help users familiarize themselves with the interface and ease up the learning curve.

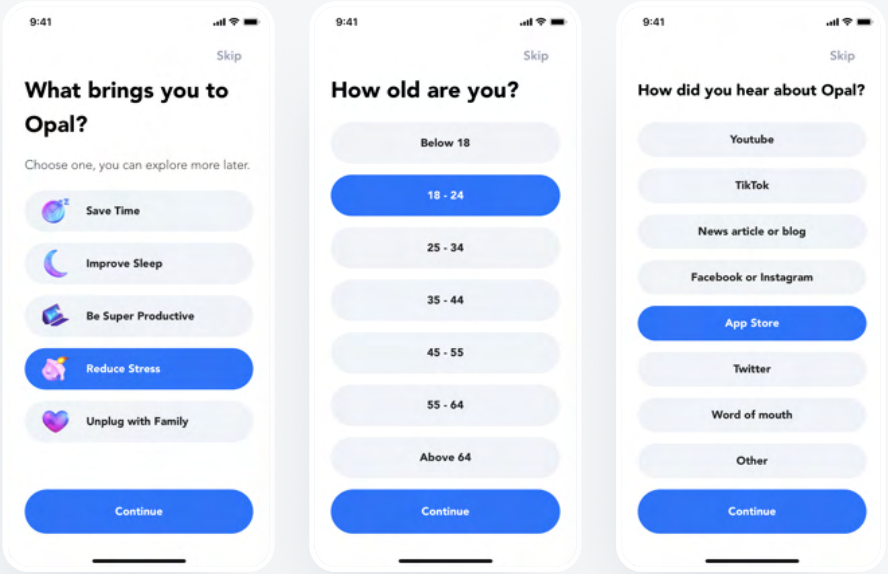


Insight Timer - Guided Tour

Personalization wizard

If your app is all about personalized experience, try adding a wizard to set preferences right at the start. It helps to provide a more immersive onboarding experience and engage your users more.

Just make sure to complete the customizer experience within a few steps and offer a skip button for users at all times. We don't want to make them feel stuck in a process with no option to skip and return to the account setup anytime later. We have to think about different use cases and journeys our users may be taking.

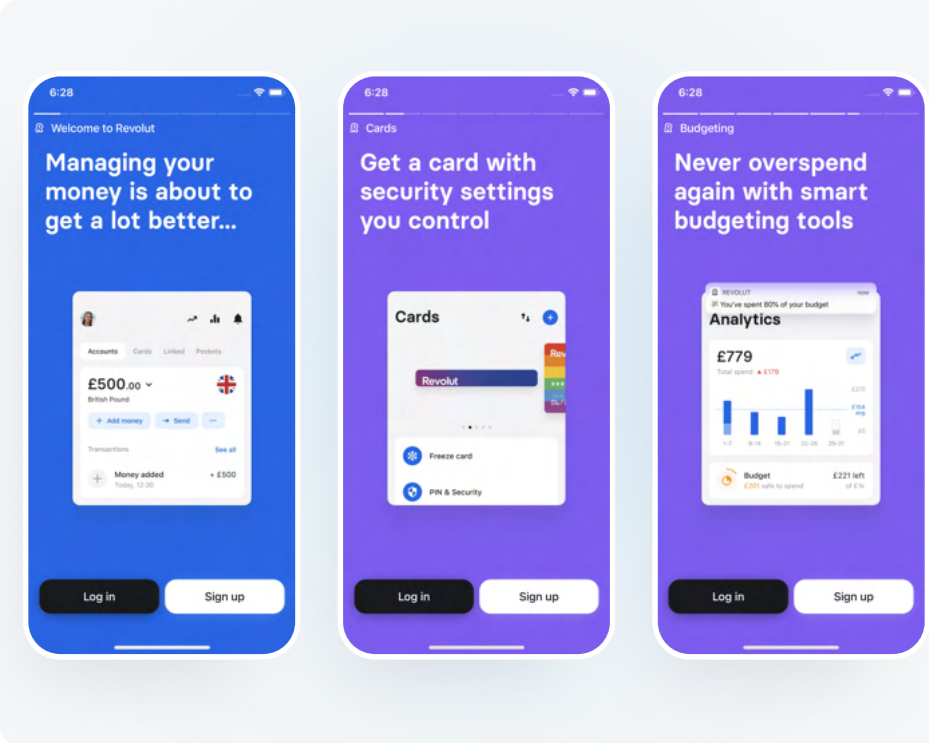


Opal Personalization Wizard

Walkthrough screens

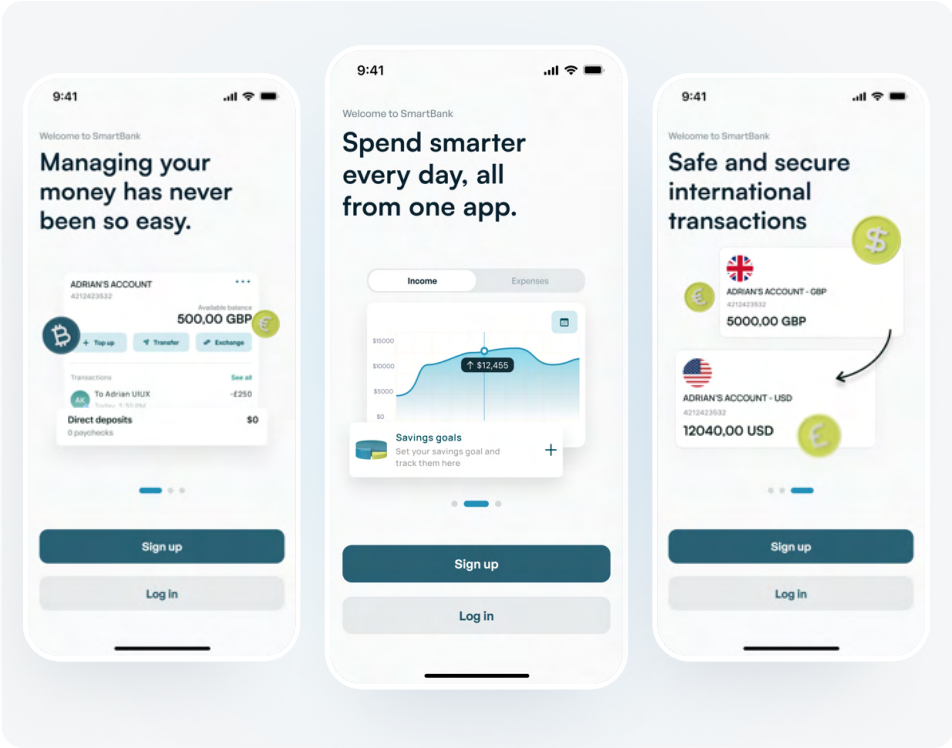
Walkthrough screens give users more context about the app's functionality. Instead of users landing in your application for the first time with no sense of which features or areas to explore first, in-app walkthroughs allow you to curate their experience and guide them through the workflows they should know about right away.

Explain what your app does, how it works, and what users can expect from it in one or two short sentences. Make it short and sweet.



Revolut Walkthrough screens

Our onboarding experience - Walkthroughs



Looking at our options, we decided that the best approach would be to go with the walkthrough screens. Our target audience is already well familiar with the concept of fintech banking. On top of that, our app features a pretty long onboarding and verifications process being a Fintech app, so having a guided tour wouldn't be ideal for introducing the app's functionalities.

Instead, we show them three reasons they should explore our app further with two CTAs to sign up or log in to continue.

Illustrations

Digital products now include all sorts of illustrations as a standard feature. They are, without a doubt, not only required to set the visual tone for your product but are also used to convey a story. This isn't something you should design as you go, but you should put a lot of thought into creating graphics for your digital product.

There are a few tips you should follow when designing and choosing your illustrations:

Most importantly - choose a style and stick with it. If you decide to go with 3d illustrations, do not mix them with flat outline drawings.

While choosing your illustrations, think of the message you want to convey through your app. The images must match the character of the product.

Choose your brand colors as your illustration colors. Don't mix too many of them as the product will lack consistency and look like you've just patched up a number of random images that don't go well with each other.

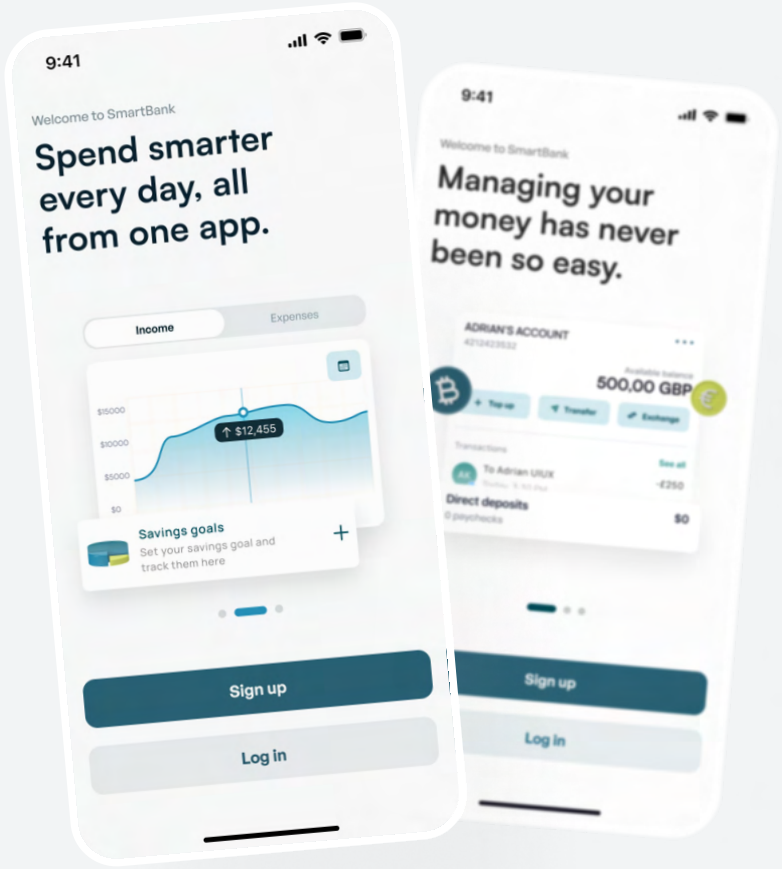
Use pre-made illustrations - look up ui8, Freepik, Storytale, or just browse through my Instagram and look for free illustration sites. I share a lot of them in my carousels!



Illustration pack from UI8.net

Suppose you're looking for something more original but can't really draw and model your own illustrations. In that case, you can look for illustrations libraries, download a few that look similar, and modify them slightly. Change the color, so they all match each other, mix them, take some aspects from the composition, and move them around.

You can really create commercial-worthy graphics by using the elements uploaded by designers in our fantastic community.



Onboarding

THE STRUCTURE

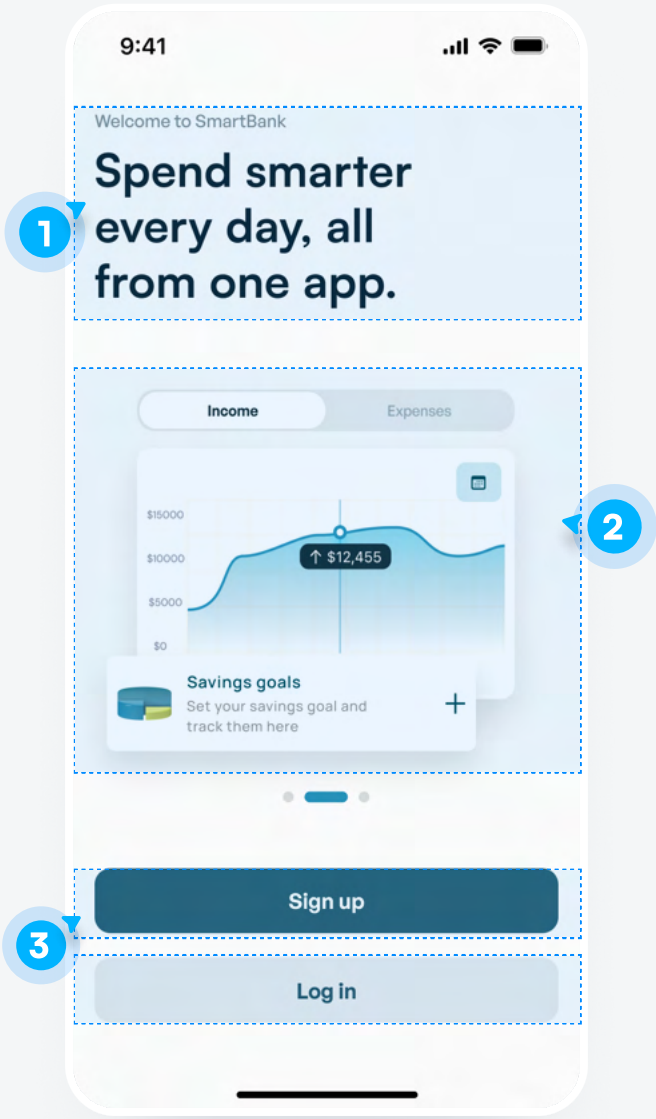
Onboarding - structure

Whenever we design, we should look for repeatable design patterns. To make it easier for ourselves, we should create our first screen for onboarding and then duplicate it twice, replace the text, and update the images. It's the easiest way to work.

What can we do to make it even more automated? That's where the reusable components come into play. Let's analyze the screen and see what it consists of.

The structure of an onboarding screen

1. A big bold title with a welcome text
2. A 3d illustration with our brand colors
3. A sign up button to initiate the sign up flow
4. A log in button to let users login with their previously created account

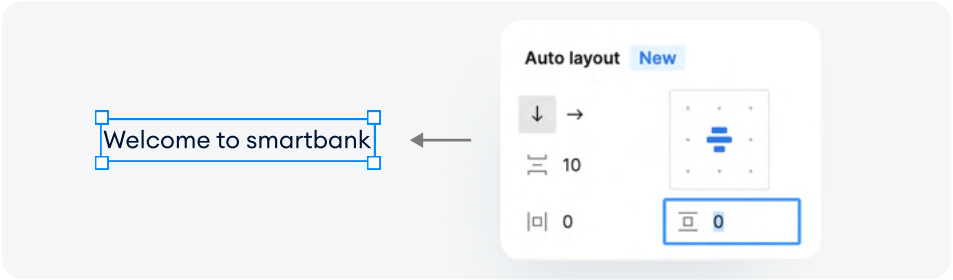


1.1.2b - Onboarding 2-3

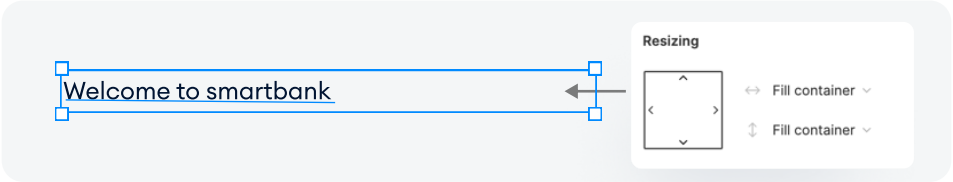
1. Text Container

Our **text container** will be repeated on almost every screen throughout our app, so our first order of priority will be to create our first text component.

Add your first “Welcome to SmartBank” text layer to our practice file frame labeled “Onboarding 1/3”. Apply auto-layout (Shift + A). Change padding value to 0.



Increase the layer's width to 343 px - our maximum safe margin width (we want to leave 16 px on both sides). Click on the text layer and change the Resizing settings to fill container for both dimensions.

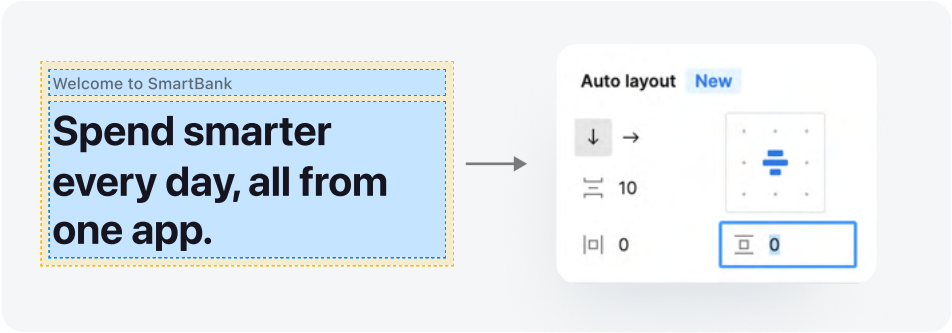


Make sure to apply our text style from our style guide - I chose **Bold/Footnote 13px** for description with 60% opacity.

We’re halfway there.

Add a new text layer below our description. Change the style to “Display” and position it below. Select it, apply auto-layout (Shift + A), and remove the padding.

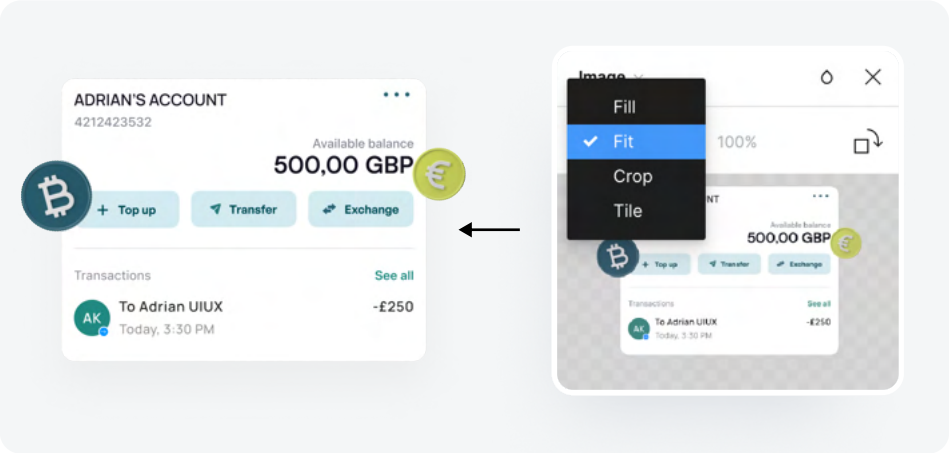
Now select both text layers and click “Shift + A” once more. With both of them selected, you should be able to edit the distance between them. Set the space to 8 pixels.



Now having this text container selected, make a new component. We’ll copy-paste it on our other screens to speed up our work. Remember, you can always find the components on our “Components” page in the provided file.

2. Images and illustrations

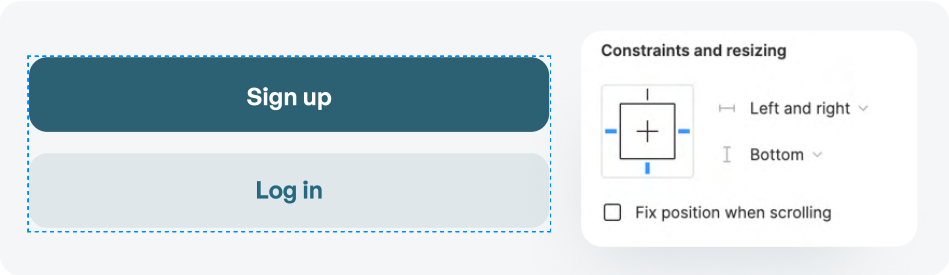
Next up is our app graphic. You don’t really need to do anything here except apply constraints - Left and Right for horizontal and Center for vertical boundaries. We’ll also need to change the image resizing to **“Fit”** so our image will always fit within the boundaries we set for it and remain responsive no matter the resolution.



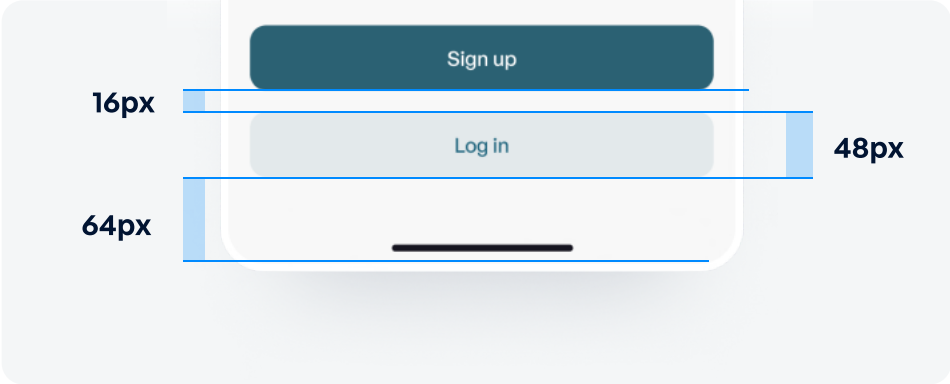
The same properties will apply to all images throughout our app so remember these settings well.

3. Buttons

Let's break down our button design now.



Remember, we created our buttons before as an example (in the "Ready, set, Figma!" chapter) and have them ready as components. Navigate to the components page and copy the primary and secondary buttons to our frame. Position our buttons 40 px from the bottom of the screen and leave 16 px between the two.

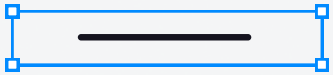


Leave 16 px space between the two buttons, select them both, and apply auto-layout. Set constraints to **Left**, **Right**, and **Bottom**. Check if your design is responsive by changing the frame size.

Remember that you should always design in the practice file. I have organized it to make it as easy for you to design as possible. When you can't make something work, just look at your left screen containing the original designs and click on every layer in each container to see the "Constraints" and "Resizing" settings. This is where all your answers should be. That's how we learn - by directly replicating other iOS app designs.

4. Home indicator

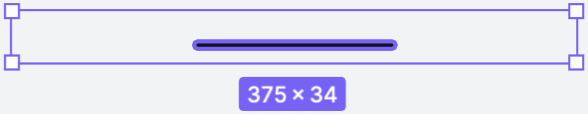
Lastly, we have our home indicator. I have already added one to your practice frames, so you don't have to replicate it yourself repeatedly, but you should still click on the selection and see what "Constraints" and "Resizing" settings have been set on it.



When designing for iOS (or Android), you should consider all the native system elements, like status bars, controllers, and home indicators. With the rise of full-screen mobile devices with no physical buttons, it's important to give users a visual indication of a home button that they can swipe up to at any time.

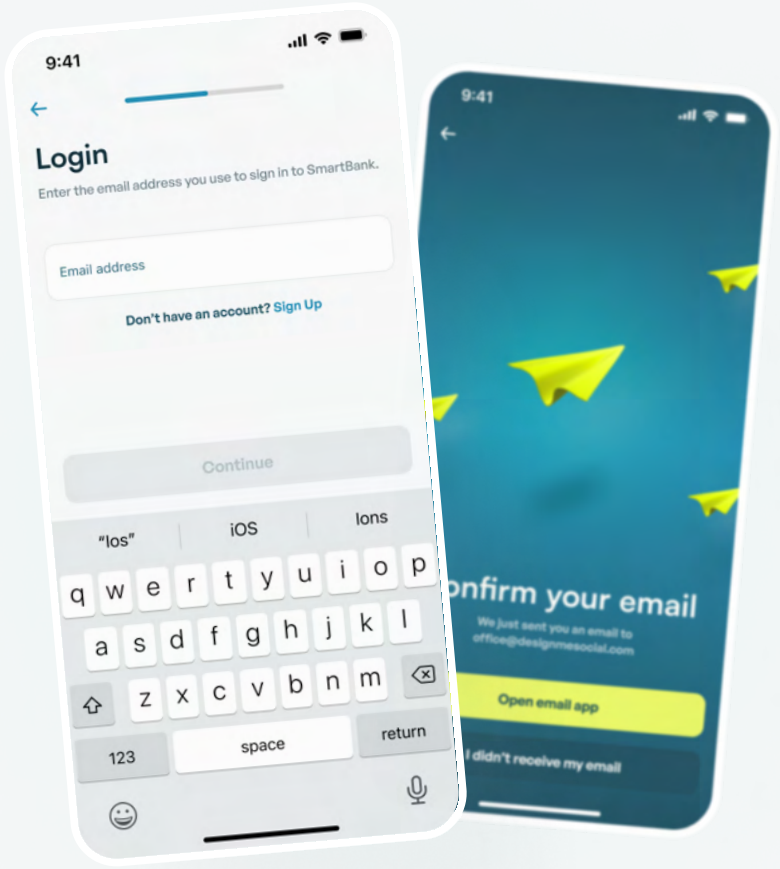
For our home indicator, we only need to set the constraints **"Left and Right"** and **"Bottom"** so it sticks to the bottom of our frame and retains the same spacing values on both sides when being resized to different devices. Remember to make it full-width (375 px in our example) and 34 px high just so we can leave enough safe margin for our designs.

When constraints are set, you can copy and paste it on all your practice frames; that's one element less we need to design.



Chapter recap

- You can use a few types of patterns for your onboarding experience, and you should pick the one that suits your app the best.
- A guided walkthrough is a quick tutorial on the app's essentials.
- Personalization wizard is an immersive onboarding experience.
- Walkthrough screens are used to give users more context about the app's functionality.
- Use reusable components for repeatable elements like text, buttons, and containers.
- Make sure image resizing is set to fit so our image always fits within the boundaries we set for it.
- Make sure to apply text styles from our style guide.



The Sign-up

THE INTRO

Sign up - the theory

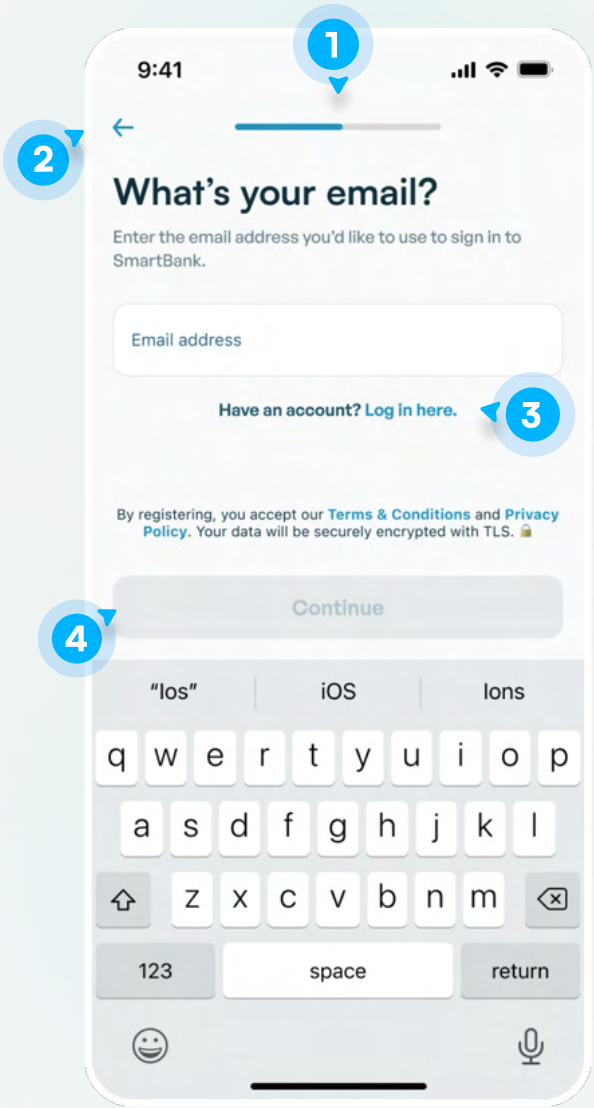
The sign-up process should be simple and intuitive and lead the user in a way he won't feel overwhelmed or confused.

When it comes to banking and fintech apps, the sign-up and verification process is probably one of the most complicated.

There are necessary steps to verify our email, phone number, residence, and identity. On top of that, we need to set up things like verification codes and biometrics. Our app deals with sensitive data, and security should be our top priority.

The building blocks of a sign-up screen

1. A progress bar to indicate how many steps are left
2. The user must have an option to return to the previous screen.
3. The user must have an option to switch to a login screen if needed.
4. The user is not able to proceed further without giving out the email address first.



1.2.2a - Email 1-2

We initiate our onboarding flow when we first launch our app and click the “Sign up” button on the Tutorial screens. First and foremost, we want to collect the user’s email. Not only to send them reminders about an unfinished setup if they skip out on any step of the process but also to let them recover access to an account if they ever lose it.

Ok, apart from the input field to collect emails, what else do we need on the Sign-up screen?

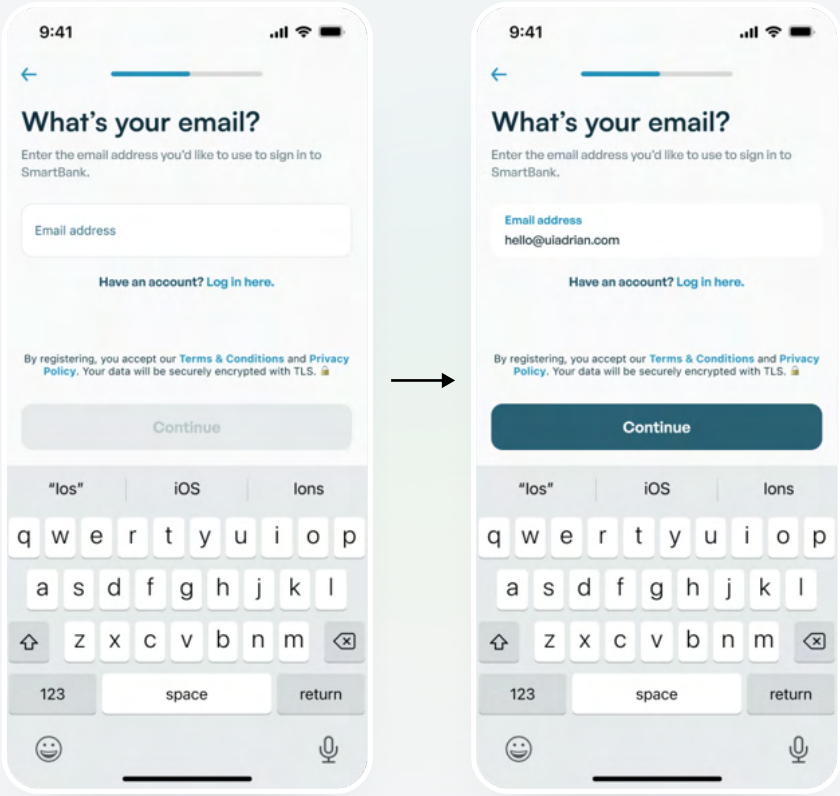
Back button

If someone accidentally clicked on the sign-up with an email, we should allow him to go back to the previous screen to change the preferred sign-up option. We also want to prompt them to log in if they have an account and mistakenly chose the wrong button.

Having a navigation element to move between screens is crucial for a better user experience.

Continue button

If the button is in its disabled state, that clearly indicates that the user is blocked from moving forward, but that may not always be obvious. Disabled button states are usually displayed at reduced opacity or not displayed at all. Instead, we could show the button only when the step is considered complete, i.e., when the user submits a correct email address.



1.2.2a - Email 1-2

1.2.2b - Email 2-2

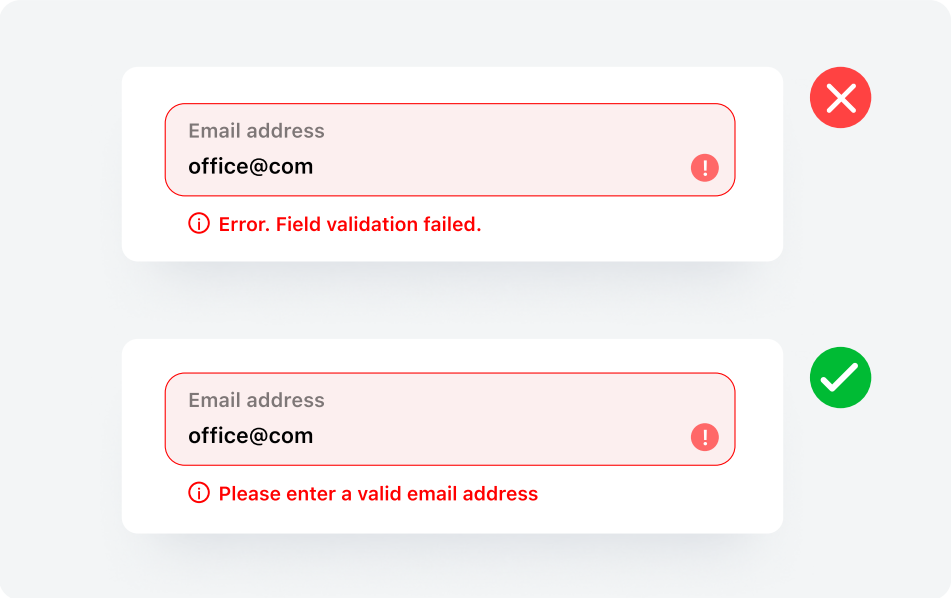
Having the button disabled may mean that the data is missing or there was a mistake made somewhere in the process that prevents the user from moving forward.

Let’s make that clear by adding another visual cue.

Error state and messages

Whenever our users make a mistake, we need to make sure that they understand the error that's been made, and for that, we'll use error states and messages.

When things go wrong, and we can't let our users advance further, don't let them wait for a page reload to display the error message. Make them instant.

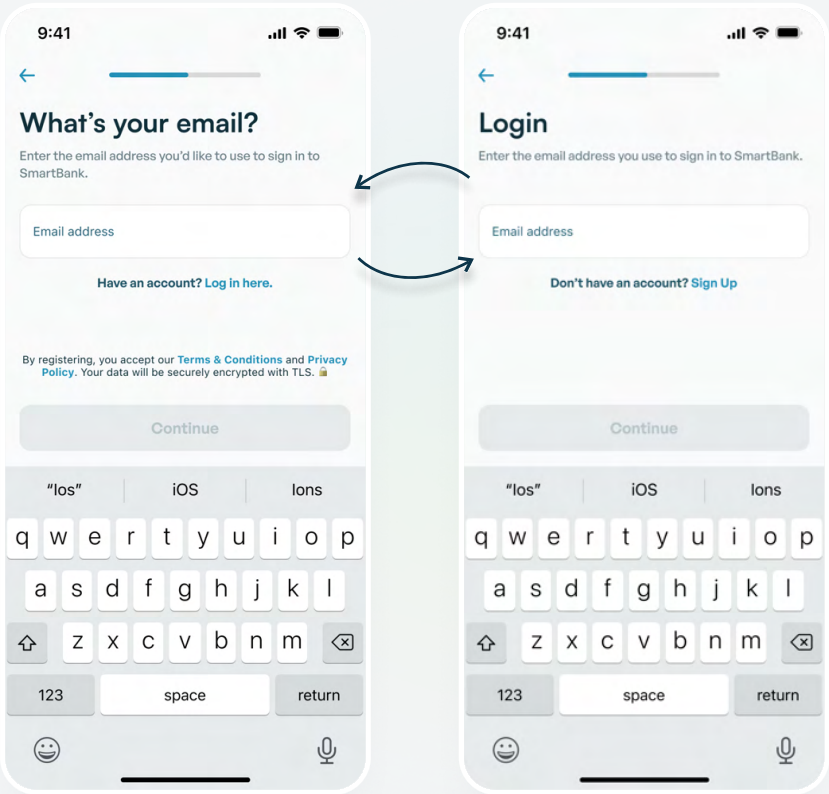


For invalid inputs, we should show error messages almost immediately to let users fix the mistake before moving on to another step. Make it prominent enough to see it, use clear language not to confuse them and tell them exactly how to fix the mistake.

Login page

We should allow for easy switching between "sign-in" and "sign up" screens. If someone clicks the wrong option, you want to make it stupid simple for them to switch from the registration flow to a login flow and vice versa.

We can add a link text under our email field to do that.

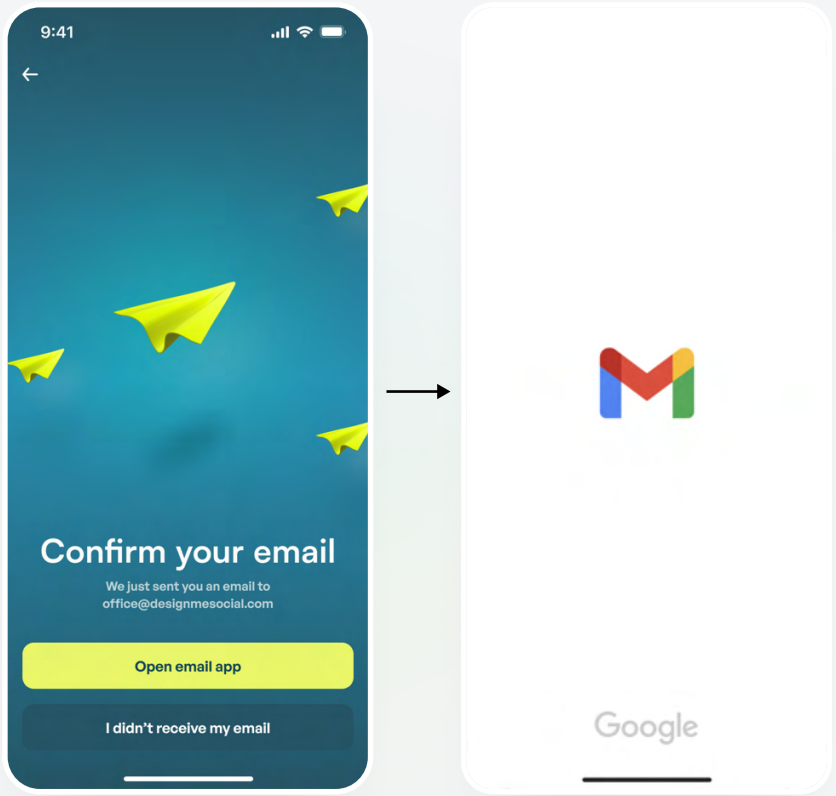


1.2.2a - Email 1-2

1.2.1 - Login

Confirm email flow

Once an email is provided, we need to ask our users to confirm it by clicking on the link attached to a confirmation email. We've also added descriptive text highlighting the email used in the sign-up process. Clicking on the "Open email app" would open your default email app, and clicking on "I didn't receive my email" would re-send it again.



1.2.3 - Confirm your email

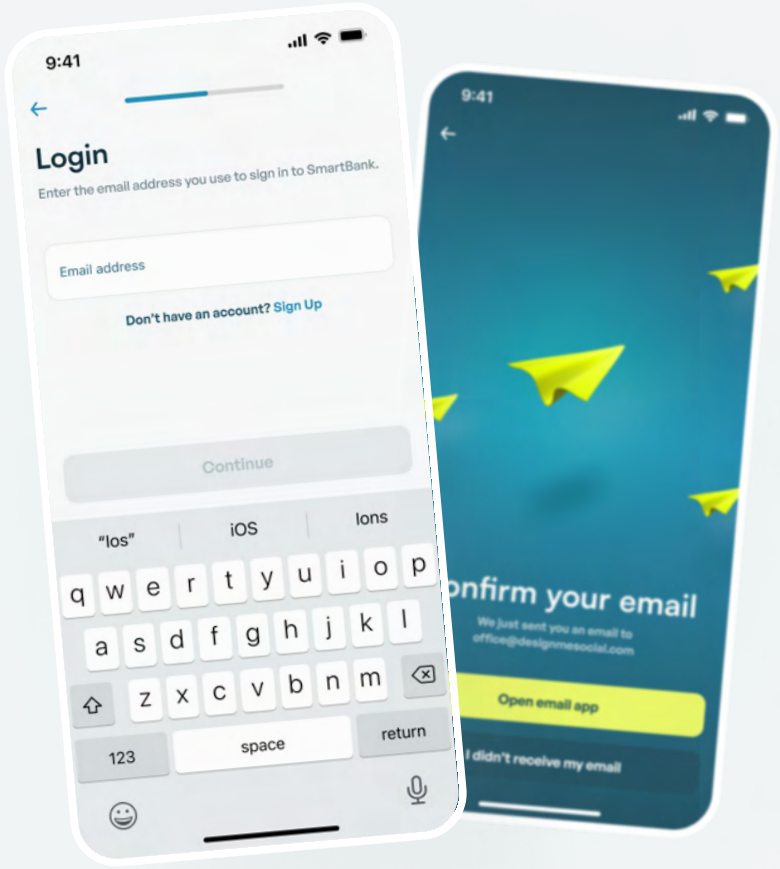
Open email app

Confirmation successful

Clicking on the confirmation link should take our user back to our app. Let's play a quick SmartBank loading animation and put our user in the account setup flow next.



That's it for the sign-up flow. We've got our user's email now, and even though that's enough to onboard a person in a regular app, we're dealing with finance, a niche regulated by many government-owned financial institutions; therefore, we'll need to verify more than just an email. Let's talk about our design structure now, and then we'll break down the Account Setup.



The Sign up

THE STRUCTURE

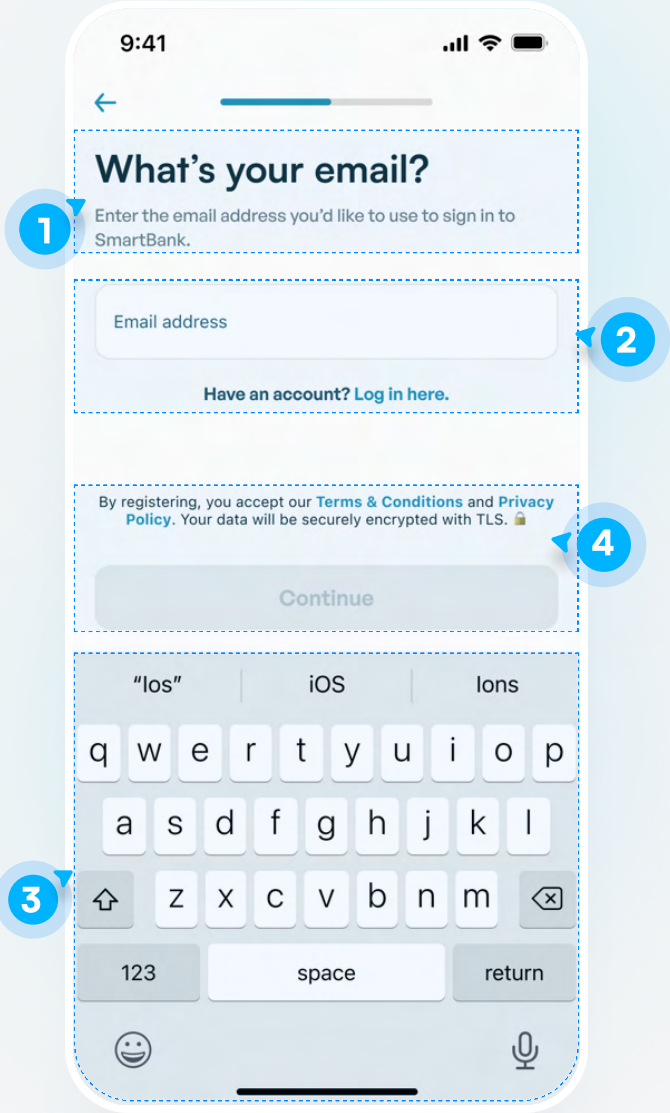
Sign up - structure

We should look for repeatable design patterns we can componentize whenever we design new screens. We'll have the same label and description container repeated across our entire app so let's create one and reuse it later. The same applies to our input fields, buttons, and iOS native views - we don't have to recreate them every time. Make them into a component and copy-paste them from screen to screen or look through the library of pre-made iOS components to quickly create native elements.

But first, let's break down the layout structure of our Sign up screen.

The structure of a sign-up screen

1. Text container - repeatable component
2. Input field - repeatable component
3. iOS Keyboard - repeatable component
4. Acceptance text and a Button - repeatable component

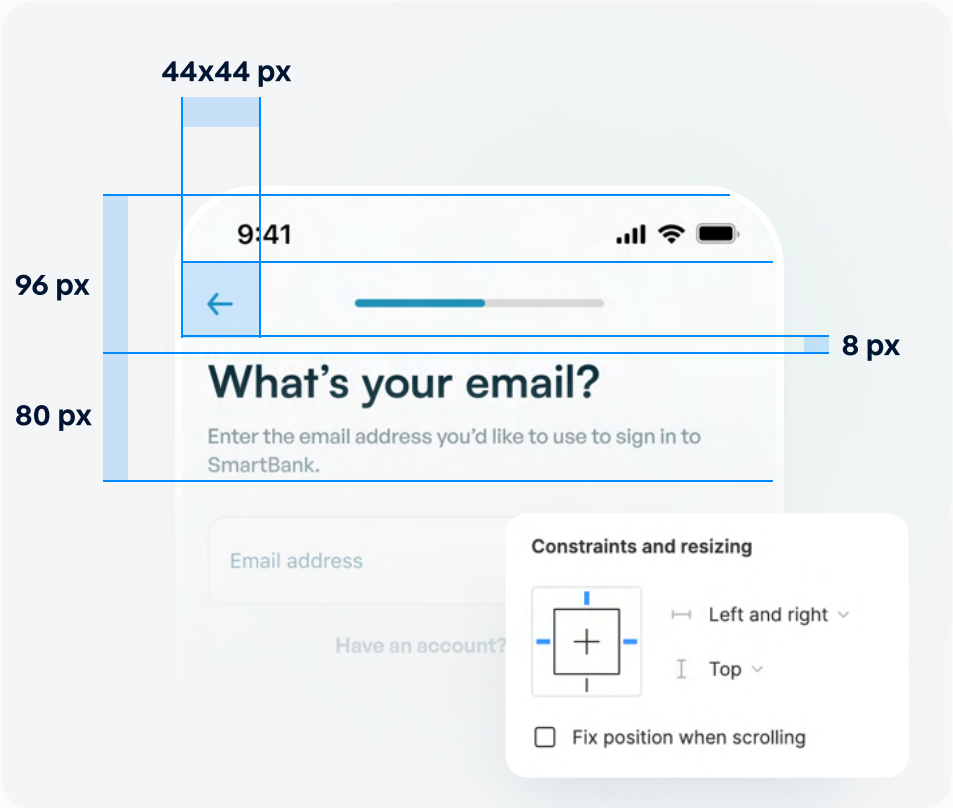


1.2.2a - Email 1-2

1. Text Container

For our text container, we can reuse the same text component. Copy it from our components page and place it 16px from the left and 96 px from the top, and around 0 px from the back button's frame (44x44 touch target).

Change the text alignment to the left, and reverse our layers by clicking up and down on the keyboard with one of the text layers selected.



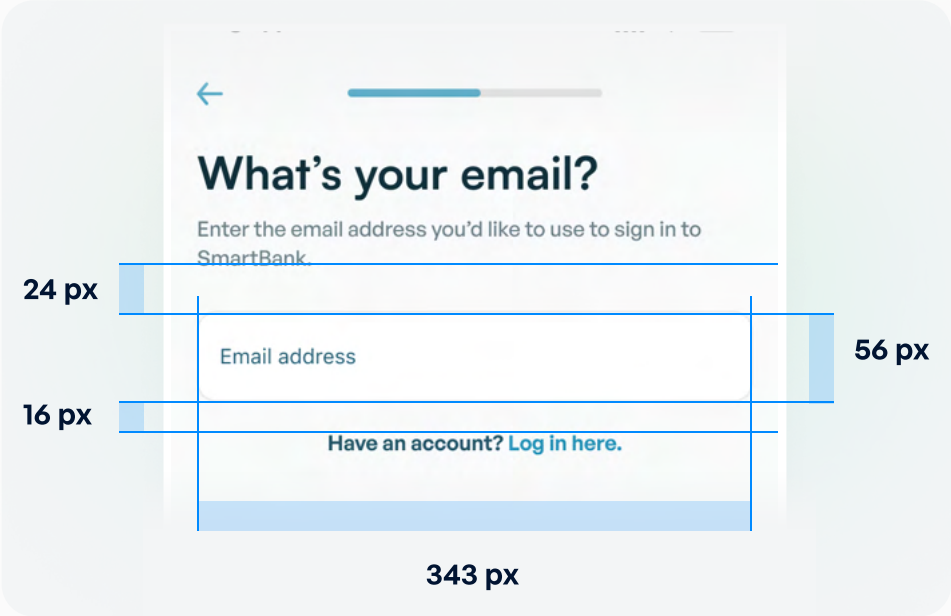
Set the layout of the top part once and copy and paste it everywhere now moving forward. Let's make it easy for ourselves. Label, description, and a back button are patterns repeatable in many iOS app screens. We should make a component out of it and reuse it every time we need it.

Remember to set the constraints to **Left**, **Right**, and **Top**. Double-check the settings on the original screen in your practice Figma file if you're in doubt.

2. Input Field

To create our input field add a text layer (click "T" on the keyboard"), make it a Footnote 13 px style, and write down "Email address." Copy the layer and place it 4 px below the Email address. Select both and apply auto-layout ("Shift + A"). Drag to manually resize it to 343 px width and 56 px height. Add a fill and change it to "White," round the borders to 8 px.





Remember that you don't have to recreate every single component described here. All of the components that I mention here are available on the "Components" page in our Figma file, and you might as well copy them and paste them into your practice frames.

Being a good designer, you should work with components to make your life easier. There are situations where you need to adjust a single element repeated on several screens. Without a component in place, you'll have to do it all manually. If you, however, use the component for repeatable patterns, you'll only need to edit your master component once, and it'll update all throughout your designs.

Just remember to go to each layer within the selected component and see the constraints and resizing values set on each element. By analyzing work and replicating it yourself, you'll learn the fastest. If you're provided with pre-made components, you should use them. Having a sound design system in place and knowing how to use it is extremely helpful to simplify your design work and design interfaces twice as fast.

*Not sure how to use design systems? Don't worry; we're working on releasing a beginner-friendly design system with clear documentation and video guides on how to use it to design faster. You can check it out on our new project's page **@ui_lab** and my Instagram stories.

Let's move to our next element!

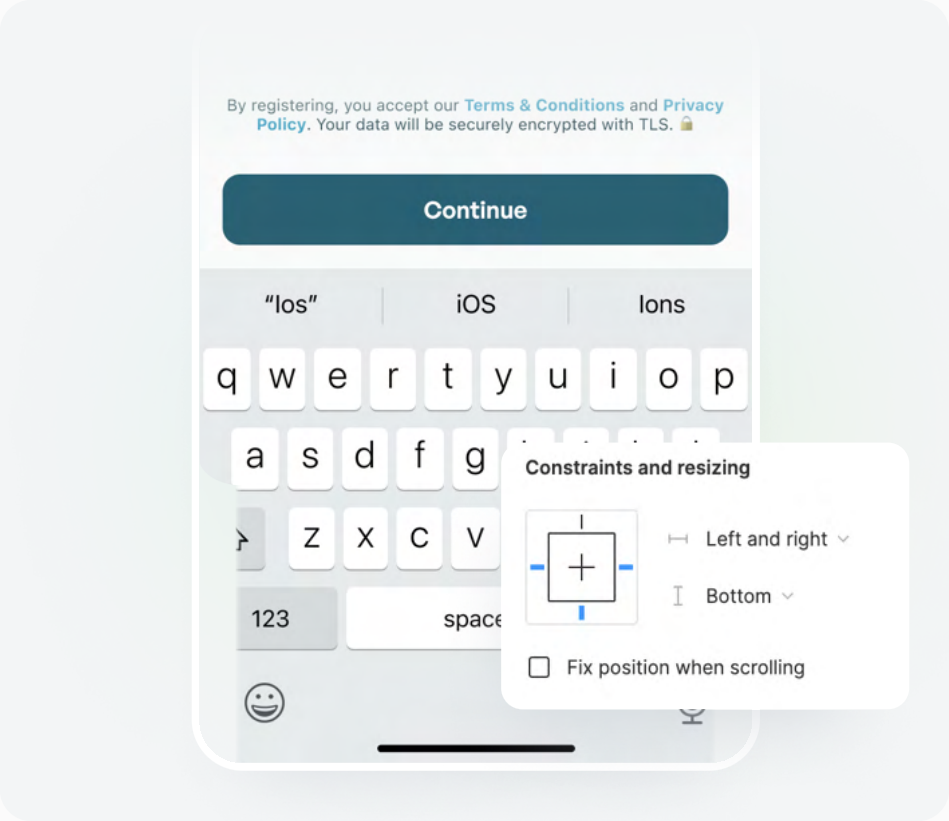
A design system is a set of rules, guidelines, and best practices that include colors, shadows, icons, menus, forms, typography, and every interface element you can imagine in all states, shapes, and sizes.

It's a step up from a style guide and goes much more in-depth about what elements to use and where.

3. iOS Keyboard

Next, we have our iOS keyboard, and that's the easiest part to create. For iOS native views, we should always look at the pre-made components from the iOS 15 UI Kit by Joey Banks.

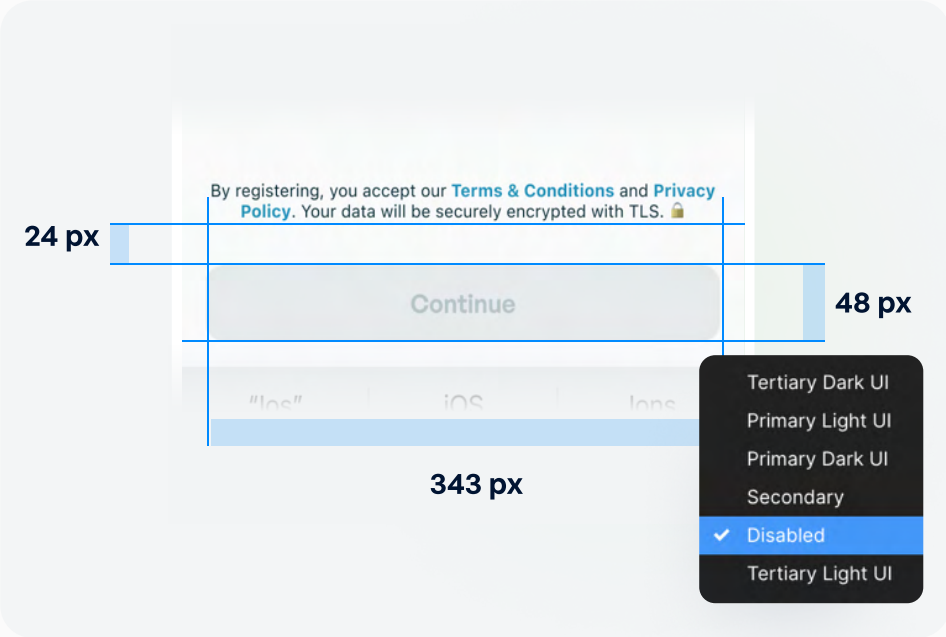
Copy it straight from that Figma file and paste it onto your practice frame. Align it in the middle and to the Bottom. Resize to fit the frame if needed. Add **constraints** - **Left**, **Right**, and **Bottom**. That's it!



4. Acceptance text + Buttons

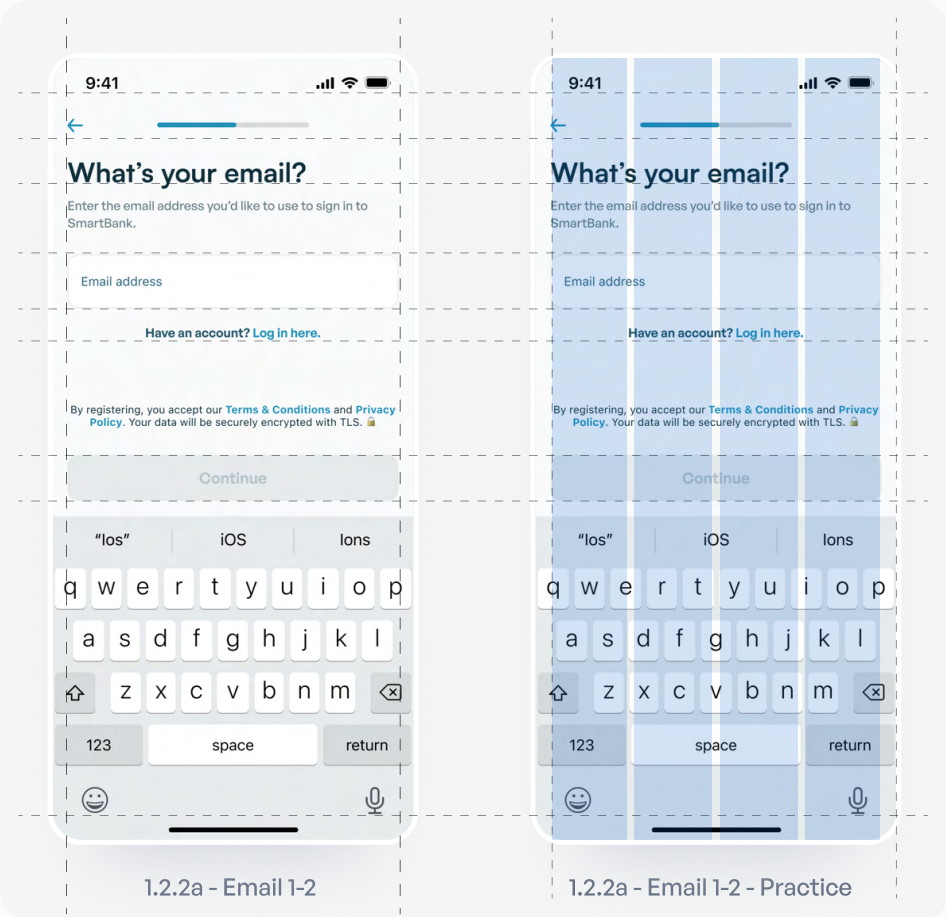
Lastly, copy our button from the "Components" page and change the Variant to "Disabled." Position it 16 px from the iOS keyboard.

For our acceptance text, we can simply add a text layer with a width of 343 px and set the **constraints** to **Left** and **Right**, and **Bottom**. Place it 24 px above our button container.

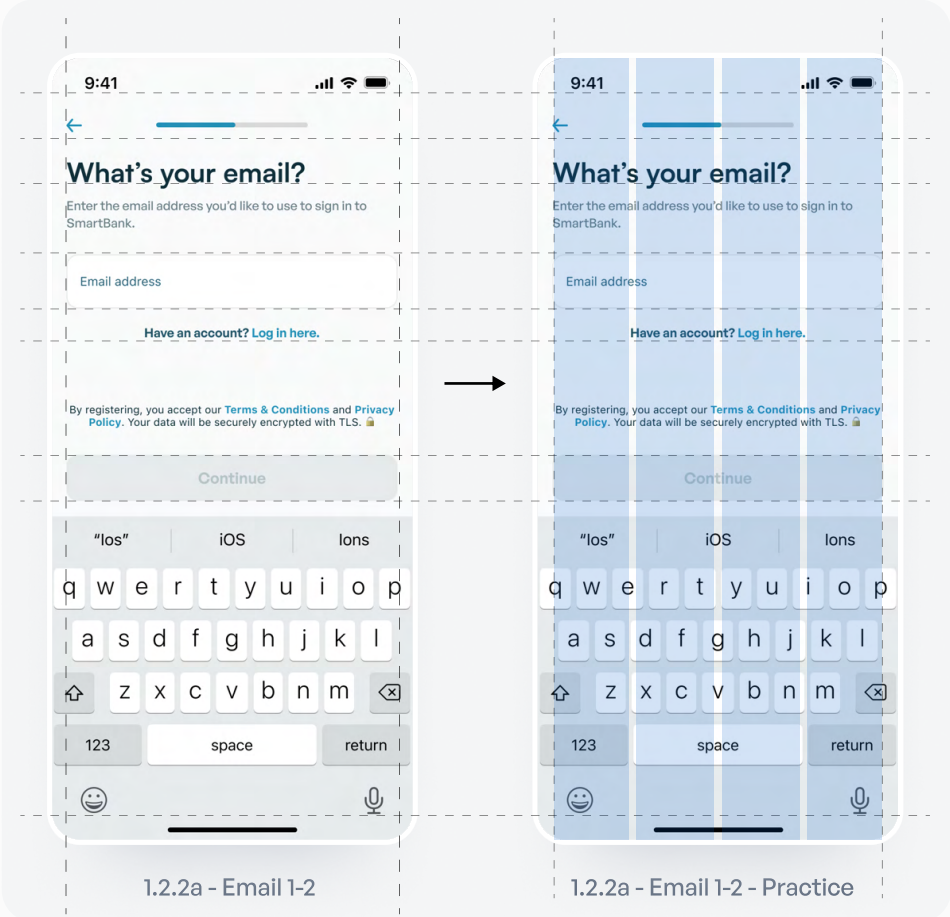


Make sure your button has **Left** and **Right**, and **Bottom constraints** active. If in doubt, check the original screen on the left of your practice frame and see the properties of every single element.

We're ready to move on to the next step.



As always, compare your practice frame with the original. Turn on your grid (**"Control + G"/Control + Shift + 4**), and double-check the responsiveness, padding, margin, and text sizes. Draw a couple of guidelines (Shift + R) to add rules and drag them in to help you align your containers the same way (play the **video** again if you're not sure what I'm talking about). Analyze everything when you're done, and let's move to another step!

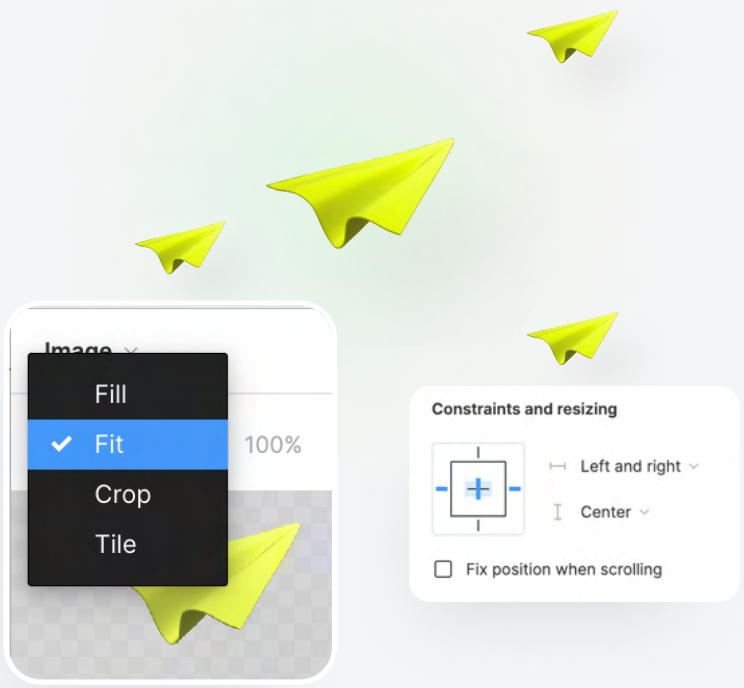


For our Login screen, we just need to copy our "What's your email" frame and replace the text in each text layer. Remove the acceptance text since it's already been accepted. Change "Continue" to "Log in," and we're good to go. Don't forget to rename the layers to reflect the text changes.

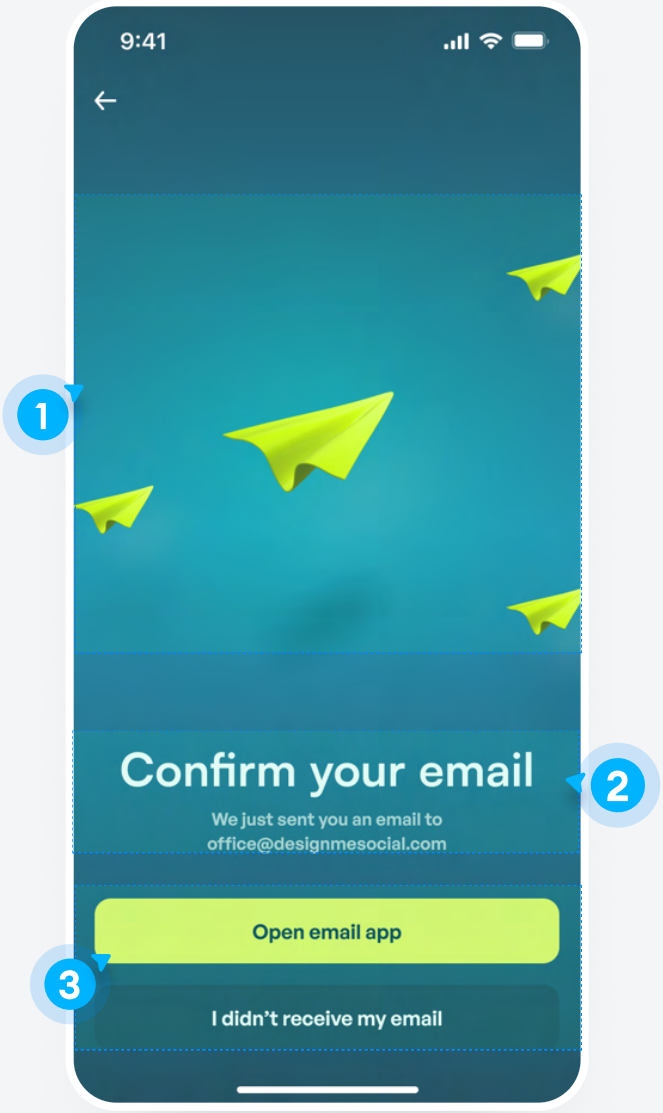
Confirm your email

1. Image

Make sure your image is set to “Fit” and your constraints are set to “Left and Right” and “Center.” That’s all we need to remember to set up our image layers; there is no need to adjust anything else. Let’s move on to our text container now.



Confirm your email Illustration

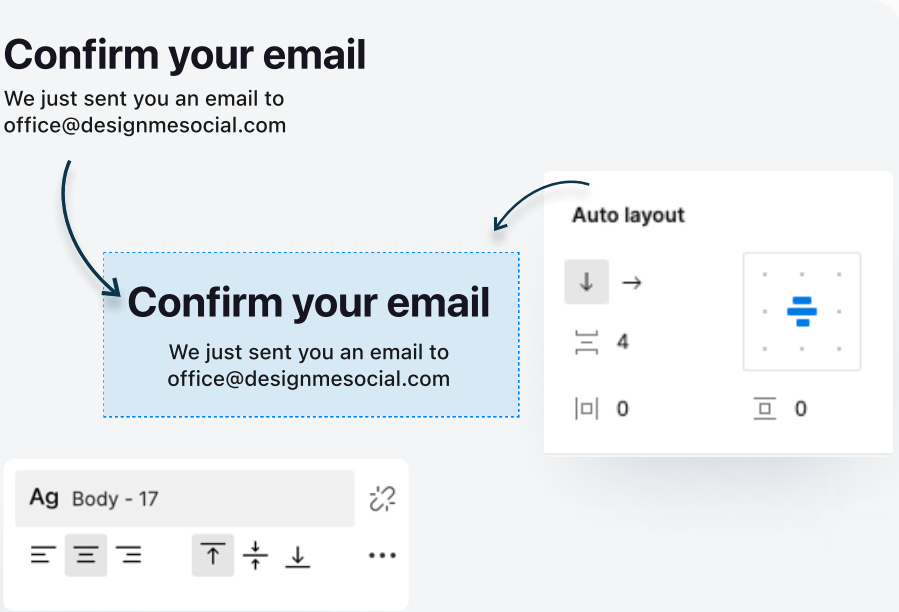


1.2.3 - Confirm your email

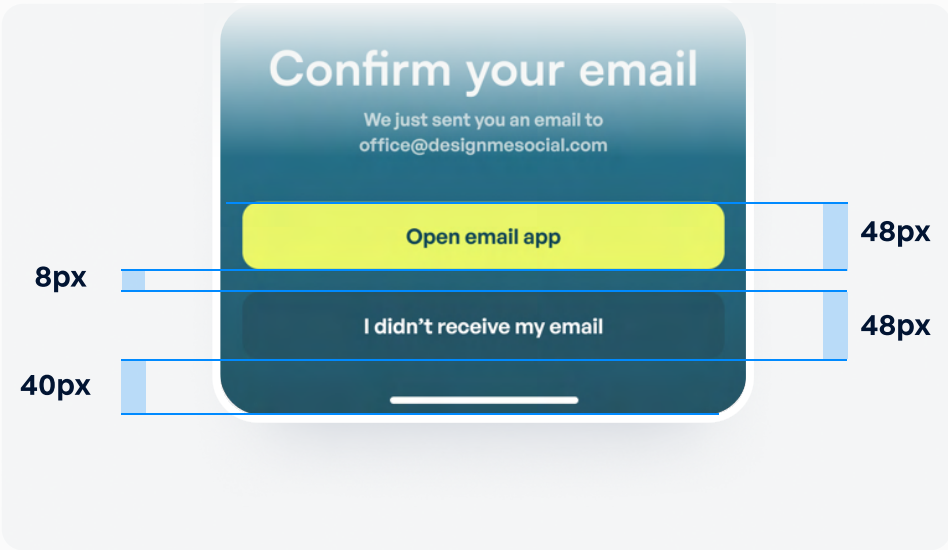
2. Text Container

We can copy one of our text containers from the “Components” page and start adjusting the style.

- Change the alignment of both layers to the middle.
- Change your description text style to Regular/Body - 17 px and increase the opacity to 100%
- Select your text container and change your auto-layout alignment to the middle.



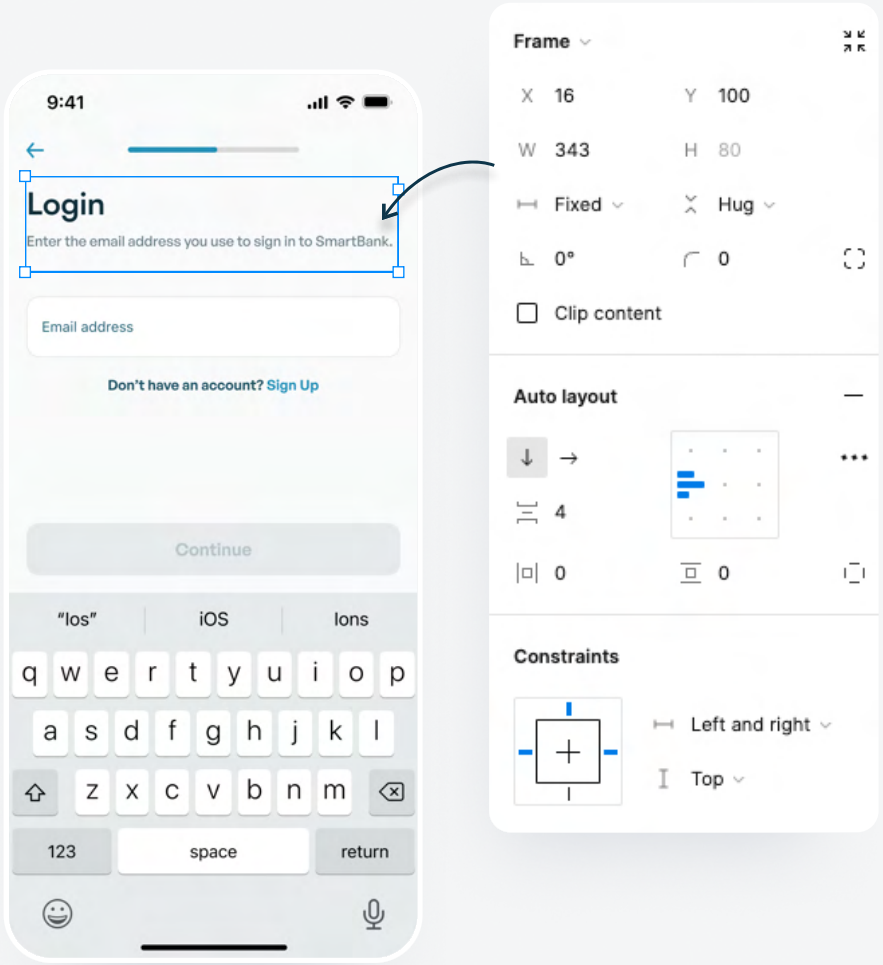
3. Button group



Let’s copy our buttons from the **Sign-up flow** and do some manual adjusting. Change the text from “Sign up” and “Log in” to “Open email app” and “I didn’t receive my email.” Remove the fill from our secondary button - let’s make it a link button instead since it doesn’t have the same importance as our secondary CTAs.

When you can’t make something work, just click on every layer on the original screen frame on your left and compare it with your practice frame on the right. Look at your group containers, see the “Constraints” and auto layout “Resizing” settings.

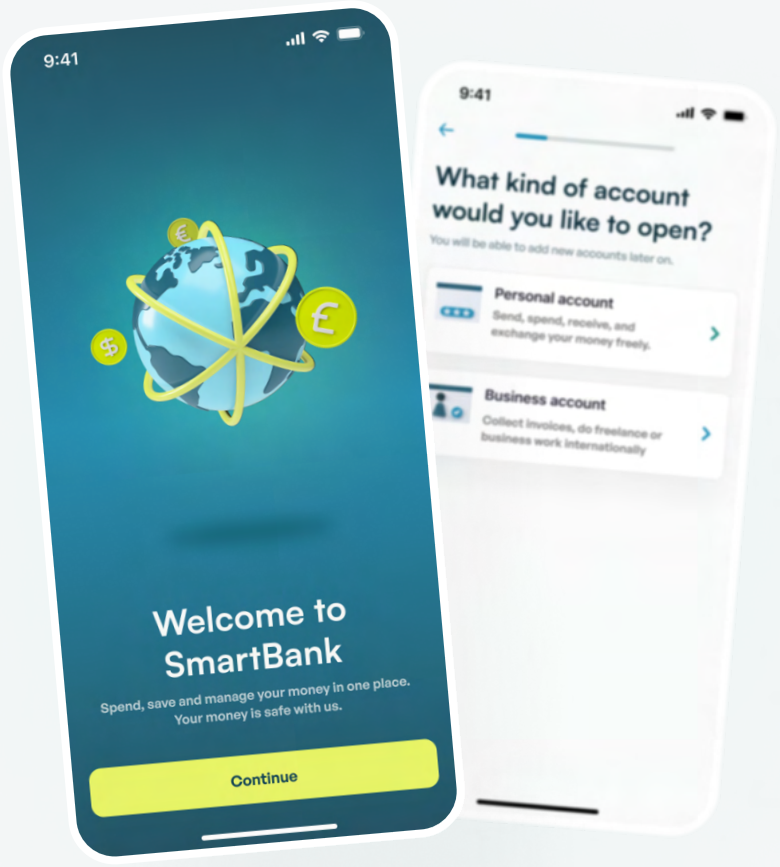
Since the latest Figma update, the “Resizing” properties have been moved up near the width and height parameters. This is where all the answers to responsive problems should be.



1.2.1 - Login

Chapter recap

- The sign up process should be simple and intutive and lead in a way that user won’t feel overwhelmed or confused.
- Add an option to switch between Login and Sign up screens easily.
- Once an email is provided, ask users to confirm it by clicking on the link attached in the confirmation email.
- Whenever we design new screens we should look for repeatable design patterns we can componentize
- If in doubt, check the original screen on the left of your practice frame and see the properties of every single element.
- Make sure your images are set to “Fit” and your constraints are set to “Left and Right”, and “Center”.
- Make sure to apply text styles from our style guide.
- Since the latest Figma update, the “Resizing” properties have been moved up near the width and height parameters. This is where all the answers to responsive problems should be.



Account setup

THE INTRO

Account setup - the theory

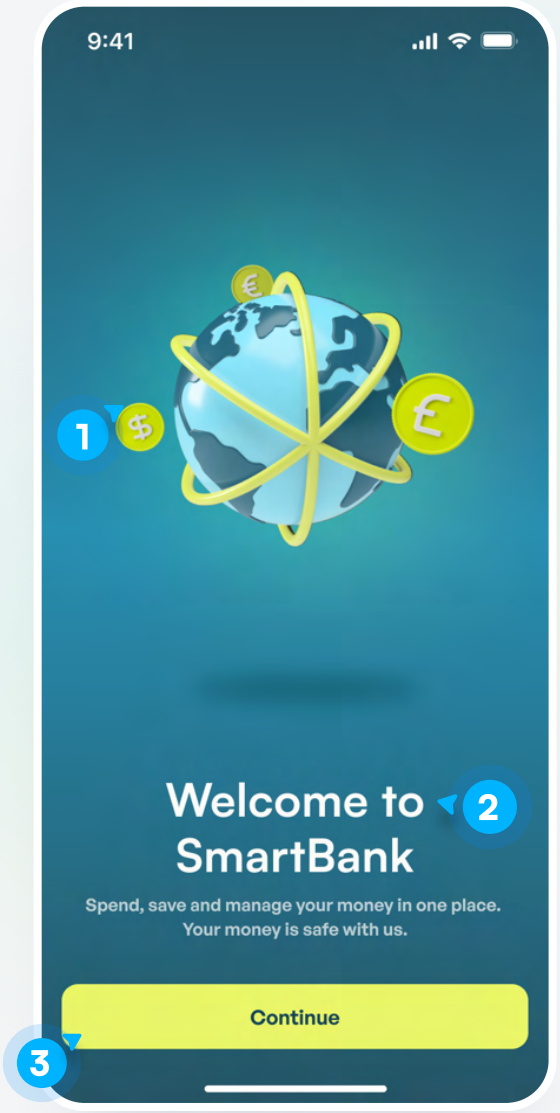
We've successfully confirmed the email and clicked on the confirmation link. Now we're taken back to our SmartBank app. Let's use a welcome screen to greet our users and add a bridge between the email flow and the account setup.

We've already mentioned our app's functionality in a walkthrough, so let's show just a simple welcome text with a nice visual.

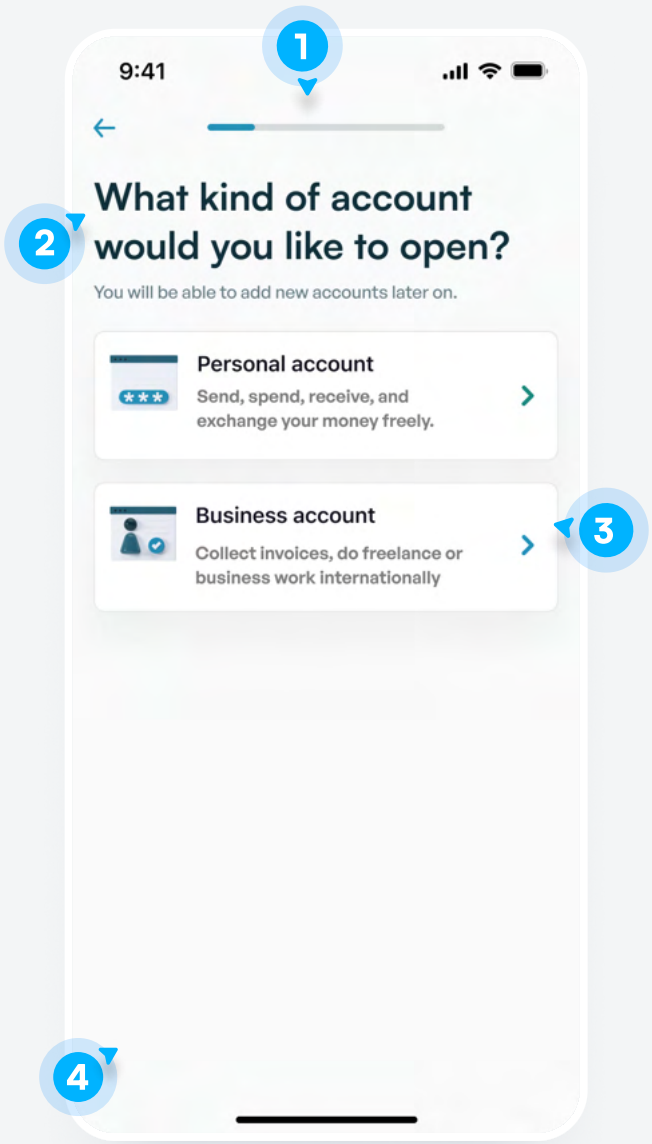
We want to imprint our brand visuals in our users' minds so that our brand is easily recognizable and leaves a good impression every time. Besides, our setup process is long and tedious, so having splash screens to break up the journey into smaller steps might be the right approach. We're also taking note of what the competition is doing.

The building blocks of a welcome screen

1. A 3d coherent illustration that goes in line with our brand guidelines
2. A big bold title with a descriptive text
3. A continue button to proceed to the next step of the account setup



1.3.1 - Welcome to SmartBank



1.3.2 - Type of an account

Choose account

The first step of setting up an account is choosing the type of account we want to create. We can choose between personal and business. For our example, we'll go with the "Personal account" flow as the Business one would require additional steps in the onboarding, and we don't want to make this project all about the account setup.

Let's add a short description to each selection to give additional context to the action to be taken.

Before we proceed, let's break down our frame.

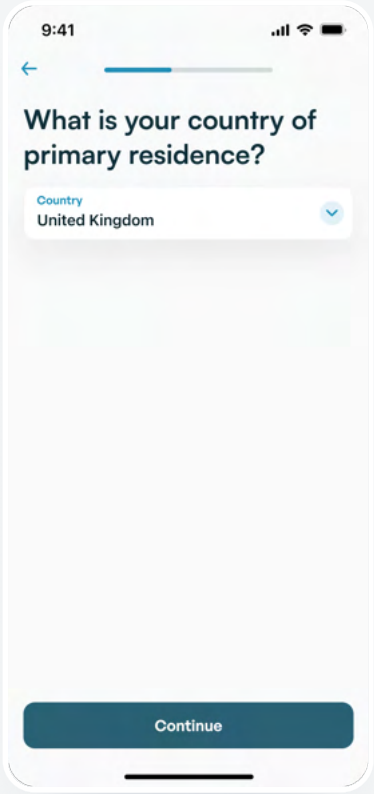
The building blocks of the choose your account screen

1. A progress bar to indicate how many steps are left
2. A title and description to help customers understand the current step in the flow
3. Cards with chevrons to indicate the action buttons and the start of a new flow
4. Absence of a CTA button - we want the user to focus on the cards

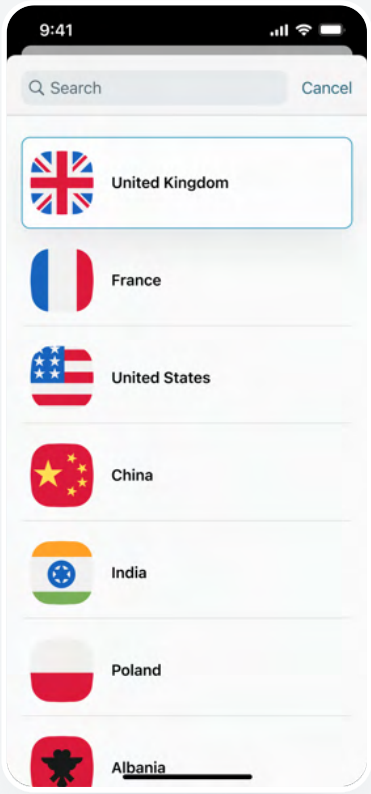
Country of primary residence

After choosing an account type, we must fill in our personal details. If it's a personal account - our personal information, if business account - details regarding our company.

In the first step, our system will require to ask about the primary country of residence.



1.3.3 - Select Country

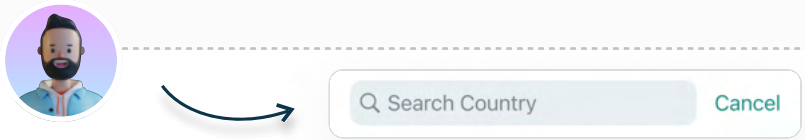


1.3.4a - Select Country 1-2

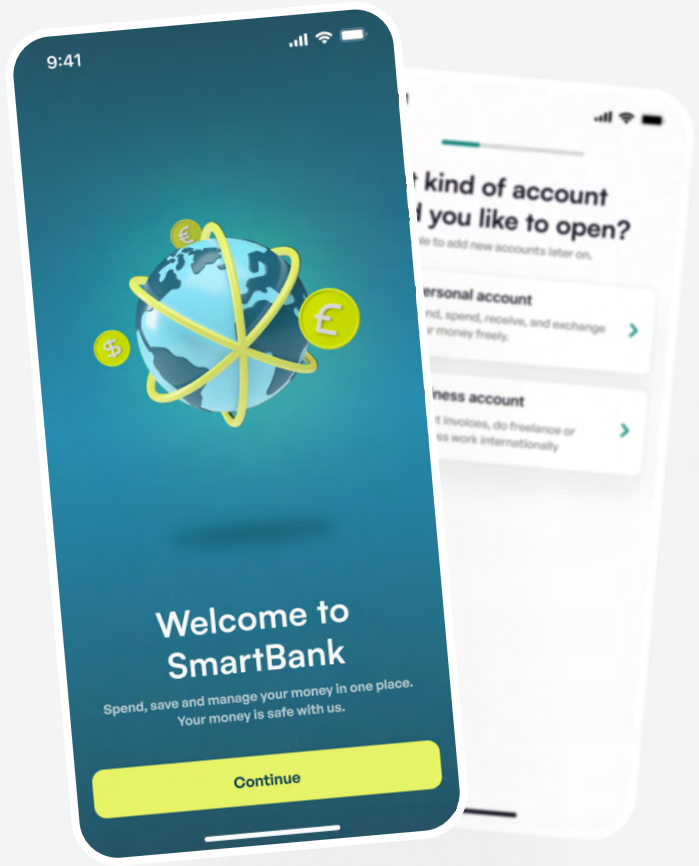
Since every country has different financial regulations, we need to get this information right. We have to verify this later on by taking specific steps and precautions, but for now, let's continue.

The app should suggest our country automatically by looking at our IP address and our default phone settings. This is a good UX practice to recommend already known information to make the setup more straightforward and faster to complete.

We should also consider adding a manual select field to change our choice. Whenever we're traveling, staying abroad for longer, or even using a VPN, our residence and financial regulators might still be in a different country.



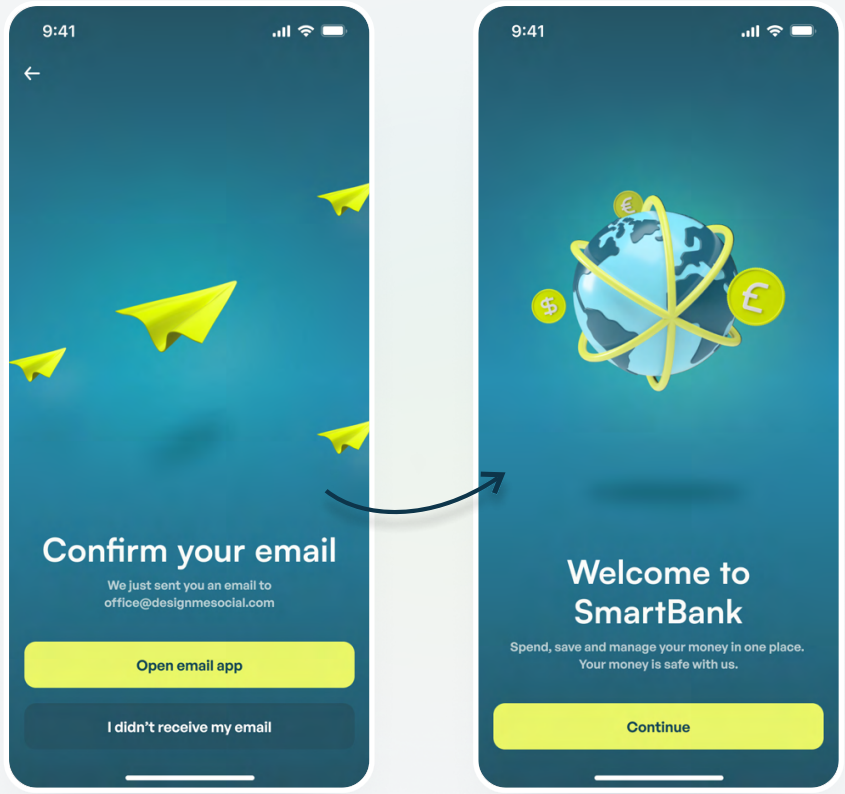
We should include a search input for lists longer than 6-8 items. Having an option to type in the keyword in the search bar is always a better experience than scrolling down for seconds on end to find an answer at the bottom of the list.



Account Setup

THE STRUCTURE

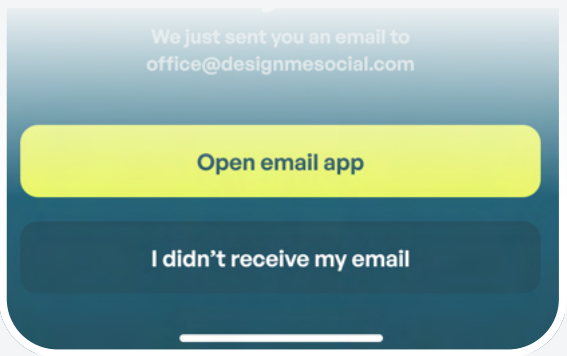
Account setup - the structure



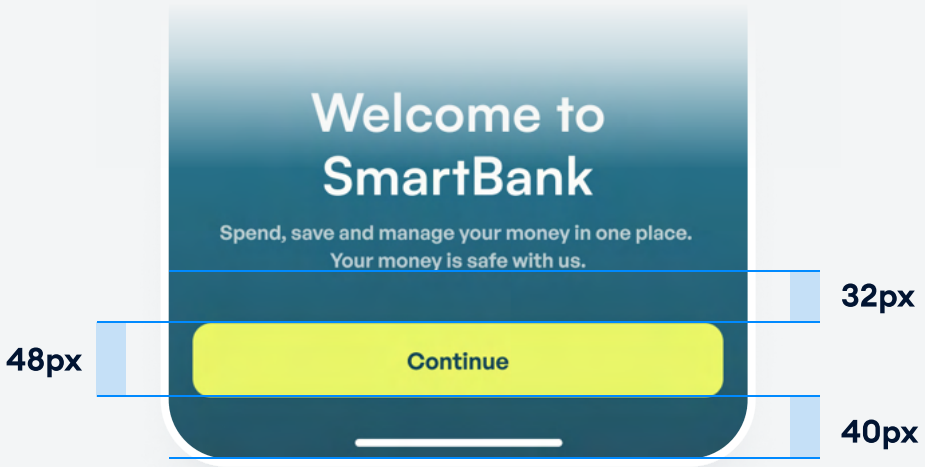
1.2.3 - Confirm your email

1.3.1 - Welcome to SmartBank

Looking at our welcome screen, we can already see a lot of similarities with the “Confirm your email” screen. Replace the text and delete the second button. Position our “Continue” button 40 px from the bottom of the frame.



I have grouped the buttons into an auto layout selection with bottom alignment. If you delete the bottom button, your content will shift downwards. This will leave an additional gap between your text layer and your button. Move the text down until it’s 40 pixels away from our CTA.



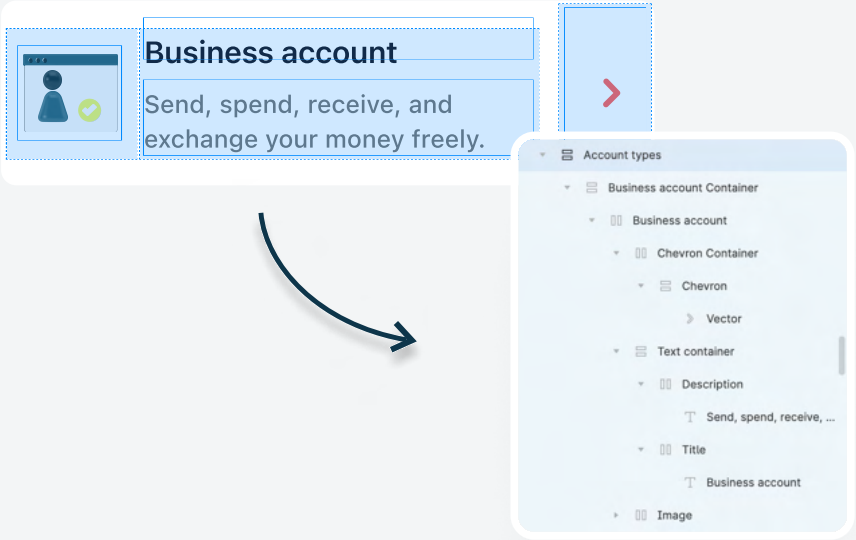
Choose account

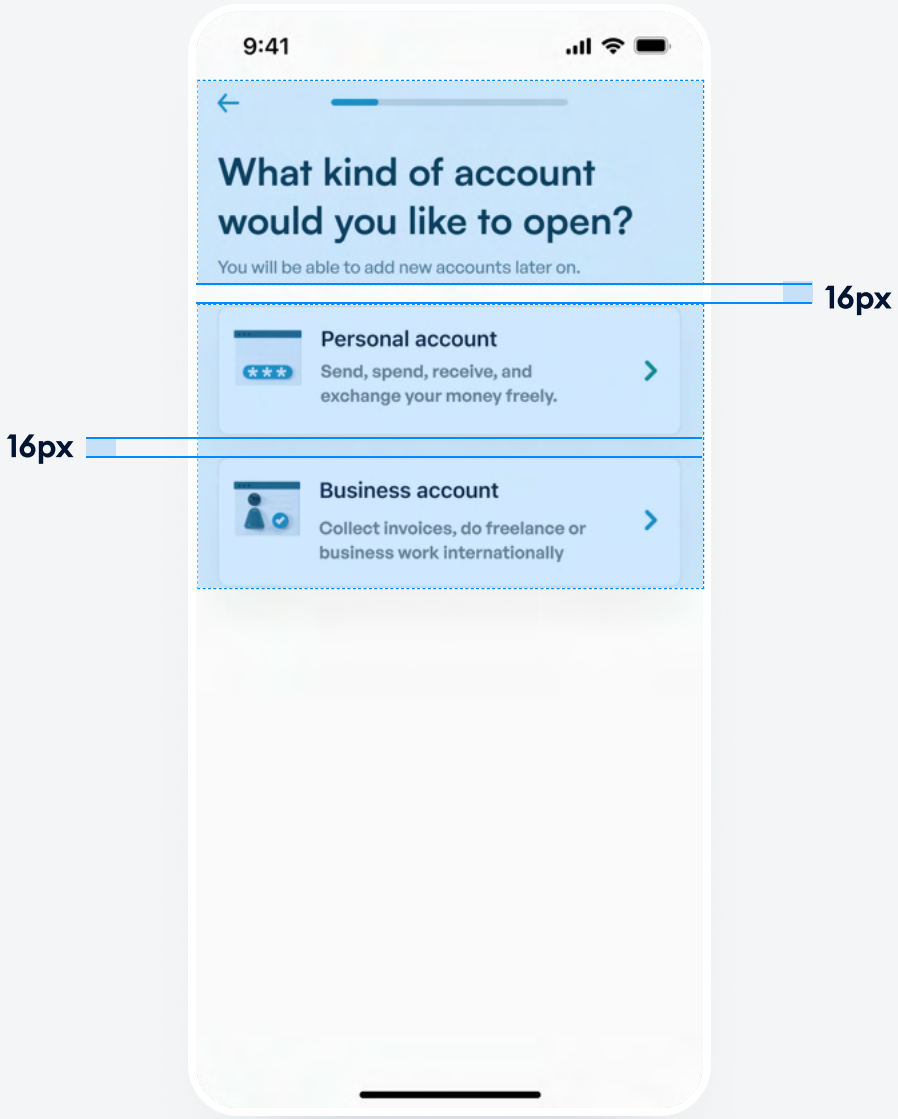
1. Text Container

Copy it directly from the “What’s your email” frame for our text container and replace the text. Double-check to see if you’ve copied the one with the back button. See if the constraints are set. If not sure, look at the left reference frame in the practice file and check the settings there.

2. Cards Container

This one’s a bit more complex, so make sure you analyze every single layer selected by comparing the auto layout settings by looking at the reference frame. Try to reverse engineer what I did and start with two text layers first.





1.3.2 - Type of an account

Apply auto layout to your Title and set the text layer to fill container. Do the same with the Description.

How to really understand auto layout

Learning auto layout is a lot like trial and error; practice it by re-adjusting your layers and ensuring the proper resizing properties are set.

Copy the card from our completed frame for this exercise and paste it onto your practice frame. If you're feeling confident, try replicating the container yourself by ungrouping the elements and grouping them again. Your resizing properties will reset, but you'll be able to trace back my steps by looking at the finished card container you have on your original frame.

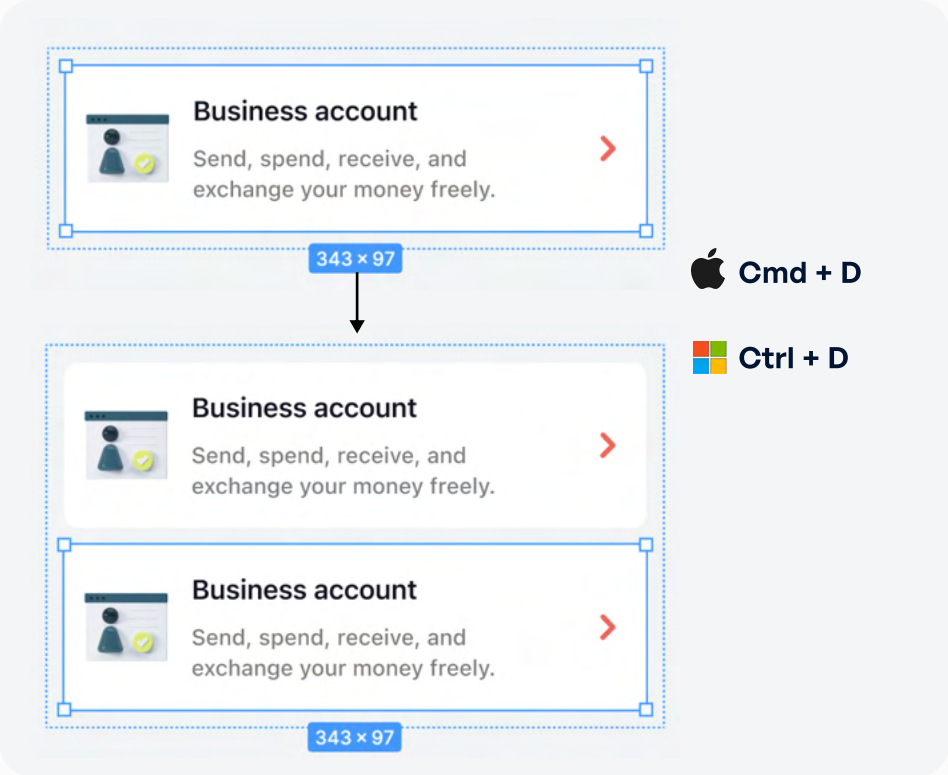
When your selection is not resizing correctly, one of the layers or groups has incorrect properties set. In this case, you need to make sure everything is set to fill the container with your image being set to a fixed width.

Open the "Auto Layout Practice file" in our e-book folders provided. It's an official practice file created by Figma for you to practice auto-layout and resizing frames.

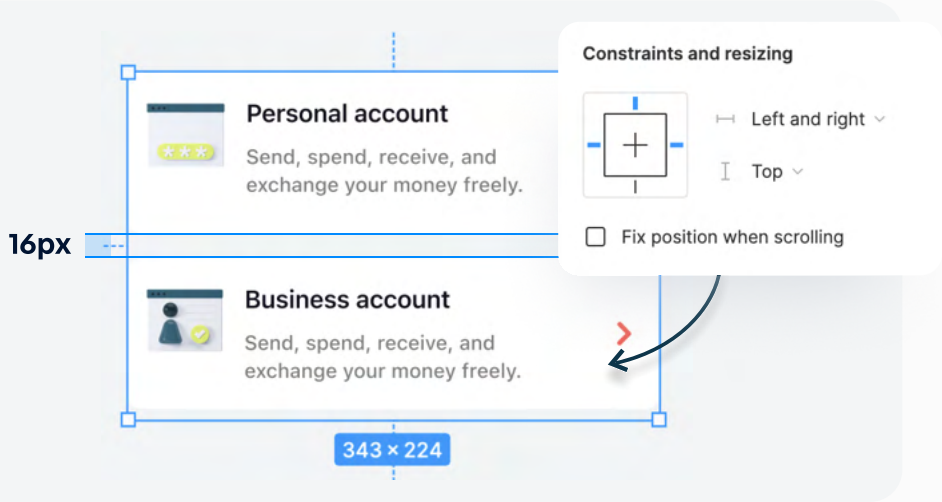
Anyway, let's continue with our flow.

When you have one card created, make a component out of it, you never know if you won't need it again. Move it out of the frame, and having it selected, click "Option + Cmd + K." Make a copy and put it back on our frame.

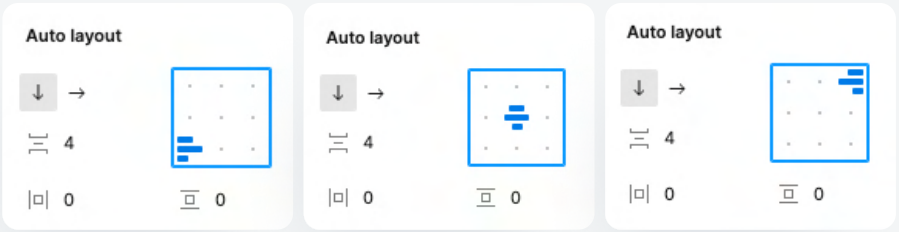
Apply auto-layout to create another container, and set the padding back to 0. We'll need it to put our cards inside. Now select one card and duplicate it. It will automatically align itself within your new container. This way, you can quickly copy components and automatically align them.

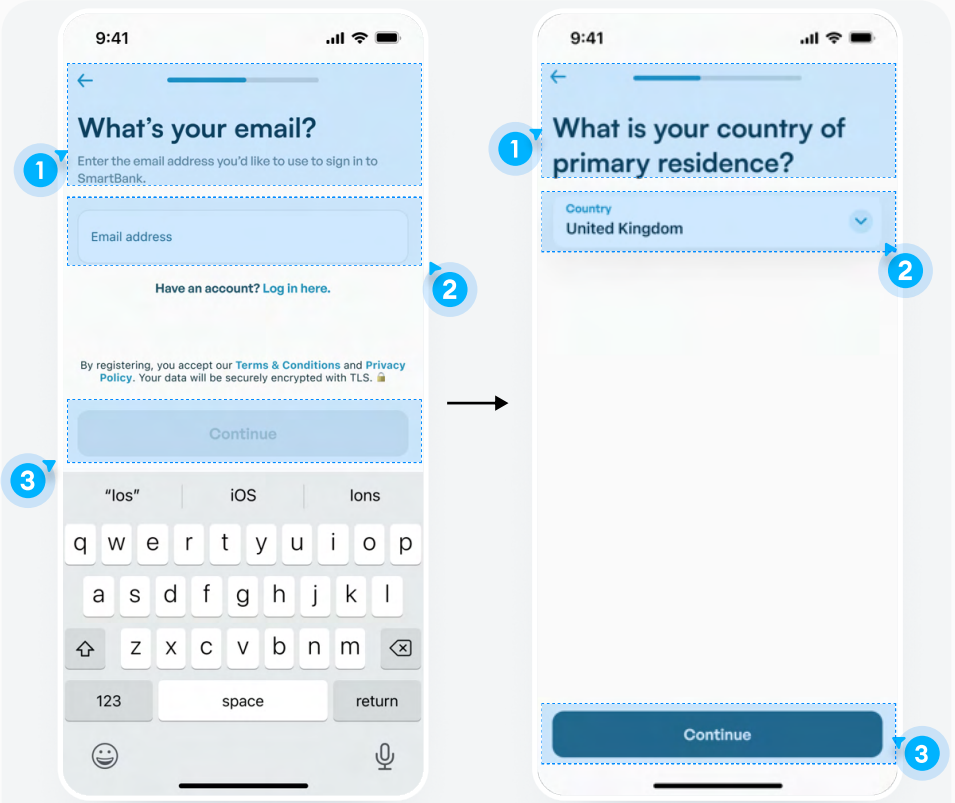


Update the text, replace the images, and apply constraints to our selection. Set them to “Left and Right” and “Top.” Set the space between your auto-layout groups to 16px.



When you want to align your auto layout selection group to the bottom, top, left, or right, look at the Auto layout property on your properties panel on the right. Remember about this moving forward.





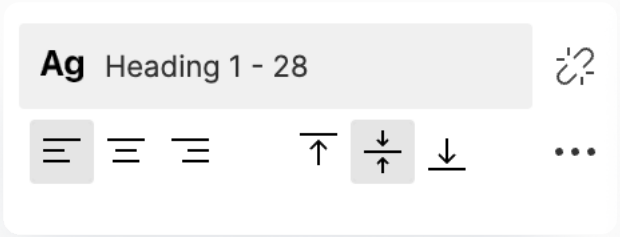
Do you see any resemblance to one of our previous screens? If you thought about the Sign-up flow, well, you guessed it. Remember about always working smart, not hard, so let’s copy our Sign up frame and start readjusting our design.

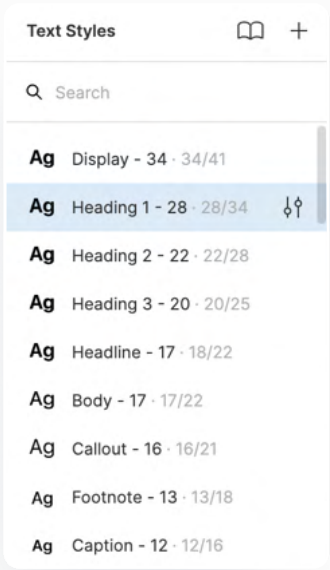
First off, let’s change our text and remove the iOS keyboard. Move our button to the bottom, position it 40px from the bottom of the frame, and change its style to Primary.

Figma styles

If you can't find the button styles or styles I'm referring to, you should always look at the properties panel on the right.

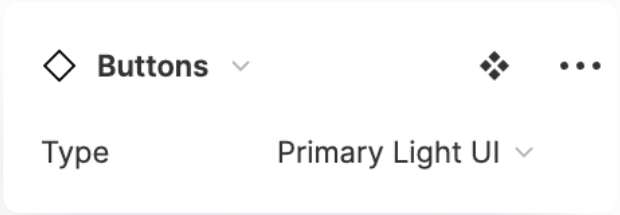
Text

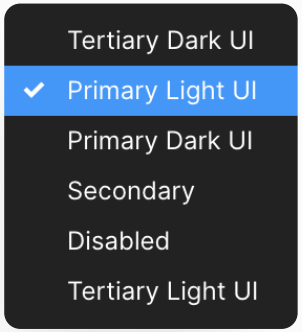




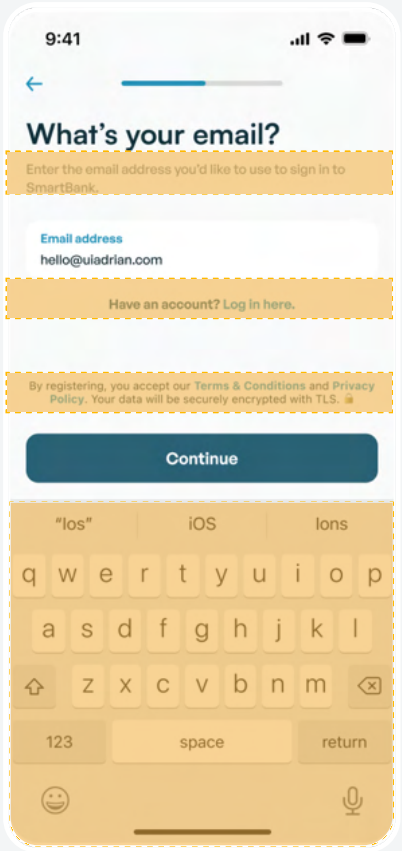
You can look up your text styles whenever you click on a text layer and click on the label with your style name.

Button





You can find the same property for your buttons. Just click on the type, and it will open a list of your available styles.



1.2.2a - Email 1-2

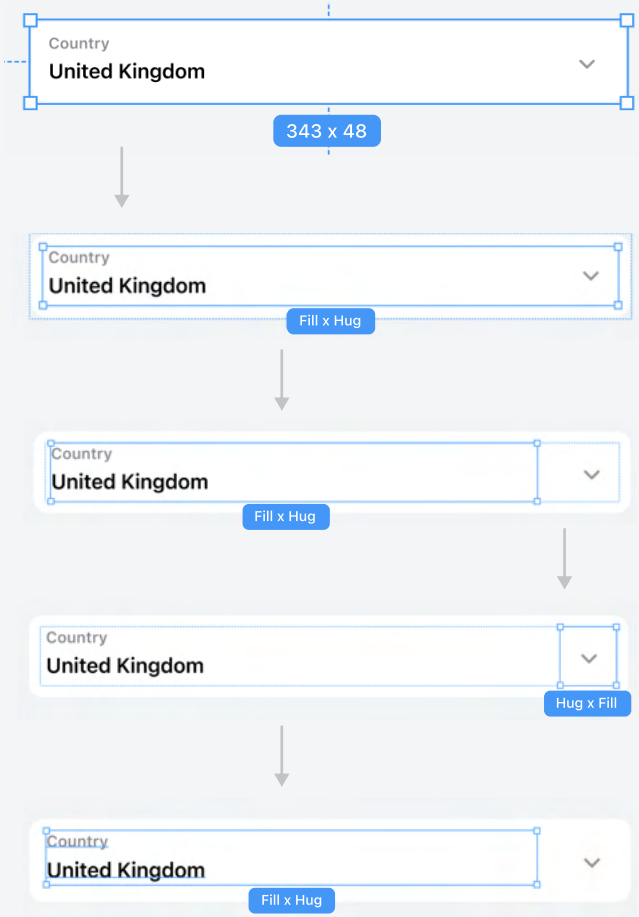
Start removing the elements highlighted in orange.

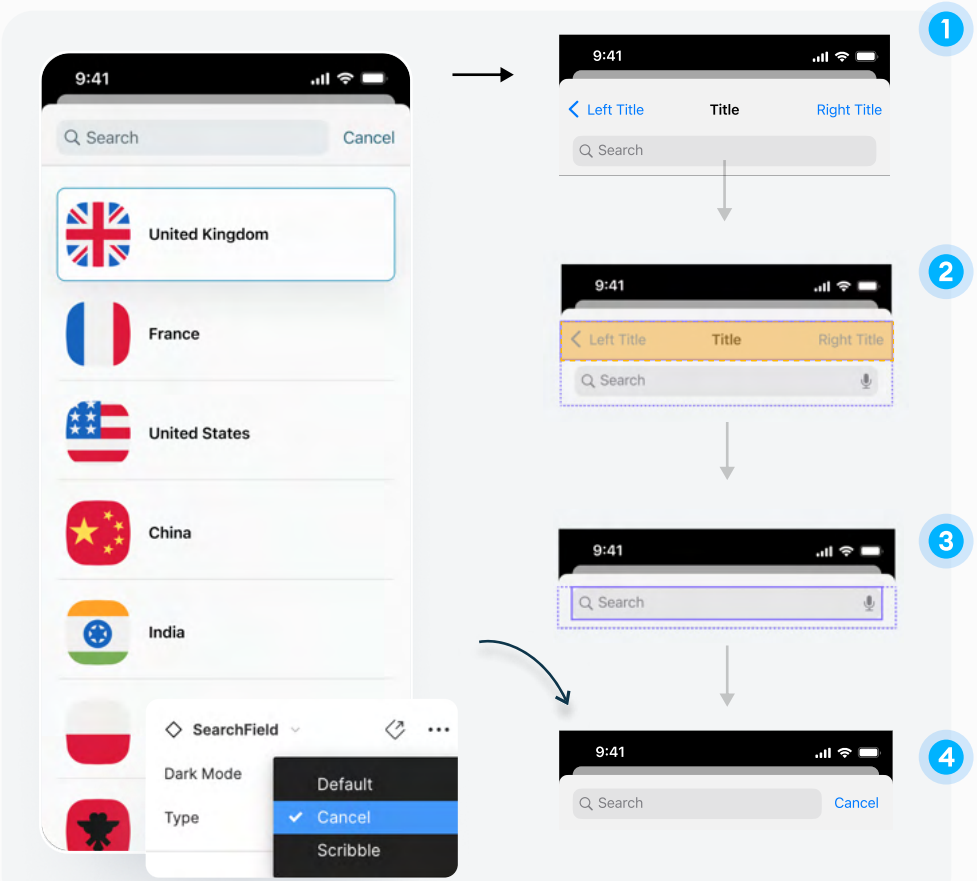
Reduce the size of our heading to 28 px, so it's more compact and legible; we don't need to make it Display size.

We'll also need to adjust our input to include the chevron.

Country Input Field

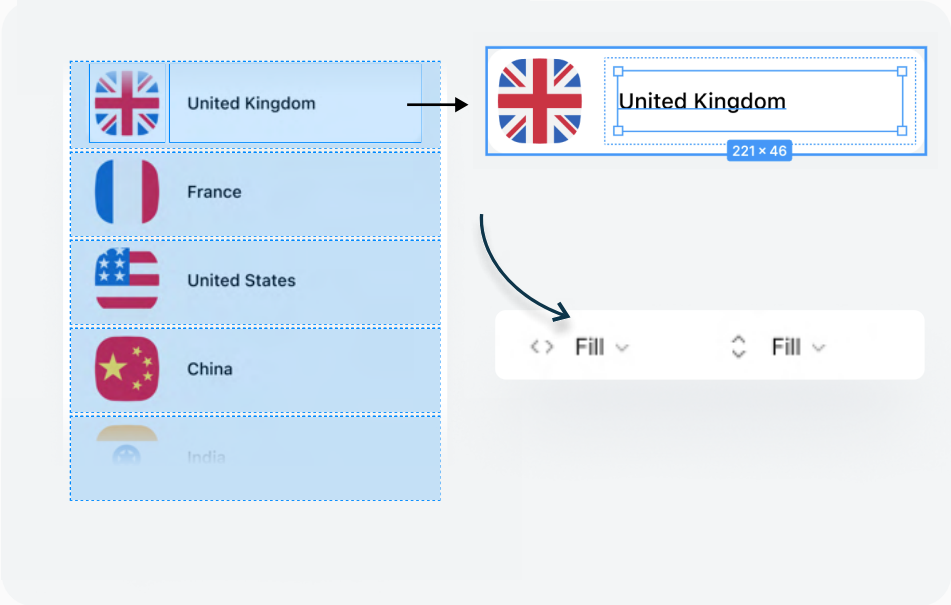
This one's a bit more complex, so make sure you analyze every single layer selected by comparing the auto layout settings by looking at the reference frame. Try to reverse engineer what I did and start with two text layers first. Here's how I did it.





We'll use a default iOS top navigation from our iOS UI Kit for the next screen. Let's adjust the style.

1. Copy and paste it into our frame
2. Remove the toolbar; we don't need it here
3. Delete the voice search icon and select our search input
4. Look at the properties panel on the right and change "Type" from "Default" to "Cancel." Remove the voice tool icon.



We'll need to add a text layer and apply auto layout for this card. Remove the padding and left-align the selection. Add the flag image and position it 16 px to the left of the text layer.

Select both layers and apply auto layout. Align everything to the left and middle. Drag the layer to make it 343 px wide and 80px tall. Make sure every auto-layout container and the layers inside are set to fill container.

You don't have to recreate everything from scratch; look at the reference frames and check every property. These are rather complex designs, and you'll need to practice quite a lot before you get them right. Take your time and do it right.

Chapter recap

- Use a welcome screen to greet users and add a bridge between the email flow and the account setup.
- Add a short description to each selection to give additional context to the action that's about to be taken.
- Include a search input for lists longer than 6-8 items.
- When you want to align your auto layout selection group to bottom, top, left, or right, look at the alignment and padding property on your properties panel on the right.
- When your selection is not resizing correctly, one of the layers or groups has incorrect properties set.
- Make sure your images are set to "Fit" and your constraints are set to "Left and Right" and "Center."
- Make sure to apply text styles from our style guide.
- The setup process can be long and tedious, and having splash screens to break up the journey into smaller steps might be the right approach for you.



Phone verification

THE INTRO

Phone Verification - the theory

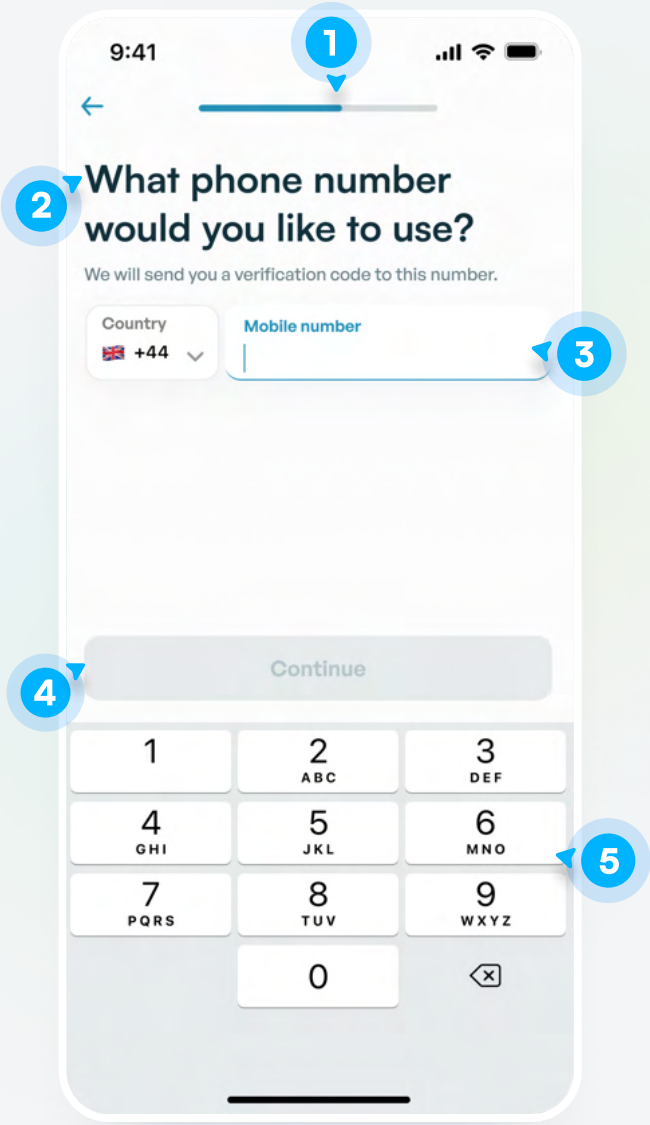
After choosing our country of residence, we need to provide our system with a phone number. We'll be using our phones to verify all actions taken within the app.

The prefix will appear automatically depending on the user's country selected in the previous steps. A prefix is here to check if the IP geolocation matches the phone number's country code - we use this to avoid fraudulent activities.

We want to ensure all phone numbers provided by the users are correct and the sign-ups are unique; therefore, an authentication passcode is essential to confirm future deliverability.

The building blocks of a phone verification screen

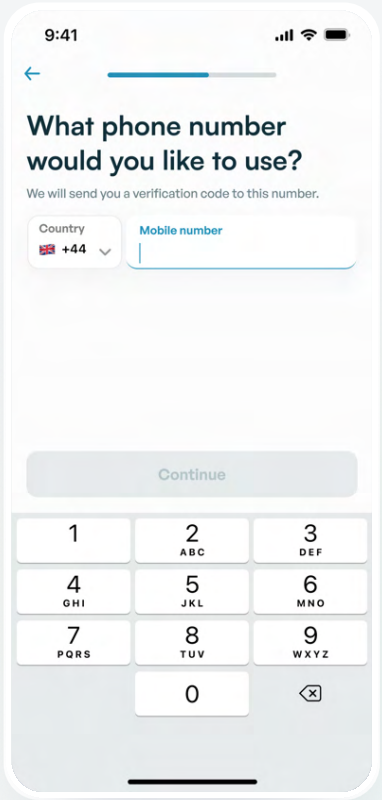
1. A progress bar to show the steps left
2. A title with a description to explain the active step
3. A number input with country codes (select input)
4. Disabled state CTA button to indicate inactive state
5. iOS numeric keyboard to type the number in



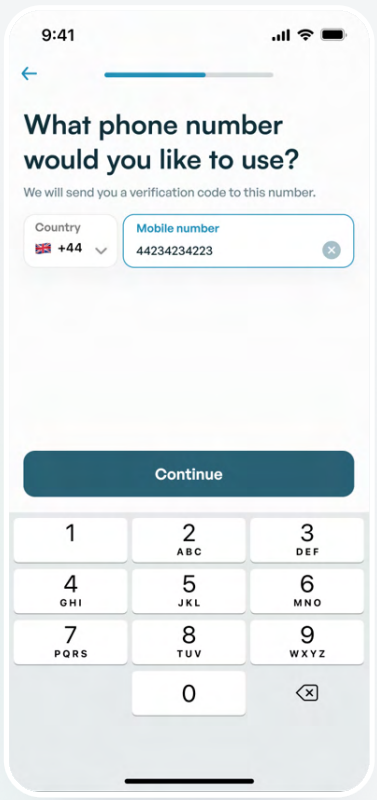
1.3.5a - Phone Number 1-2

Verify your phone number

A green outline will indicate a success state when the phone is added, and the button will become active. After clicking the continue CTA, a user will be taken to the next stage of the process, a One-Time Phone Verification (OTP). A verification code will be sent to the number provided in the previous step.

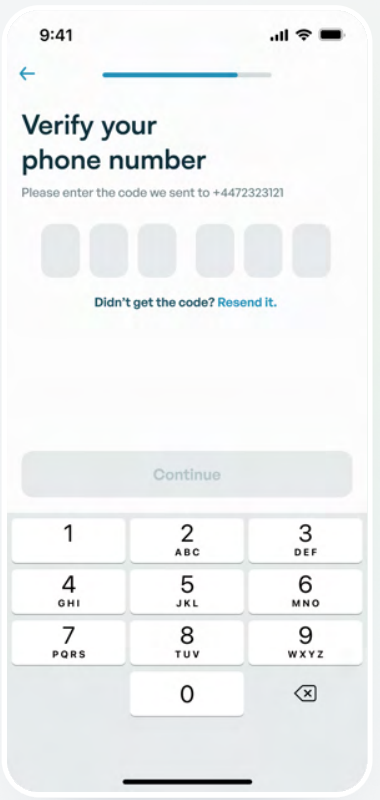


1.3.5a - Phone Number 1-2

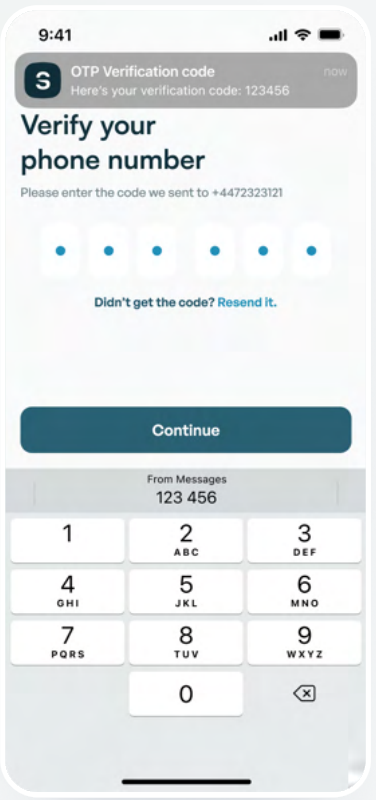


1.3.5b - Phone Number 2-2

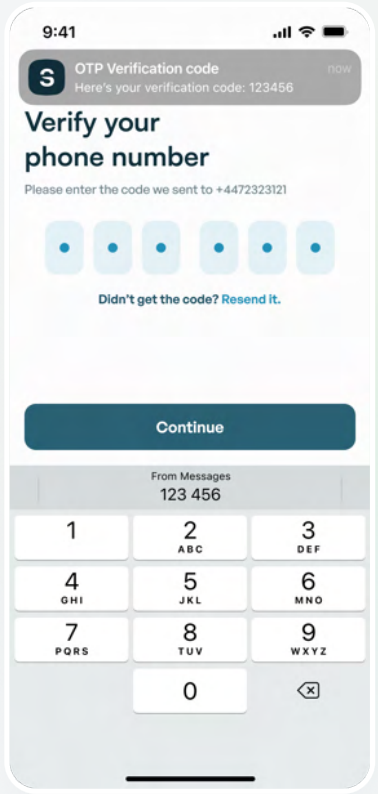
Using iOS push notifications, we should be able to suggest the received code on our numeric keypad automatically by using a built-in iOS functionality. If we didn't receive a verification code, there should be an option to resend it, although we should limit the code resend to 1 every 30 seconds to avoid spamming our system. Humans are impatient by nature, so having a built-in retry logic is a must.



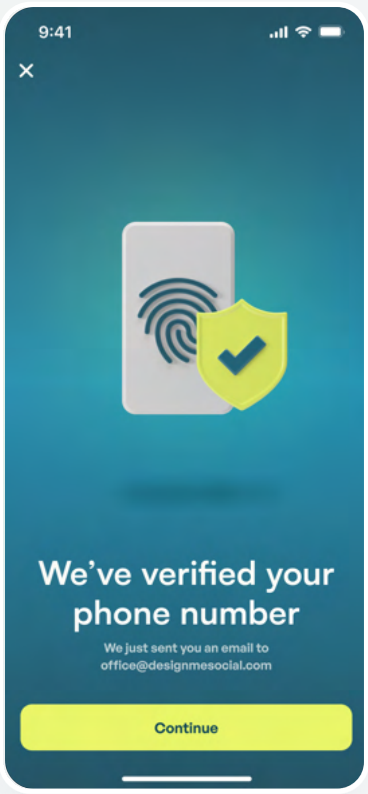
1.3.6a - OTP 1-3



1.3.6b - OTP 2-3



1.3.6c - OTP 3-3



1.3.7 - Phone verified

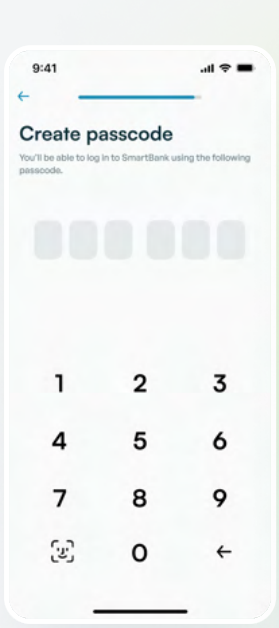
A one-time password (OTP) is a number or alphanumeric string of characters created automatically and used to authenticate a user for a single transaction or login session. It's the most common way to verify a phone number.



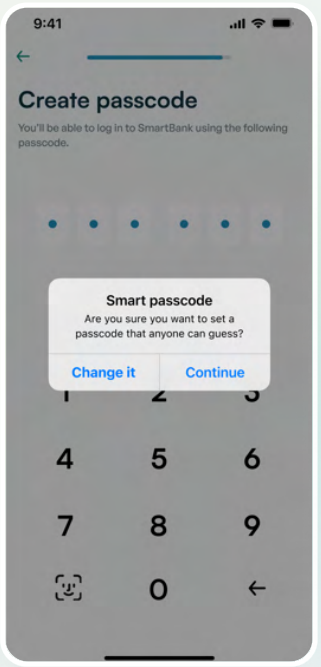
Create a passcode

Our next priority is to set up a passcode. We will be using it to confirm our transactions, request sensitive data, and log in to our app in the future.

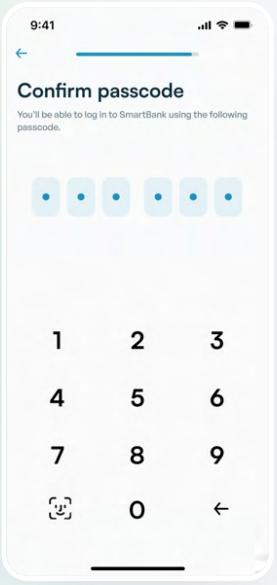
This flow is relatively simple. Create a passcode and confirm it. If the passcode is too easy, for example - it follows a string of numbers like 1111 - shows a notification suggesting to change it or asking if users want to proceed with it.



1.3.8a - Passcode 1-3



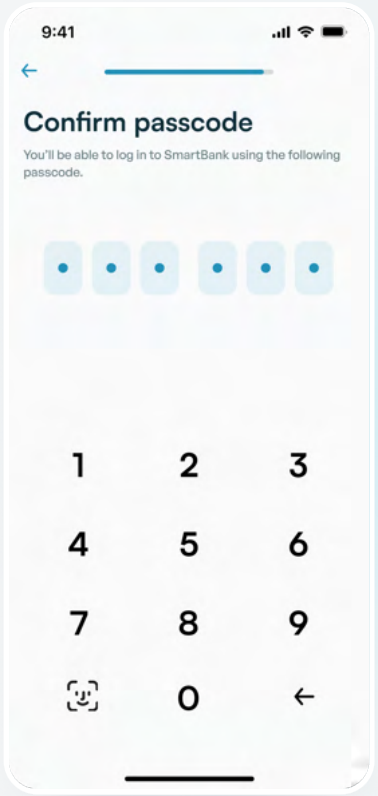
1.3.8b - Passcode 2-3



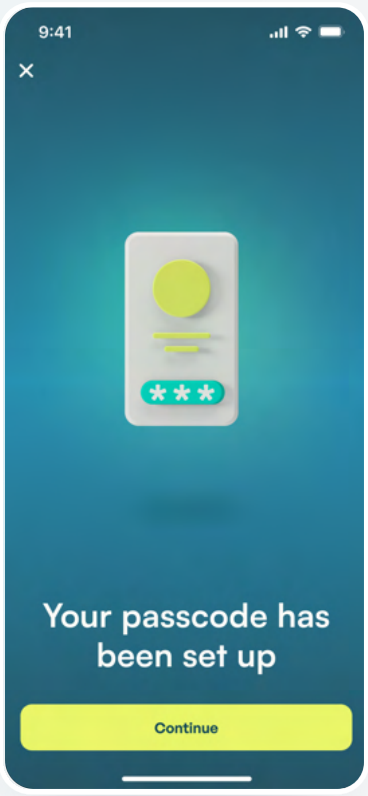
1.3.8c - Passcode 3-3

We'll, of course, need to type our passcode again without making any errors to activate it.

Remember, don't leave your users without success states. Either show an in-app notification, a success toast, or an entire success splash. In our example, we went for the latter. A nice, colorful, juicy, and bold success screen. No need to guess if the step's been completed here.



1.3.8c - Passcode 3-3



1.3.9 - Passcode set up

Why passcode and not Face ID?

Here's the thing, Touch ID and Face ID are pretty secure ways of protecting your phone, but they both have vulnerabilities, and neither method is perfect.

Touch ID has been shown numerous times to be vulnerable to high-tech fingerprint copies. There are many examples of other devices that have been unlocked using a photograph of the user's face put real close to the camera. Those aren't likely to be of much concern if you leave your phone at the gym or stolen from your backpack since it's not really expected to be worth the effort to unlock.

On the other hand, if someone has possession of your device and can place your finger on the sensor or point it at your face, they can potentially decrypt your device. The fact that someone could use your biometric data to unlock your device is why you have to enter your passcode every time you restart your iPhone or MacBook. In fact, if you're really interested in protecting your data, a passcode is actually far more secure, provided you can keep it a secret.

For example, a four-digit passcode takes about seven minutes for a computer to guess. On the other hand, a six-digit passcode has one million possible combinations. It's far more secure!

We'll start with passcodes and see if our users prefer to have biometrics enabled - we'll test it, iterate it, and improve it if needed.

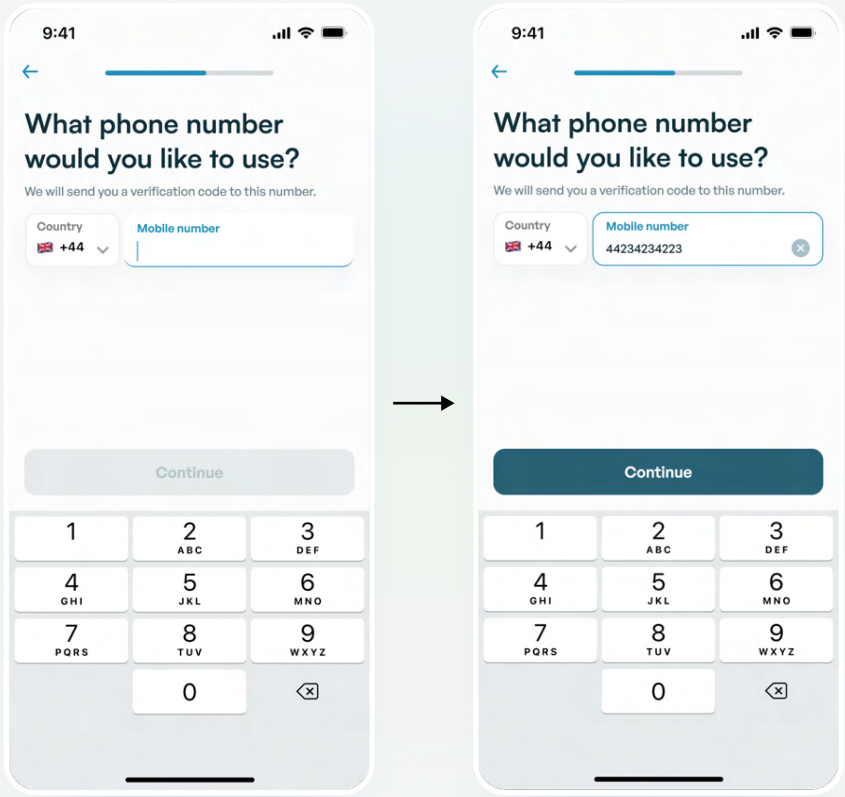


Phone Verification

THE STRUCTURE

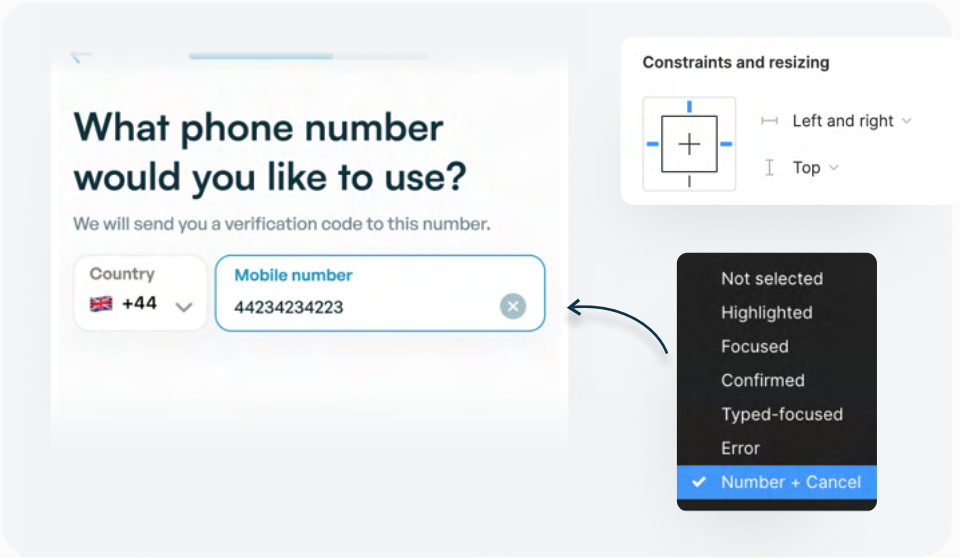
Phone Verification - the structure

We'll need two screens with two states for this short flow - one unfinished with a disabled CTA and a second one with a successful input and button state. For our top title part, let's copy-paste our Title component and replace the text.



1.3.5a - Phone Number 1-2

1.3.5b - Phone Number 2-2

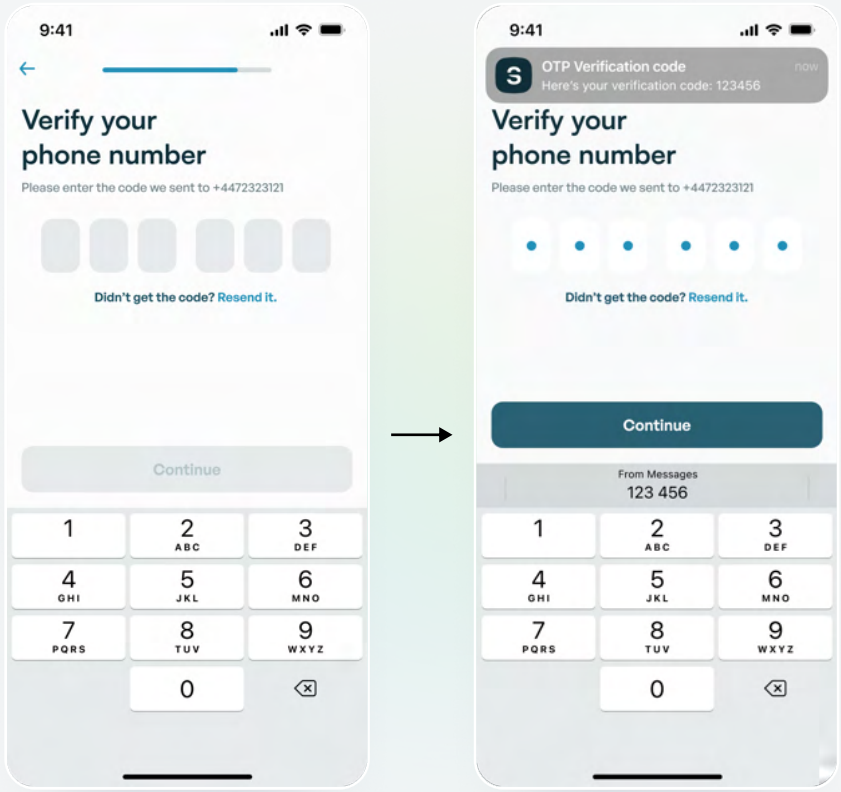


Phone number input field

Copy the input field from the **"Country of Residence"** frame. Reduce the width to 240 px. Now duplicate the input field you just created, reduce the width to 98 px and place it 4 px to the left of your mobile number input. Select both layers and apply auto layout. Add **Constraints** "**Left and Right**" and "**Top**."

This will need quite a lot of manual adjusting to make everything work. It's best to copy the input field from the reference Figma file and check every setting. Practice it yourself for a while and see if you can make it work.

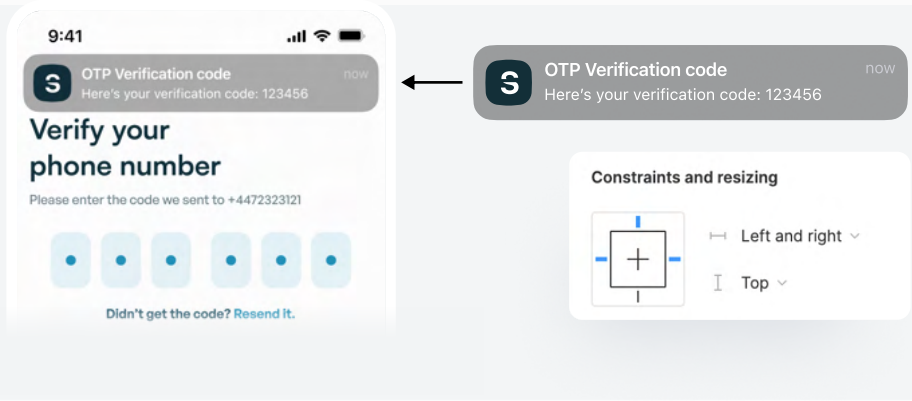
For the completed screen, let's change our **button style** to "**Primary**" and our **input field type** to "**Number + Cancel**."



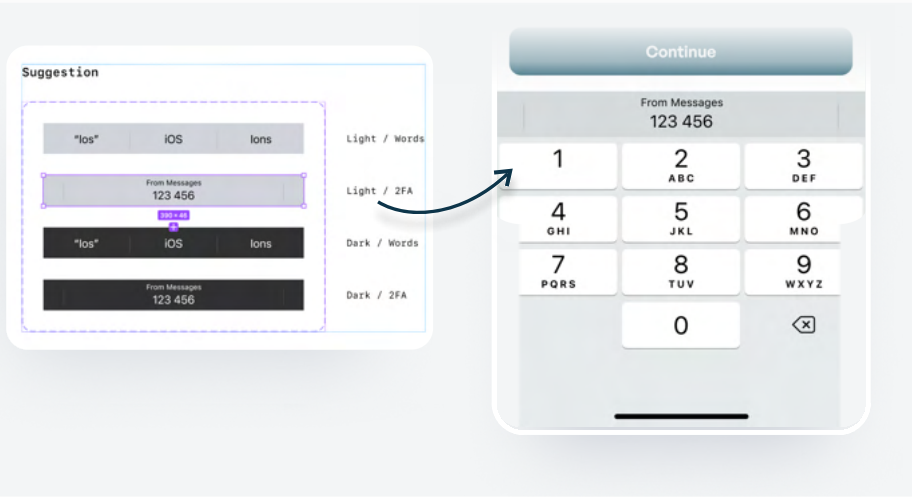
1.3.6a - OTP 1-3

1.3.6b - OTP 2-3

Copy the top section from a previous screen and replace the text. Create a rectangle (40 x 56 px size) and duplicate it 2 times. Group the selection and duplicate it again. Place them 20 px apart and apply auto layout. Set “Left and Right” and “Top” constraints. Copy the iOS Keyboard from our iOS 15 UI Kit and align it to the bottom and middle. Apply constraints “Left and Right” and “Bottom.”



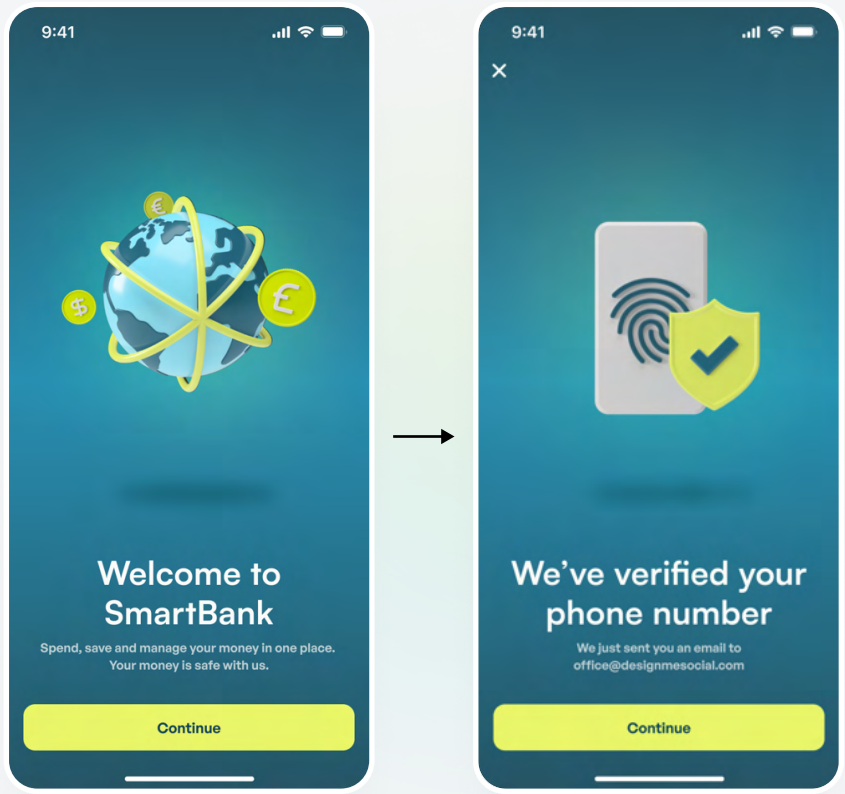
Let’s copy a push notification from the iOS 15 UI Kit, replace the text, and align our notification container 44 px from the top. Apply “Left and Right” and “Top” constraints.



Copy the “Suggestion” strip from our iOS 15 UI Kit (**reminder** - you will find the link in the **resources folder**) and place it on top of the iOS Keyboard. Resize it to fit the frame, apply constraints, and change the button state to Primary. Check it against the original frame.

We've verified your phone number splash

This one's super easy. You can copy our "Welcome to SmartBank" entire content and replace the image, title, and description text.



1.3.1 - Welcome to SmartBank

1.3.9 - Passcode set up

Tip: Always look for repeatable design patterns you can easily duplicate to speed up your work.

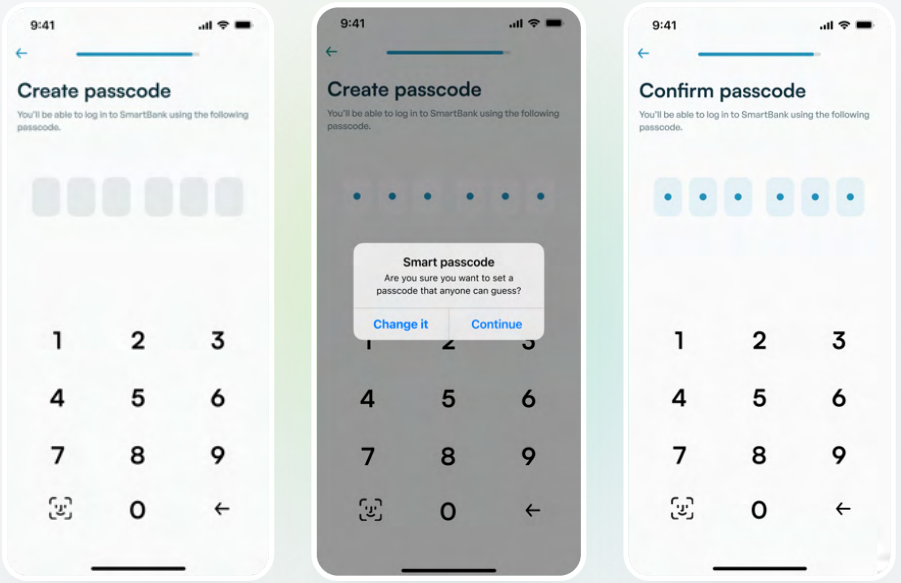
Passcode screens

Copy the top section from one of the screens before.

Create four rectangles, select them and apply auto layout. Make sure your rectangles within the auto-layout selection are set to fill container. Apply constraints to the left and right, and top.

Create our keypad, align it, group it and apply auto-layout. Center our selection and add our constraints - "Left and Right" and "Bottom." **Look at the reference frame if lost.**

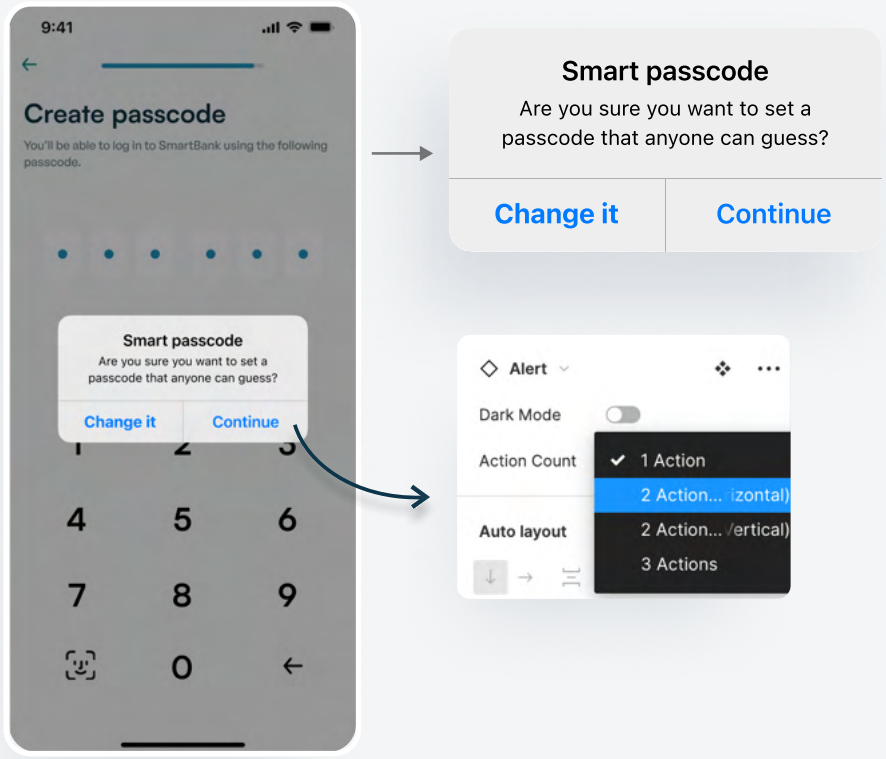
We'll need an alert popover to show weak passcode settings.



1.3.8a - Passcode 1-3

1.3.8b - Passcode 2-3

1.3.8c - Passcode 3-3

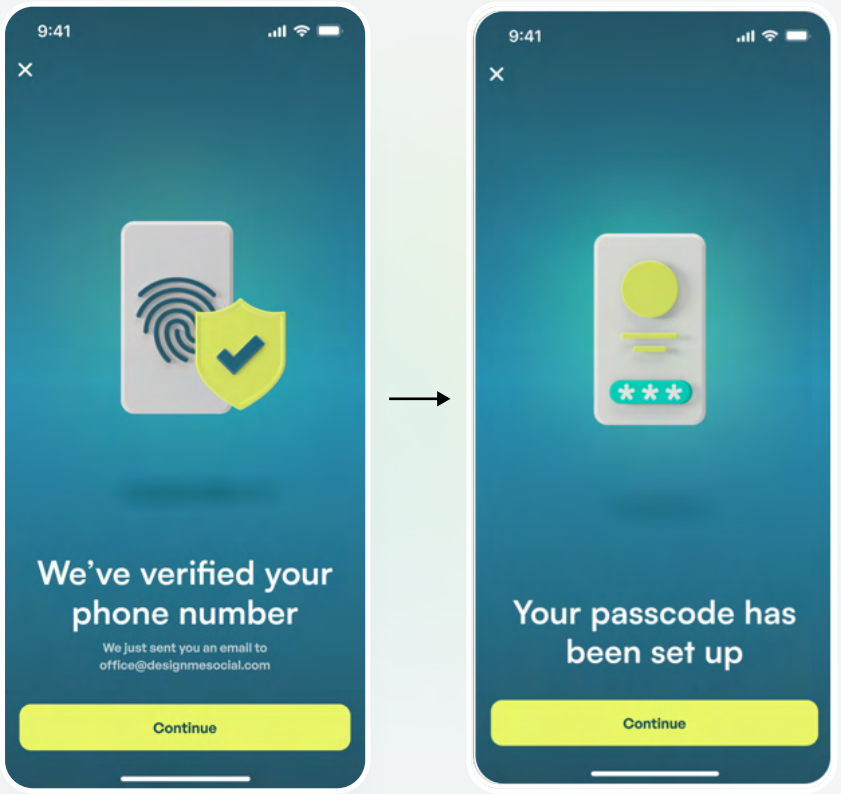


1.3.8b - Passcode 2-3

We're once again using the one provided in the iOS 15 UI Kit for the alert notification. Copy the alert message with the overlay background, resize it and change the text of our alert template.

To do that, select the Alert message and navigate to the properties panel on your right to add two action buttons. Click on the "Action Count" and choose 2 Actions. Replace the text copy.

Passcode has been setup



For the "Passcode setup complete" screen, we can simply copy the content from the "We've verified your phone" frame we just created. Replace the image, add a new title, and update the description text.

Feeling lost?

I feel like I have to mention this again. If you ever feel lost in the design process, look at the frames provided in the Figma practice file. The easiest and the best way to practice design is by replicating the work of other designers.

How you want to approach this practice file is up to you. If you wish to place the original frame on your practice frame, reduce the opacity, lock the layer and trace the design, or if you want to replicate the work by manually creating containers and elements that you see on your left. Either way would work.

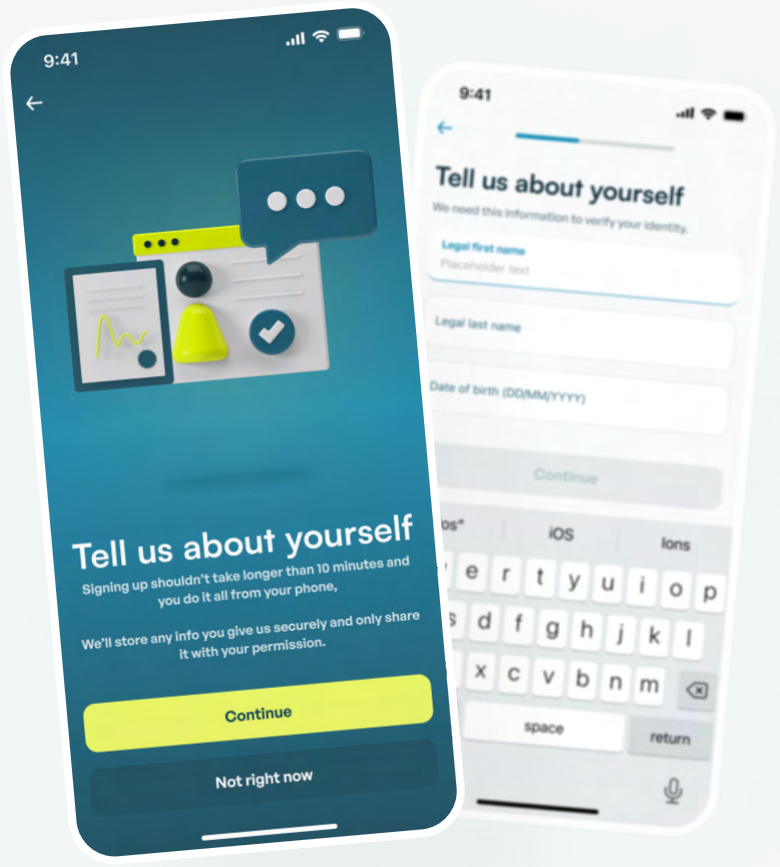
If you're a beginner, I'd suggest working on improving your eye for design, not necessarily trying to understand all there is to know about the auto-layout and containers, just really getting to understand the basic principles.

Many designers are not actually using auto layout in their designs, and they're still great at their job. Focus on recognizing design patterns and just design, design a lot. You'll get better in no time, trust me.



Chapter recap

- Welcome screens are screens that users see when they login to the app.
- Adding a short description to the provided choices is also a great practice.
- Always make sure constraints are set. Look at the left reference frames and check the settings there.
- When your selection is not resizing correctly, one of the layers or groups has incorrect properties set.
- Make sure every auto layout container and the layers inside are set to fill container.
- For the alert messages, use the ones provided in the iOS 15 UI Kit under "Alert Template."
- For every iOS native view, remember to use the ones provided in the iOS 15 UI Kit.
- If in doubt, check the original screen on the left of your practice frame and see the properties of every single element.



Legal information

THE INTRO

Personal details - the theory

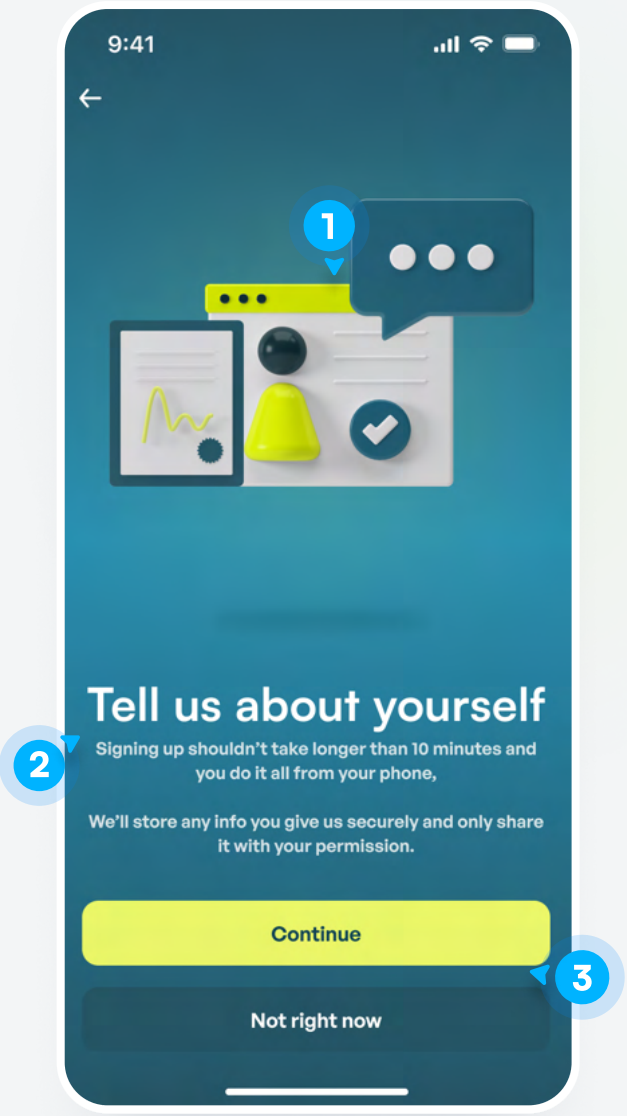
Onboarding a finance app user is a tedious and lengthy process; therefore, it's essential to get it right. We should inform our users about the time engagement needed to complete the next set of actions to set some expectations.

Compared to a traditional in-person bank onboarding, it's still a much faster experience, especially when you can do it all from your phone. Still, our attention span has decreased throughout the years of using mobile devices, and it'd be good to include an option to skip the verification process for now and let them continue in the future. Always leave one door open, so your users don't feel locked in.

However, we should inform them that they won't be able to send or receive payments until the verification is complete.

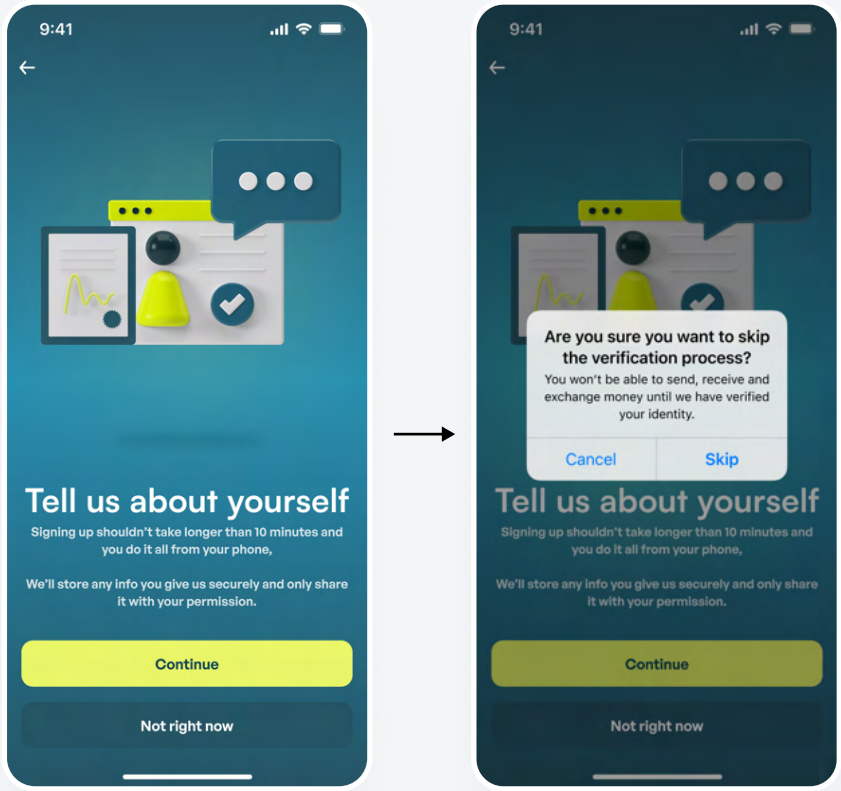
The building blocks of a phone verification splash

1. A 3d illustration that matches our copy
2. Title with description to set the expectations before starting the verification process
3. Primary CTA to continue, secondary to skip the process



1.4.1a - Get started 1-2

Skip the verification reminder



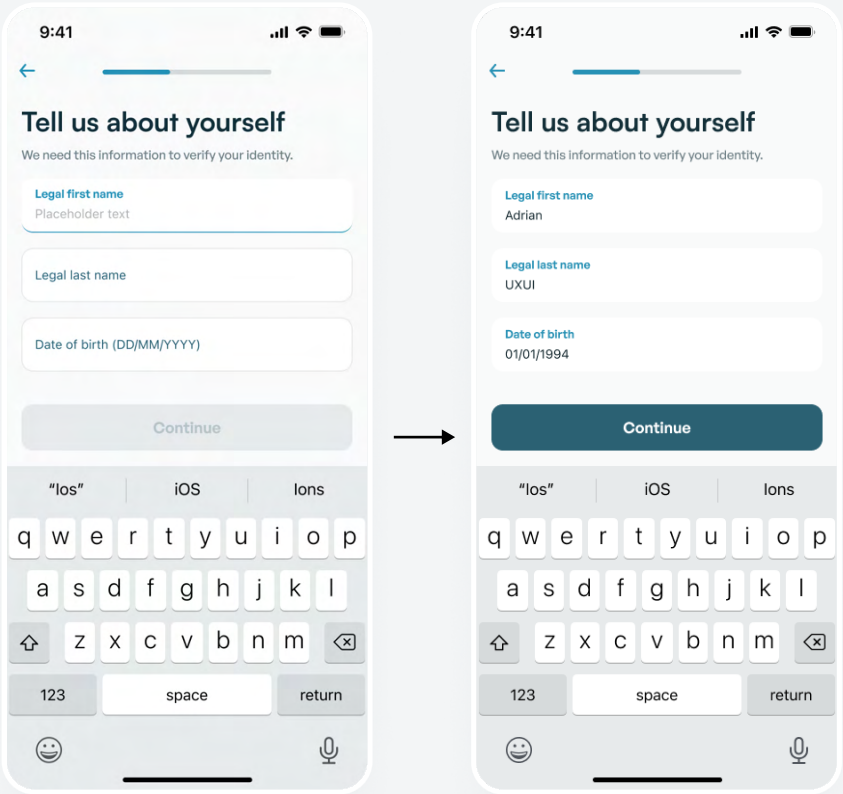
1.4.1a - Get started 1-2

1.4.1b - Get started 2-2

iOS Alerts - when to use - Alerts disrupt the user experience, so it's important to use them infrequently and only to deliver critical information or confirm irreversible actions.



Personal details



1.4.2a - Legal information 1-4

1.4.2b -Legal information 2-4

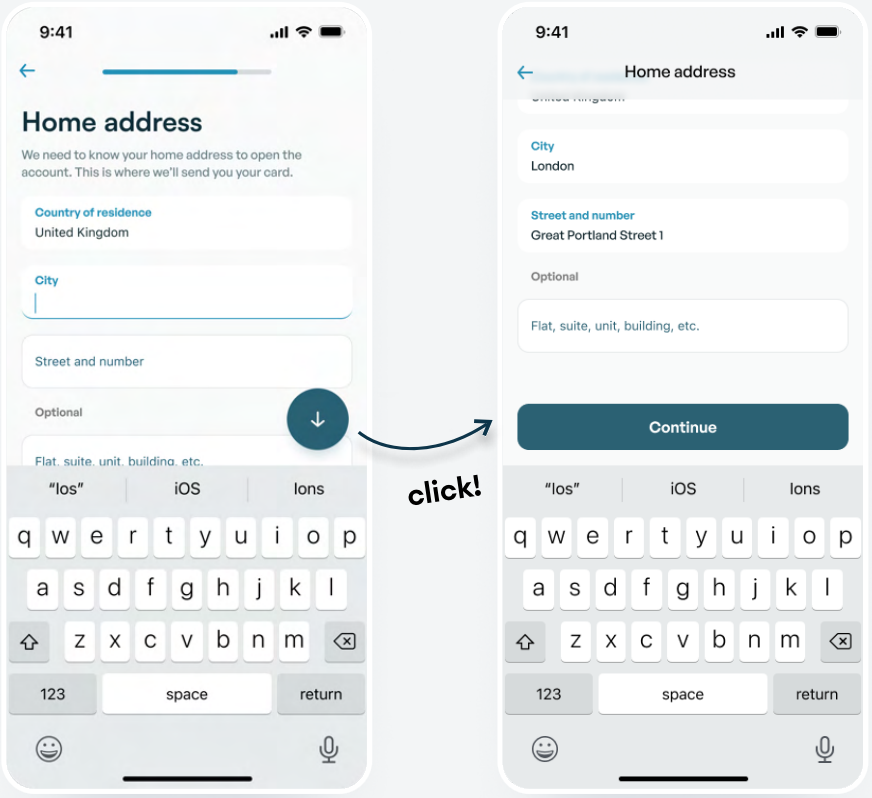
Next up, we'll have to get the personal details. If the process is long, you should always break it into smaller steps. Don't let users stay on one screen for too long.

We figured the best idea would be to split the verification process into personal information, i.e., name, last name, and DOB, and follow up with address information.

Address Information

Not every app requires sharing such precise information and personal details. Still, in our case, we're creating an international bank account where we'll be managing our money, making payments, and opening currency accounts.

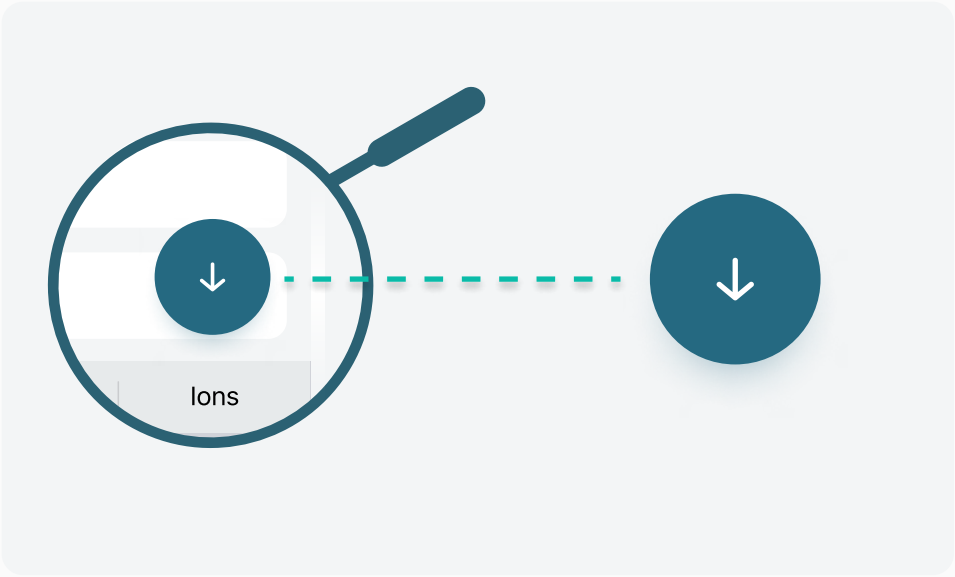
Every detail must be confirmed and valid.



1.4.2c -Legal information 3-4

1.4.2d -Legal information 4-4

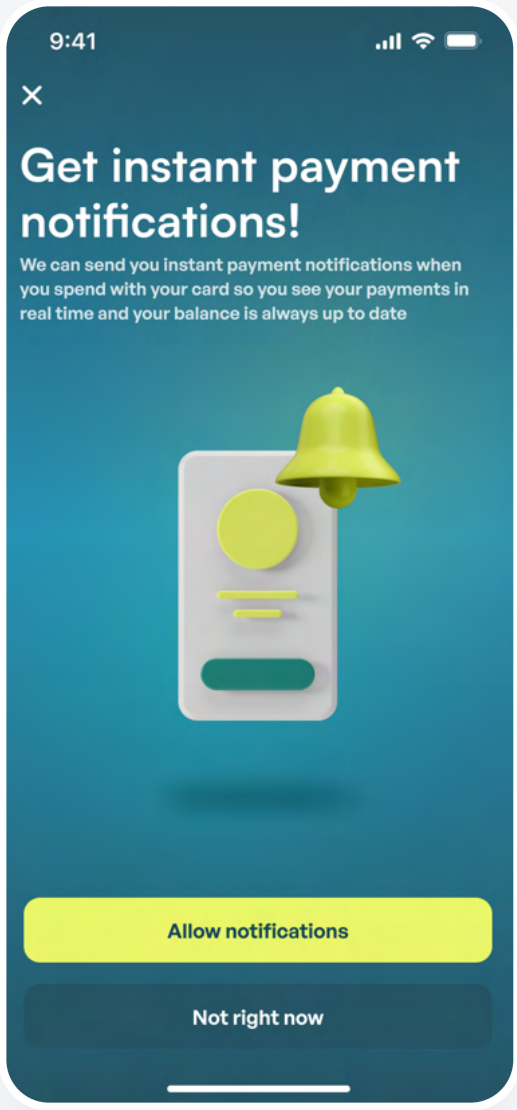
1. Anchor button



With screens requiring scrolling to continue the setup process, we can introduce a small anchor button that floats just above the keyboard and doesn't obscure the screen view. We can position it in the bottom right, above the keyboard, and anchor links it to another input field.

Firstly, by having it visible, we know that there's more to fill out. We could simply click on the button to automatically navigate to the following focused field without taking our hands off the keyboard and manually scrolling down. I know it might seem irrelevant to have it here, and while it's not a massive improvement of the UX, it's a nice little gimmick worth considering nonetheless.

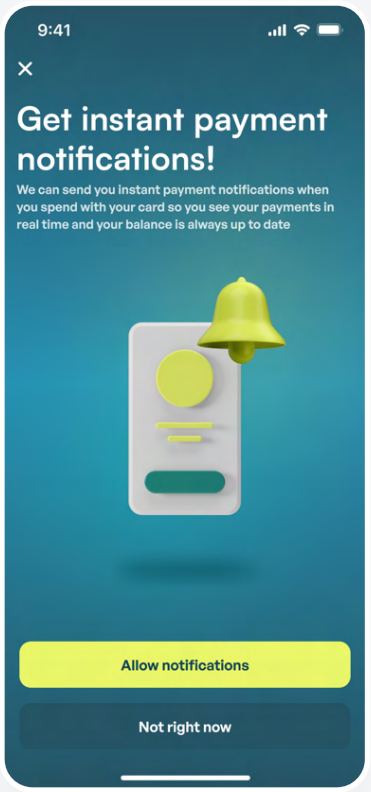
Finish this step by adding a Continue button.



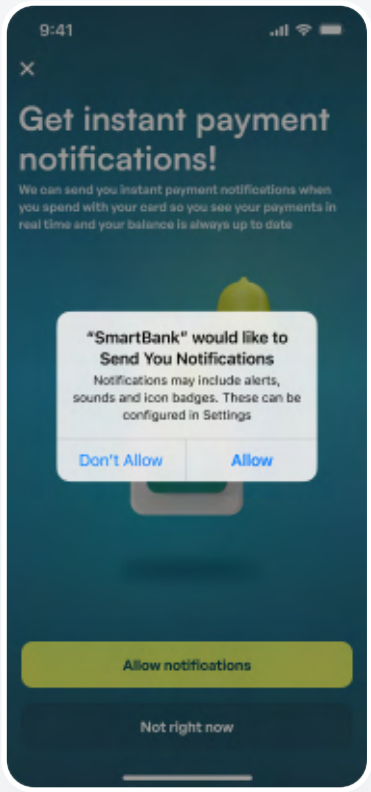
1.5.1a - Notifications on 1-2

Enable notifications screen

We have our user's email, phone number, personal details, and home address. We've warmed him up enough in the account creation process already. What better time to ask about the in-app notifications than now? We want our users to log back into our app as often as possible, so having a push notification system sending relevant push messages could help us do just that.



1.5.1a - Notifications on 1-2

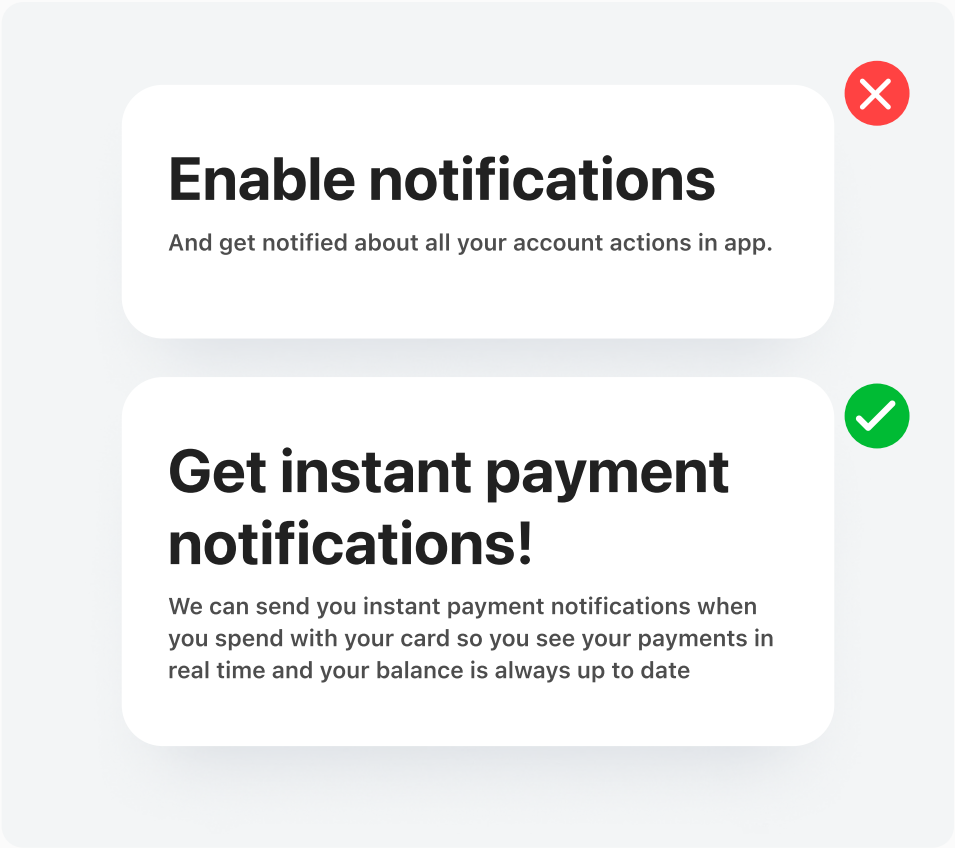


1.5.1b -Notifications on 2-2

When you are creating your splash screens, make sure to add brand visuals and accent colors. For promotional screens with inverted colors, remember to invert the buttons and play around with the opacity of your text.

If your description text is secondary, make it less prominent. Titles should be short and easy to understand.

UX Copy



A smooth user experience comes not just through an organized and clear journey but also from understandable, jargon-free language. Content is critical for converting sales, establishing a consistent voice for the brand, and driving a positive user experience. When it comes to app copy, clarity and consistency are both kings, and you might consider hiring a UX writer to ensure that all your text is as good as it can be (example on the next page).

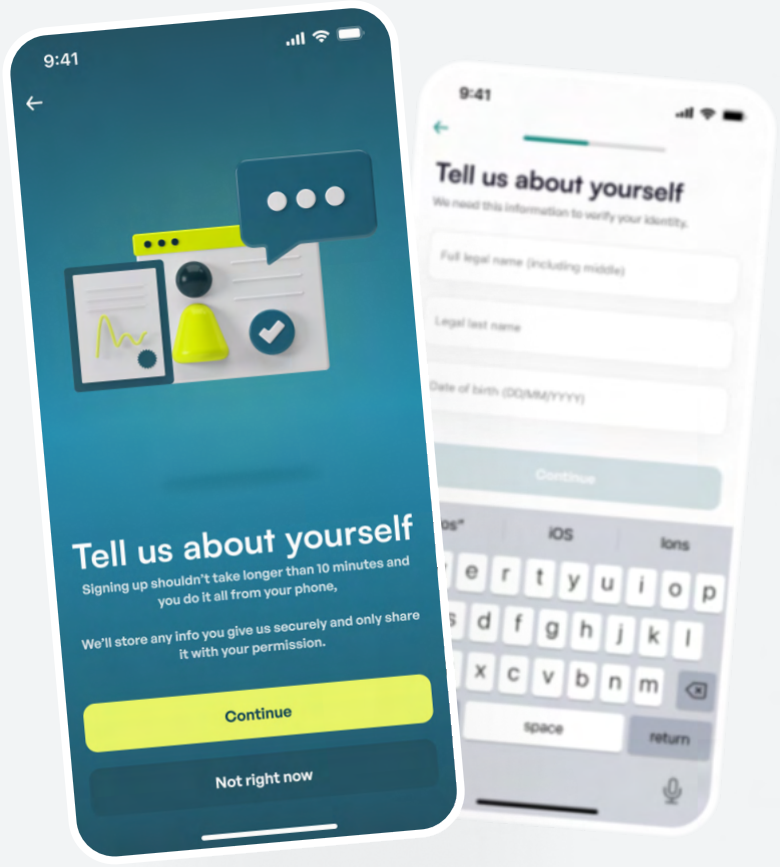
It’s always helpful to build user personas and deeply analyze our target audience, drilling down not just into demographics but also their behaviors (psychographics), pain points, and motivations.

Push notifications

We need to use an alert notification with a matching request text for every time we need to access the phone's settings. It's required by iOS, and users need to give explicit permissions before we can proceed.

As for the push notifications, be careful with the number of push messages you send to your users. We don't want to annoy them. Many apps don't follow the Apple guidelines and send out multiple, repetitive notifications about similar actions.

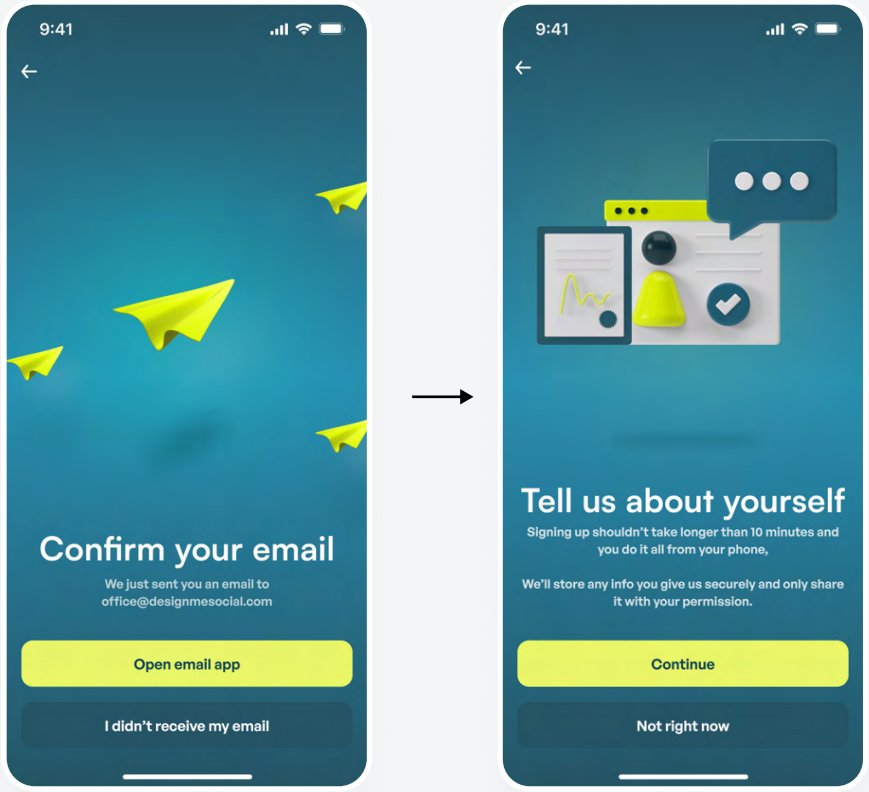
We're pretty much done with the first part of our account onboarding and can move on to the next step.



Legal information

THE STRUCTURE

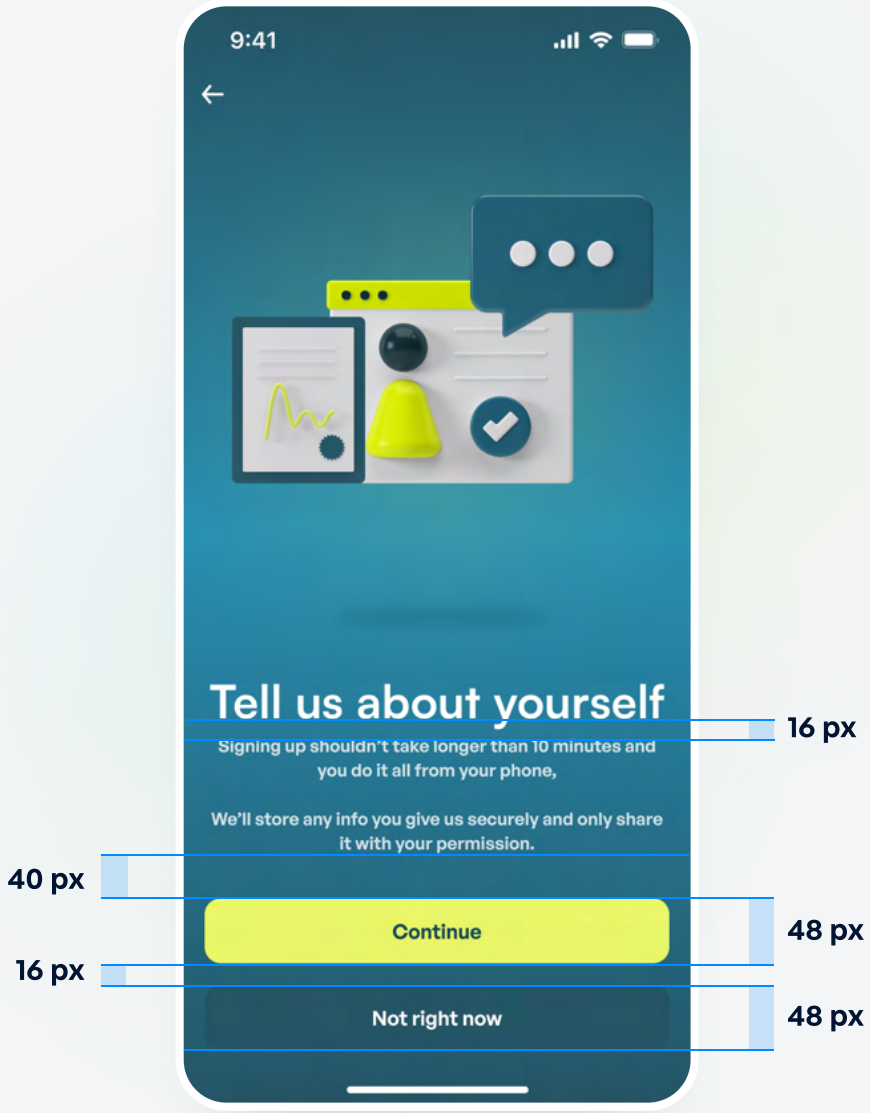
Personal details - the structure



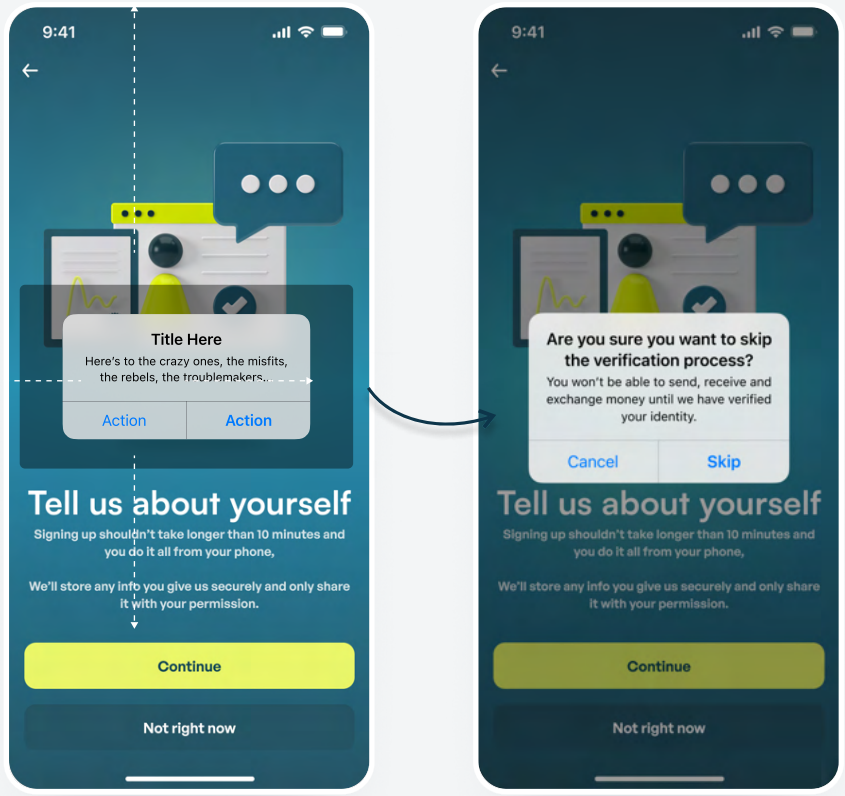
1.2.3 - Confirm your email

1.4.1a - Get started 1-2

First off, replace the text and update our image. If your auto-layout is set up correctly your text should resize the container and push it to the top. Look at the padding values on our left (original) frame and make sure your replicated design looks the same.



1.4.1a - Get started 1-2

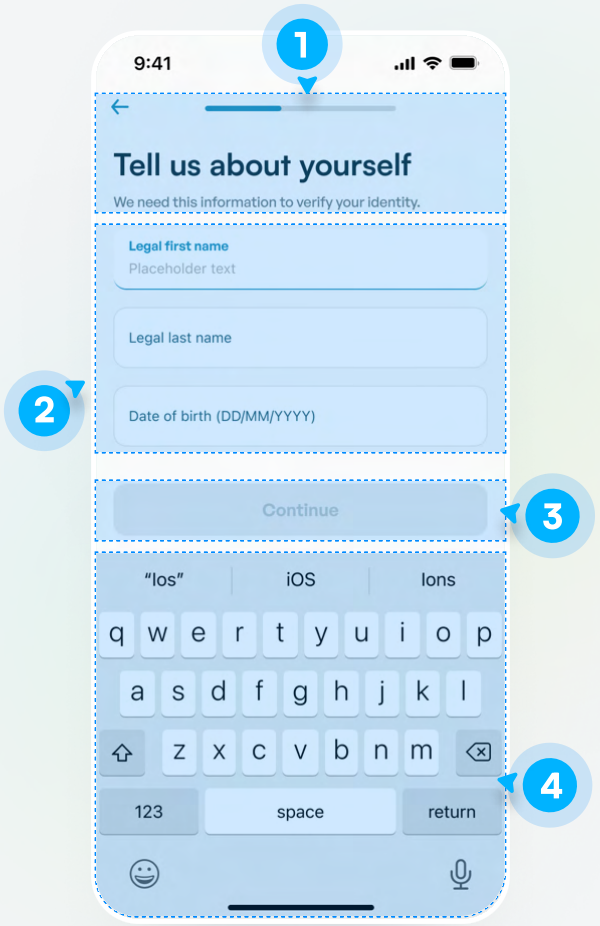


1.4.1b - Get started 2-2

1.4.1b - Get started 2-2

For the alert notification all we need to do is copy our iOS 15 UI Kit alert template, paste it on our frame, and resize it to fit our frame. It should automatically align itself to the middle of your screen. Choose the “2 Action Buttons” Action count, replace the text and that’s it. We’re ready to move on to the next part of our flow.

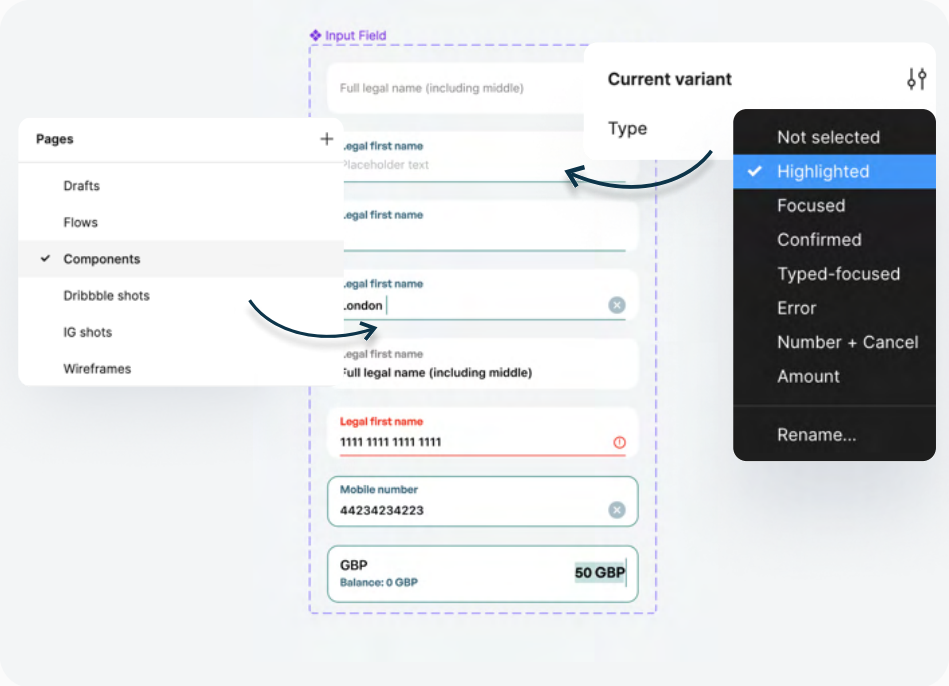
Tell us about yourself



1.4.2a - Legal information 1-4

All the screens using input fields are very similar to each other. To design them, we just need to reuse the existing components. You should be able to find all the inputs on the Components page.

Input field component variants

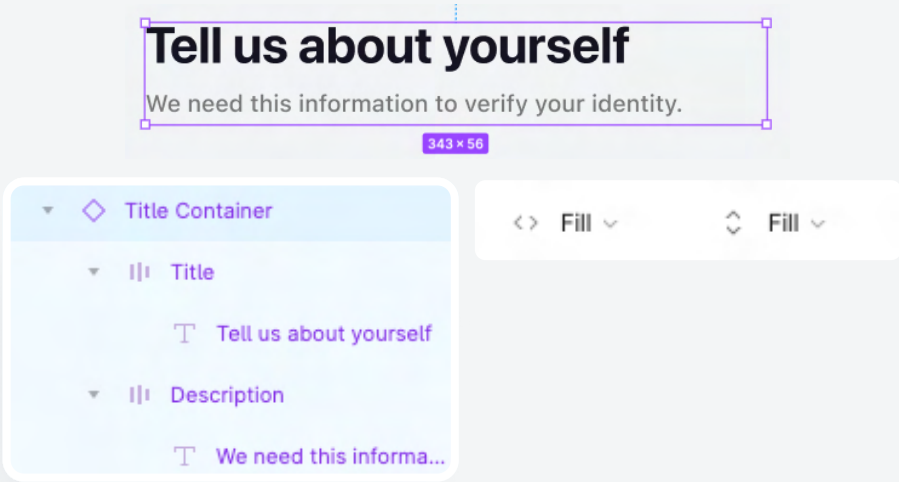


To make your life easier, whenever you start a new project and you need to design an input field, a button, or an interactive element that has multiple different states and variants, just design all the states you can think of at the beginning. Your developers will need to know what different states look like, and you'll need to create them to show different user flows in your app. You'll also need them for the prototype to show different interactions.

I created them for you, but it would be best if you could try and replicate them yourself.

1. Text Container

For every text container we can reuse the same component so let's do that. Navigate to our components page, copy the text container and paste it on the frame. Place it 16px from the left and 90 px from the top. Copy and paste our back button and place it right below our status bar (44px from the top of the frame.

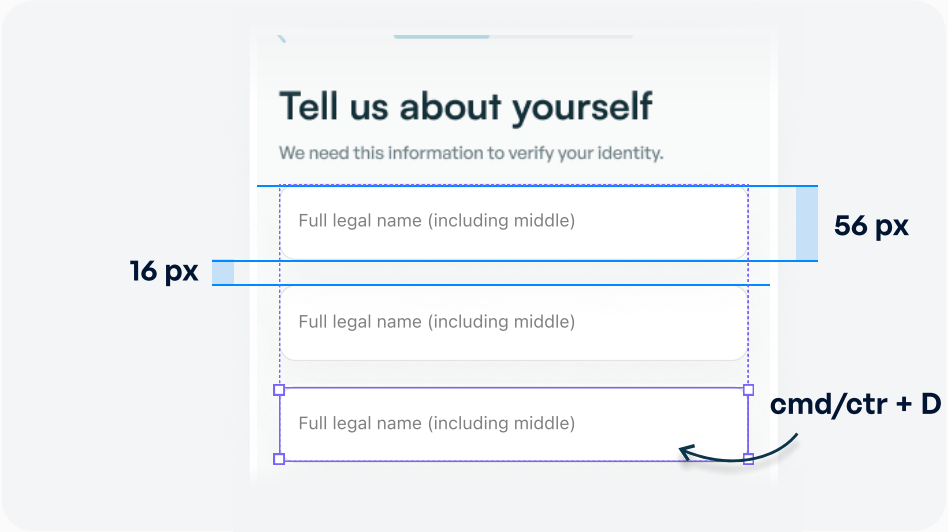


You can make components from every interface element, and I suggest you do that. Making global changes is so much easier if you can just edit one master component, and every interactive element will automatically change to reflect the changes made its master.

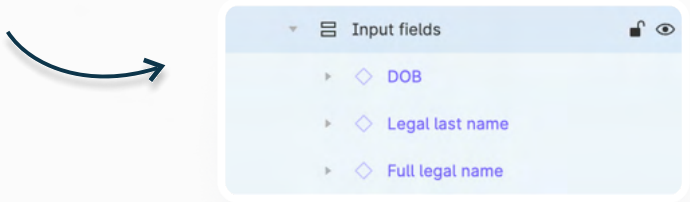
You can make a whole new design in 10 minutes when all your interface components are dynamic and linked.

2. Input Field

We can copy one from our previous frame, “What’s your email,” or paste it directly from our Components page for our input fields. Place it 24 px from the text container and apply auto-layout (“Shift” + “A”), rename the layer to “Personal information container.” This is where we’ll contain all our input fields. Select our input field and click “Cmd/Ctr” + D.

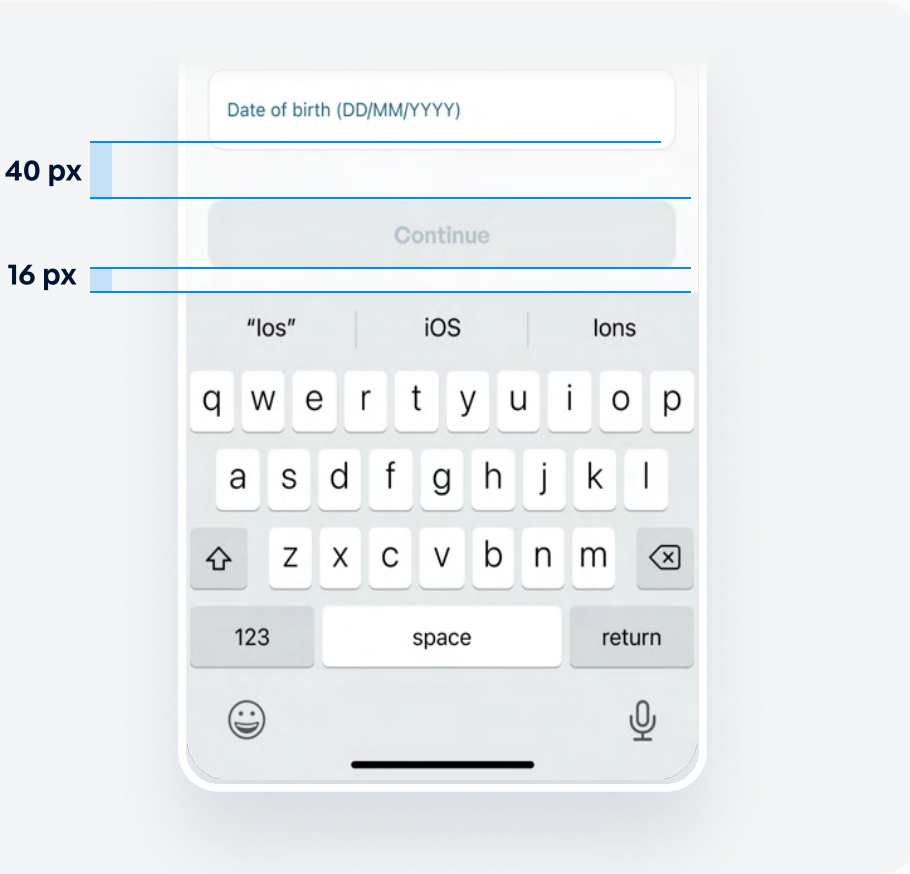


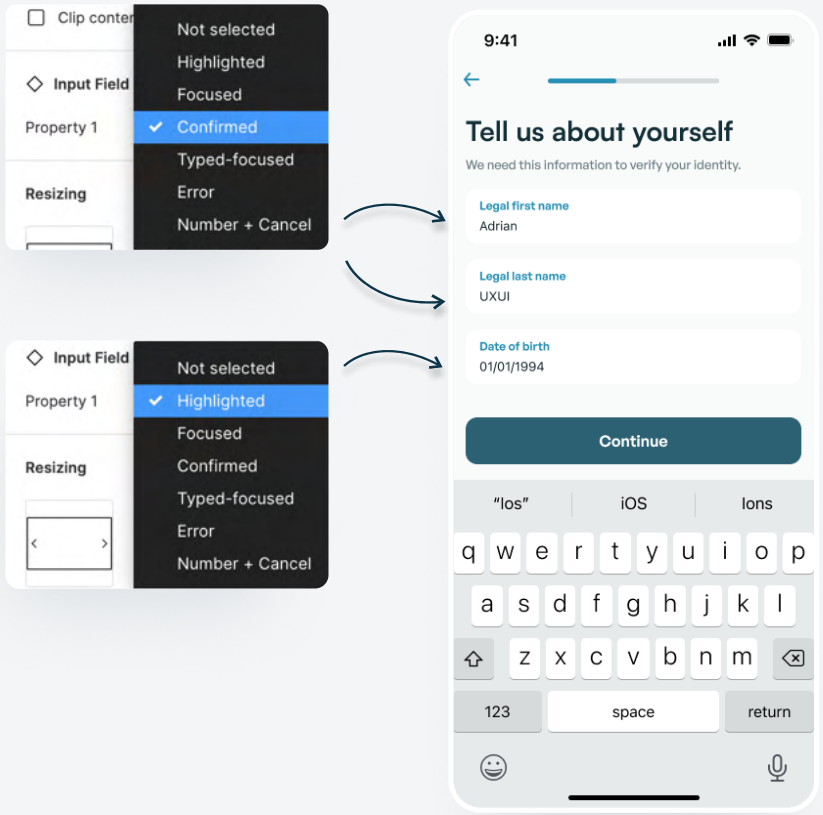
Change the text in each of the input fields to match our design. Rename the layers accordingly to find them quickly, and your file is well organized.



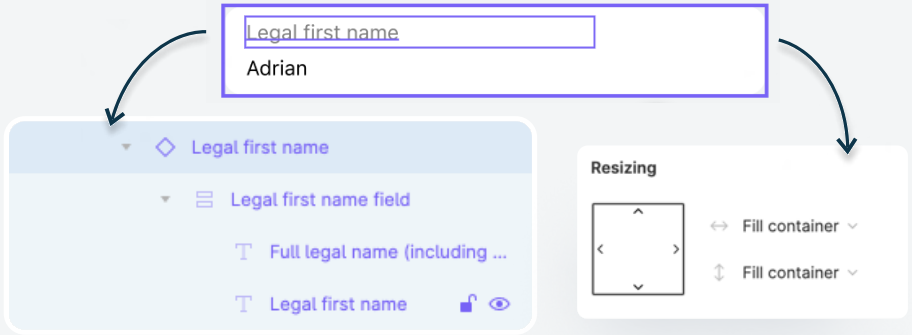
3. Continue button + iOS Keyboard

You probably know this by now - copy our button from the “Components” page and change the Variant to “Disabled.” Position it 24 px from the text layer. If your input fields and spacing between them is set on a 4 pt grid, meaning all the values are multiples of 4, then your spacing between the button should be 40 px. Place your button 16 px from the keyboard. That’s all we need to do here.



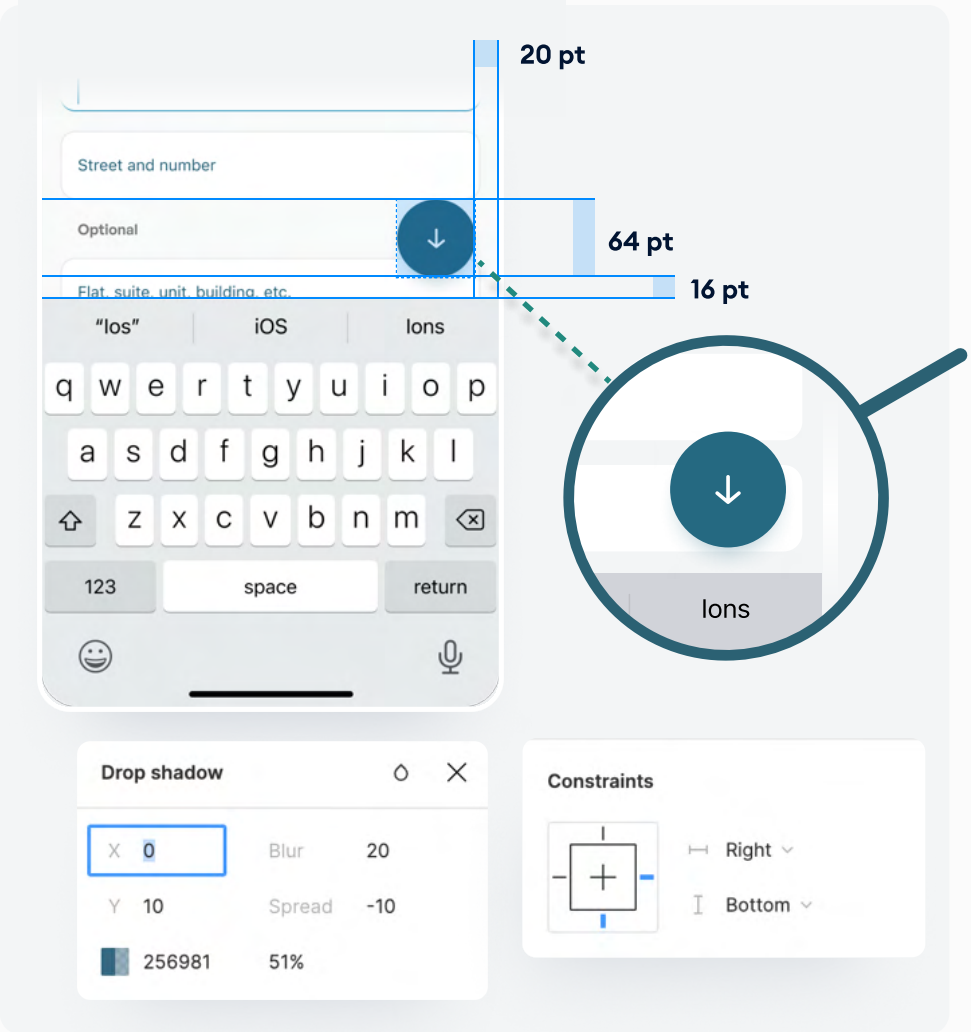


All we need to do to design our confirmed input screen is either update the input's design manually or by selecting each of them and choosing a different variant state. Click on the properties panel on the right and select the "Confirmed" variant on the top two input fields and "Highlighted" on the last one. We just want to show our developers all the possible states and their appearance.



If you want to recreate the input fields yourself - try this.

- [x] Add a text layer -> Legal first name -> Change its style to "Footnote 13px" -> reduce opacity to 60%
- [x] Add a text layer -> (your name) -> Change it's style to "Footnote 13px" -> leave opacity at 100%
- [x] Select them both and apply auto layout
- [x] Resize the auto layout selection to 341 px width and 56px height. Center your auto layout in the middle
- [x] Apply color fill - white and round the corners to 8 px
- [x] Set every layer within the container to fill container.



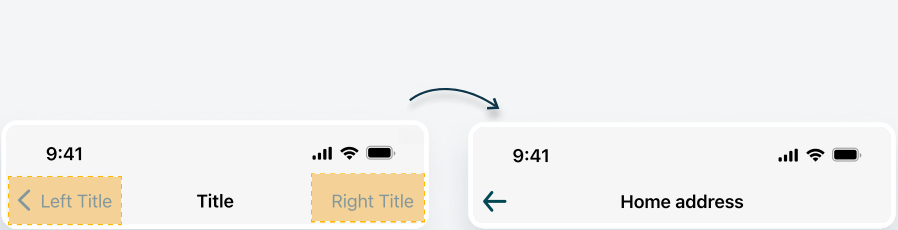
We can copy the same exact layout from our “Tell us about yourself” frame for our two following screens. Add another input field and remove the “Continue button.”

Instead, let’s create a circle with 64 px width.

Apply a soft drop shadow - you can look up the exact values on the snippet on the previous page (Drop Shadow card). Constraints should be set to “Right” and “Bottom” since we want our button to stick to the right border of the screen and position relative to the keyboard below.

The next screen’s header is a bit different. It is different in the sense that it uses the centered top navigation native to iOS.

The bold title becomes a centered heading with a semi-transparent blurred background when you scroll past a specific viewport (around 10%). This is very common for iOS apps, so let’s replicate the same design pattern.



Whenever I’m talking about iOS native elements, you should refer to the iOS 15 UI Kit for Figma and look for a suitable component. Copy it and paste it back into our frame. Let’s play around with it and adjust the toolbar, so it matches our design.

Let’s remove the right action button, remove the text from our left button, and replace the chevron with our back arrow. Change the text from Title to Home address.

I made a component out of it and added it to our header variants so we can reuse it everywhere we need to. You can look them up on our Components page.

If we notice a similar design pattern, I will break it down and let you know which frame from our original Figma file should you refer to adjust the designs. I will also link the page number next to the descriptive text so you can quickly scroll back up and see our breakdown.

We will not repeat the same things over and over again to bore you down or, for that matter, make this ebook 1000 pages long 😅

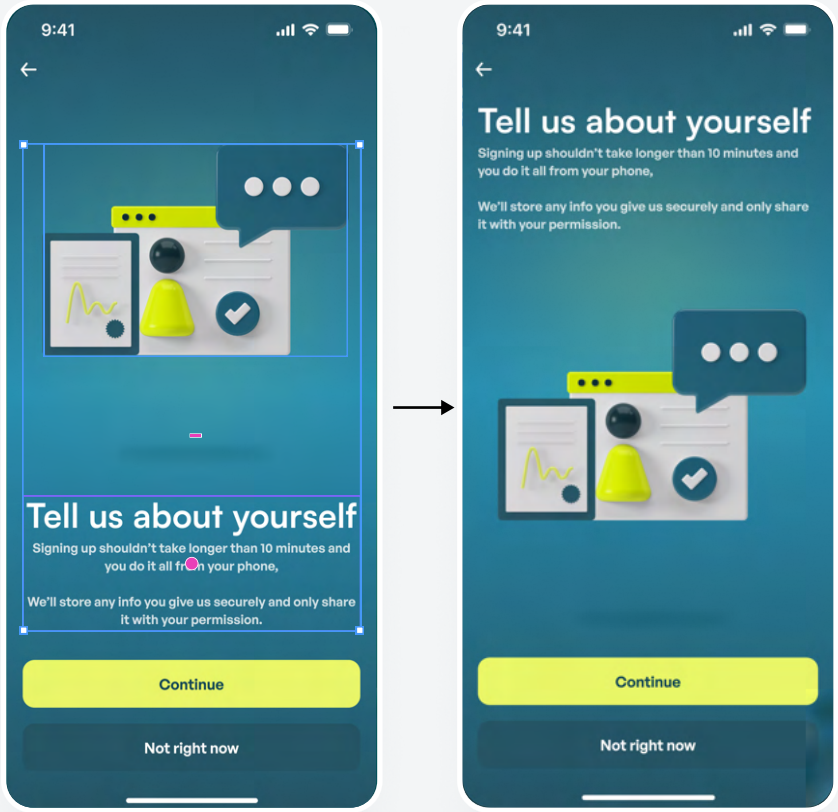
You shouldn’t take this guide word for word. You should try and replicate all the designs you get in the practice files yourself. It’s your shortcut to becoming a better designer.

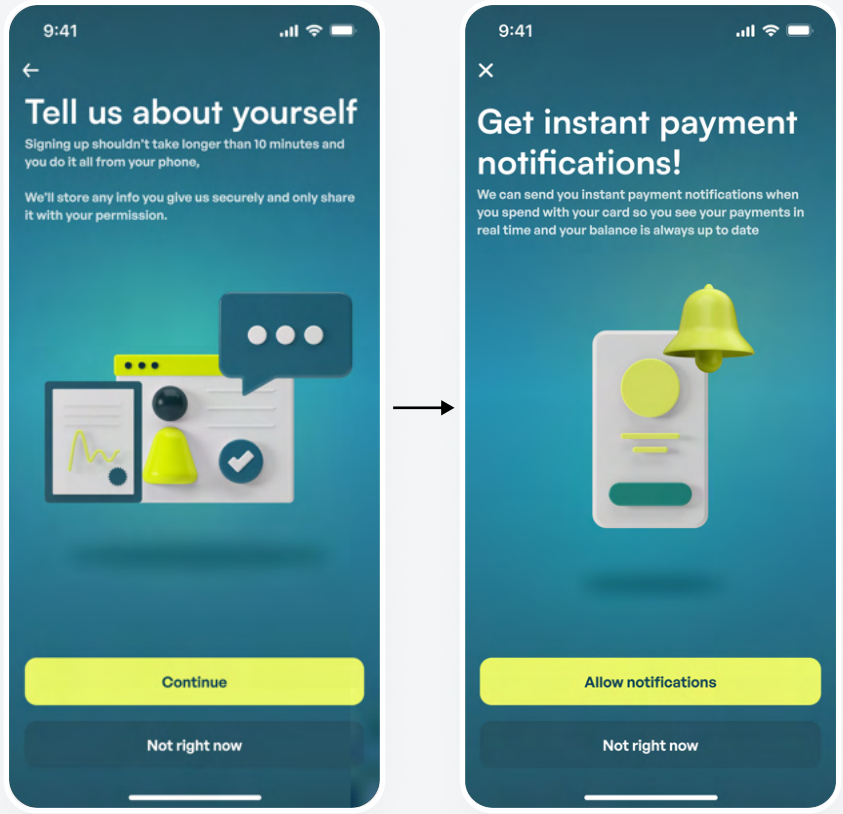
Trace it or copy it, and remember the patterns. Refer back to this part when needing guidance 🙌



Enable notifications screen

Select your image and the Title container. Click and drag the red circle down or up to reverse the placement. Position your Title container 96 px from the top, select each text layer, align it left, and replace the text. Lastly, copy a back button from one of the previous screens and place right under our status bar and align it to the left border.





Replace our image and reposition it in the middle. Change the text on our CTA buttons.

Double-check if you've got everything appropriately designed. If so, let's move on to discuss another flow. Great job! 🙌

Chapter recap

- Alerts disrupt the user experience, so it's essential to use them infrequently and only deliver critical information.
- If your description text is secondary, make it less prominent.
- Titles should be short and easy to understand.
- Content is critical for converting sales, establishing a consistent voice for the brand, and driving a positive user experience.
- A smooth user experience comes not just through an organized and clear journey but also from understandable, jargon-free language.
- Always leave one door open from every flow so your users don't feel locked in and can navigate to where they want freely.



Verify identity

THE INTRO

Verify identity - the theory

Verifying account holders is crucial for protecting user data and our business's reputation. As our world becomes more digitalized, we expect fast, efficient, and immediate results. If an app requires you to follow an exhausting and tedious verification process, the chances are you won't stick around for long.

If you like it or not, this needs to be done due to regulatory requirements to help avoid errors – or fraudulent activity that our app might be used for.

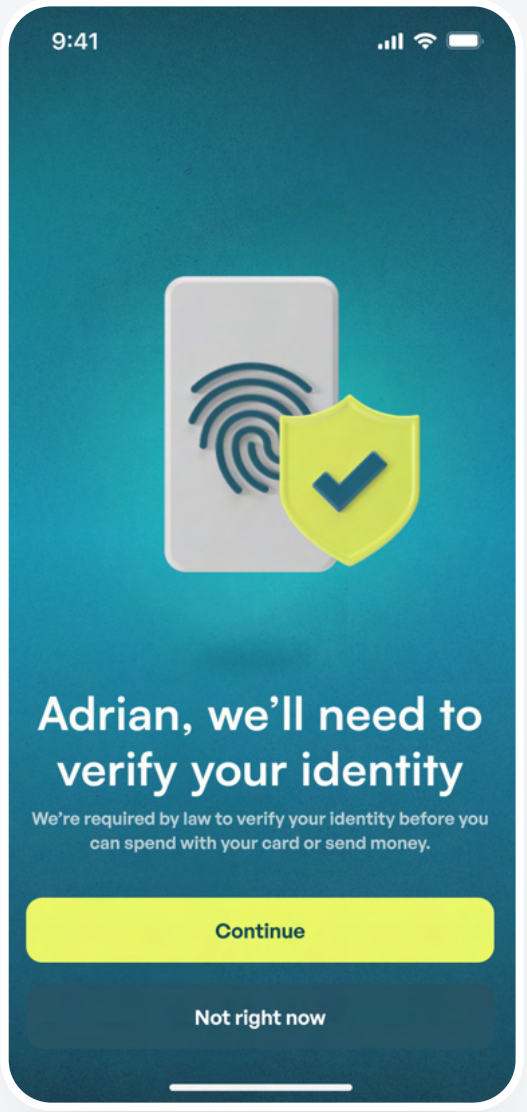
Even though verifying the identity is optional at this point, we have to remind users that they will not be able to send, receive and deposit money until the verification is complete.

Having said that, let's add a second button to skip the verification for now. We don't want to make our users feel trapped at any step of the onboarding process.

Finance might be pretty complex to the average user by its very nature. We want to visually guide them through the onboarding process instead of spamming them with a continuous setup flow.

Show them, screen-by-screen, what should happen next. Use simple calls to action throughout to guide them along the way.





1.6.1 - Verify Identity - Splash

Verify your identity methods

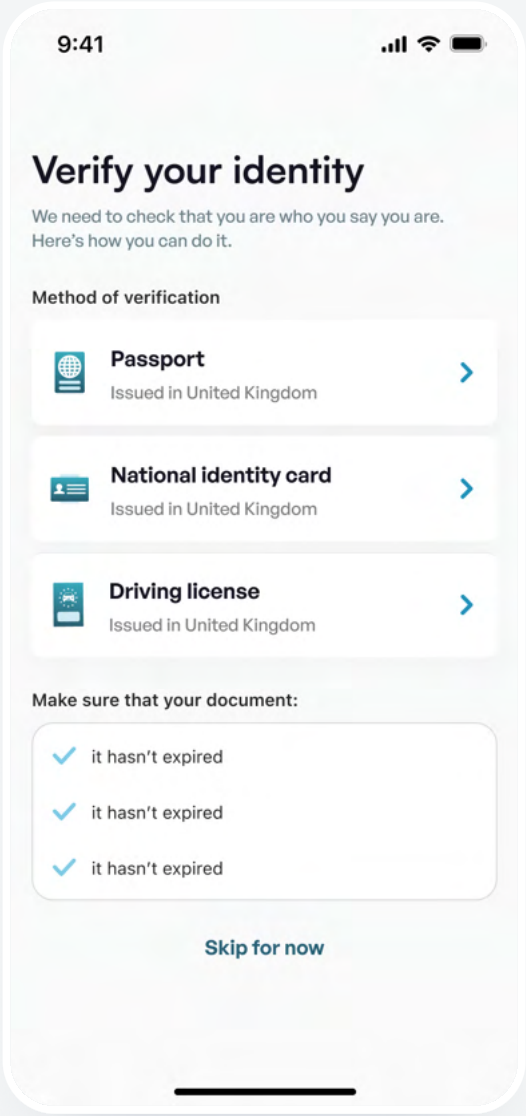
Onboarding in Fintech is critical because it may make or break your product. According to Thomson Reuters research from 2020, it takes an average of 24 days for a bank to complete the customer onboarding procedure, and 89 percent of clients have had a terrible KYC (know-your-customer) experience due to the length of the onboarding.

As a result, at least 13% of clients switch to another service provider. So, what should we do to avoid falling into a similar trap and losing our customers before gaining them?

As much as possible, we should try to simplify and shorten the process through the use of machine learning solutions such as facial authentication and data extraction from ID photos. First, you scan the ID with your phone, then ask your users to record their faces.

But before we move to that step, we should show them a screen with all available verification methods. As a standard, we can identify a person by looking at his passport, national identity card, or driving license.

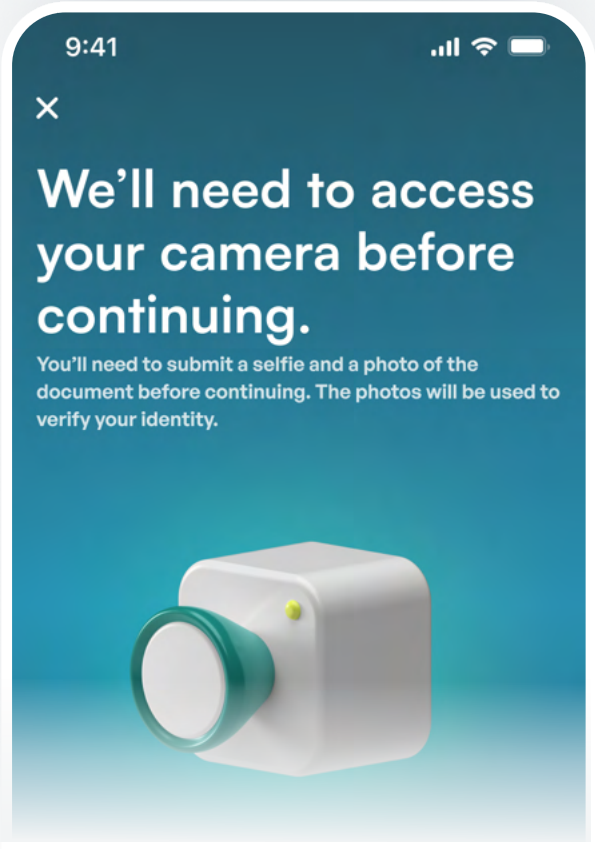
However, the document should be issued in the country of your residence, as previously stated in the account setup.



1.6.2 - Verify identity type

Camera access

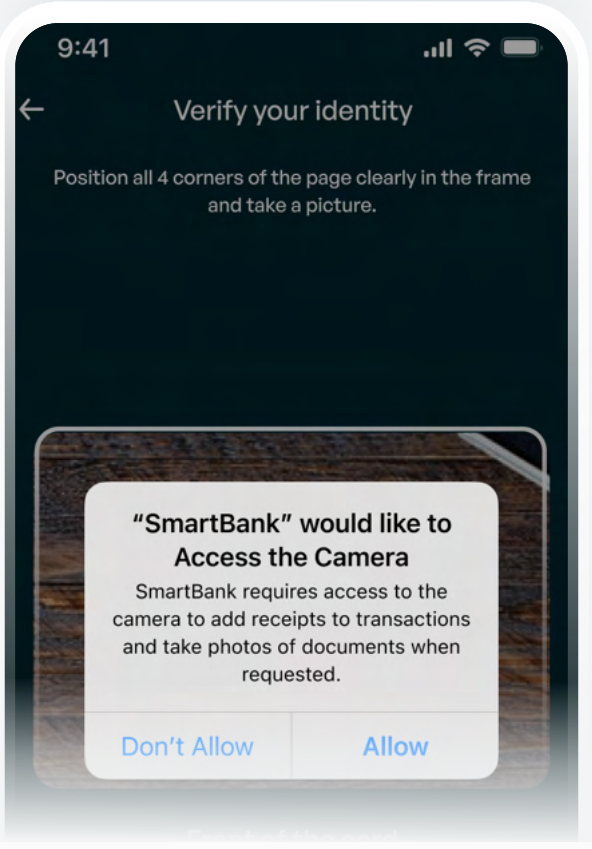
The worst thing an app can do is bombard users with permission requests without any explanation when it comes to requesting permission. Necessary permissions (the app cannot work without those permissions) should be requested up-front, while secondary permissions may be requested in context. Let’s offer our users additional context on why we need to access their cameras.



Access the camera alert

After choosing an option the user is taken to the next screen where he needs to give permission for camera access. Let’s fire up that iOS alert modal with access request.

In order to verity identity we need to provide the app with a photography of both sides of the chosen document.

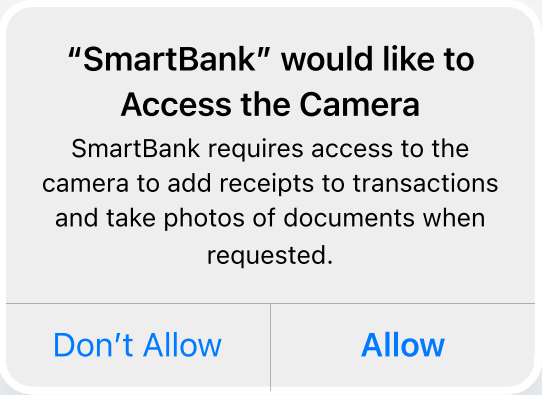


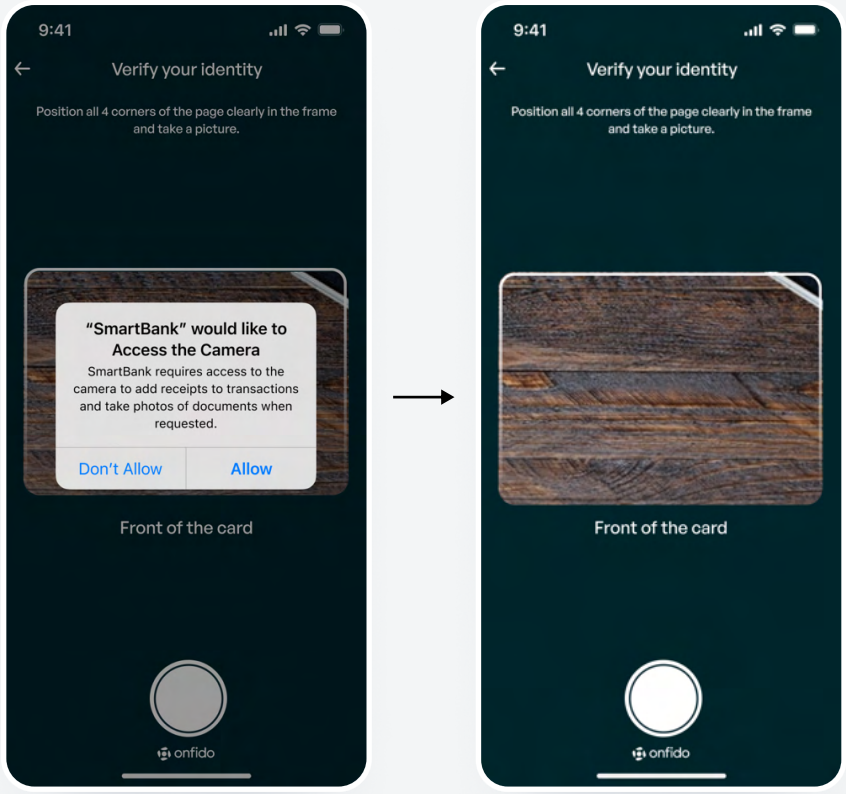
Alert messages - what text to use

Create short, logical button titles. Aim for one- or two-word titles that describe the result of selecting the button. Prefer verbs and verb phrases that relate directly to the alert title and message.

Place buttons where people expect them. In a typical two-button alert, people are most likely to tap the button on the right side, which means that the Cancel button should be placed on the left side.

The button text will differ based on the context of the action taken.



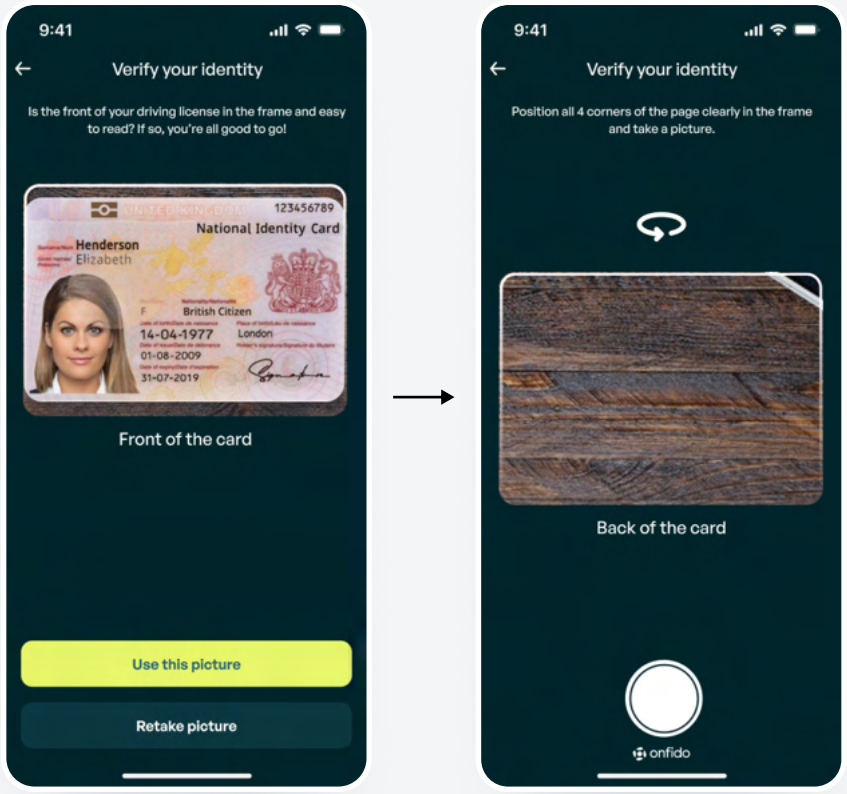


1.6.4a - Verify Identity 1-5

1.6.4b - Verify Identity 2-5

We want to ensure we aren't accidentally cropping out an essential piece of information from the ID document. By leaving the corners and edges of the document intact, we'll be able to find all of the necessary information we need.

Let's add a border around our frame to highlight the image are and add a short description that explains the process in more detail.



1.6.4c - Verify Identity 3-5

1.6.4d - Verify Identity 4-5

Verify your identity - confirm picture

After the shot is taken, we should ask to confirm the picture or prompt users to retake it if needed. If you take a look at the text up top, it also changed to reflect that action.

Do the same for the back of the card. Position the ID in the frame, take a picture, confirm, and move on to another step.

Uploading screen

People don't like waiting unless there's a good reason to, and the longer the wait is, the less that reason stacks up. This is where loading indicators come in. A good loading indicator lets users know the application is working, makes the wait tolerable, and gives them something to look at.

"The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time."

Nielsen's 10 usability heuristics for UI design.



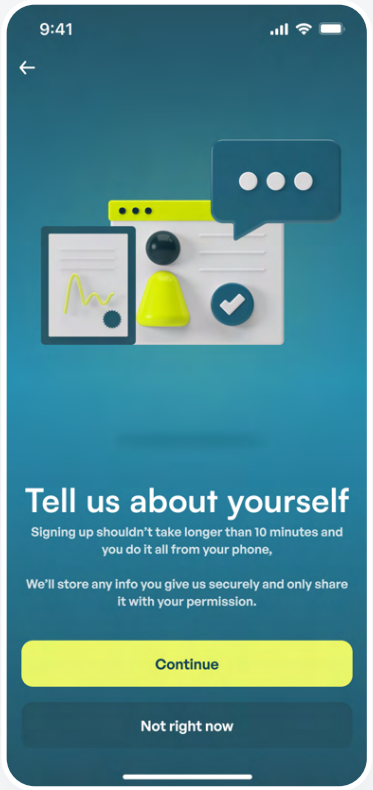
1.6.5 - Loading screen



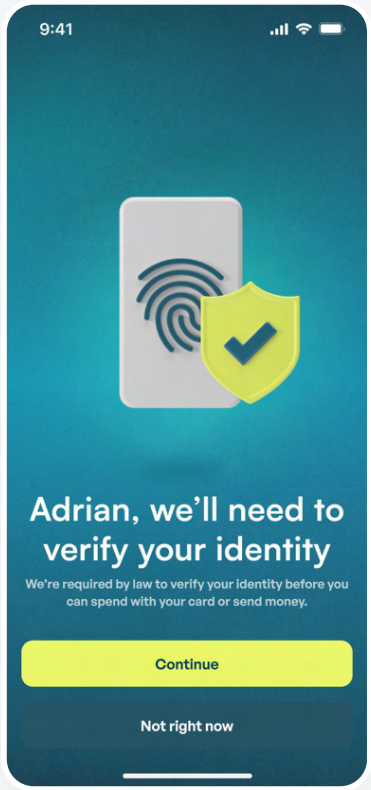
Verify identity

THE STRUCTURE

Personal details - the structure

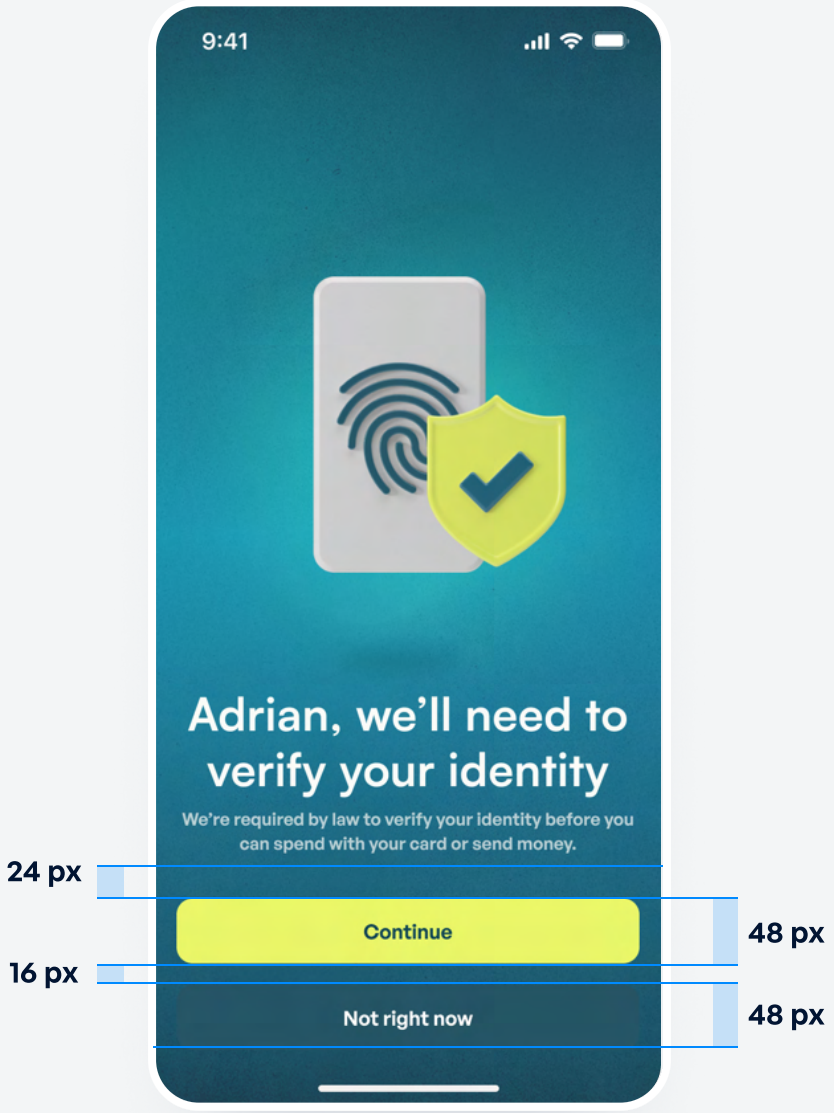


1.4.1a - Get started 1-2

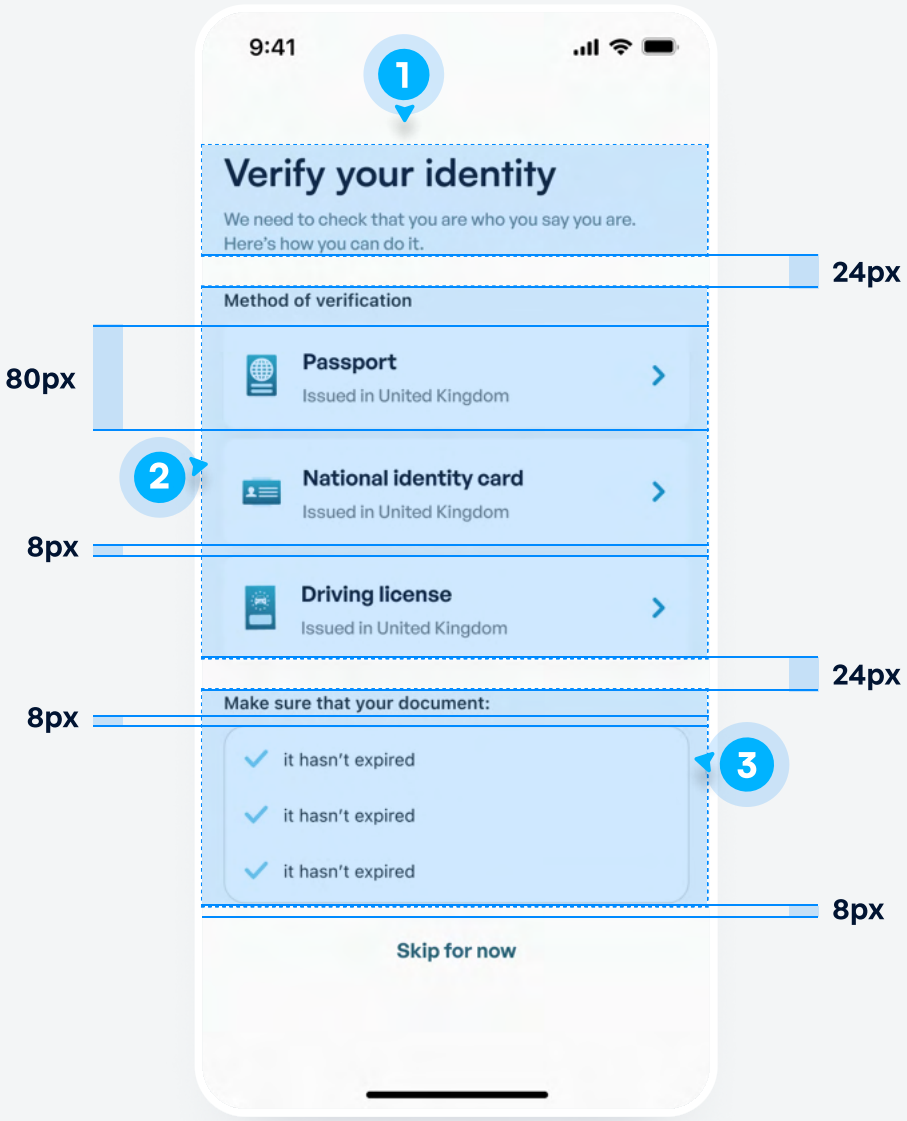


1.6.1 - Verify Identity - Splash

To replicate this splash, we must copy our previous “Passcode set up” frame and replace the text of our title and description. Replace the image. What’s the shortcut to replace the image and fit it into the same container? Copy it from our assets page, select the old image, and click “Shift + Cmd/Ctrl + Option/alt + V”



1.6.1 - Verify Identity - Splash



1.6.2 - Verify identity type

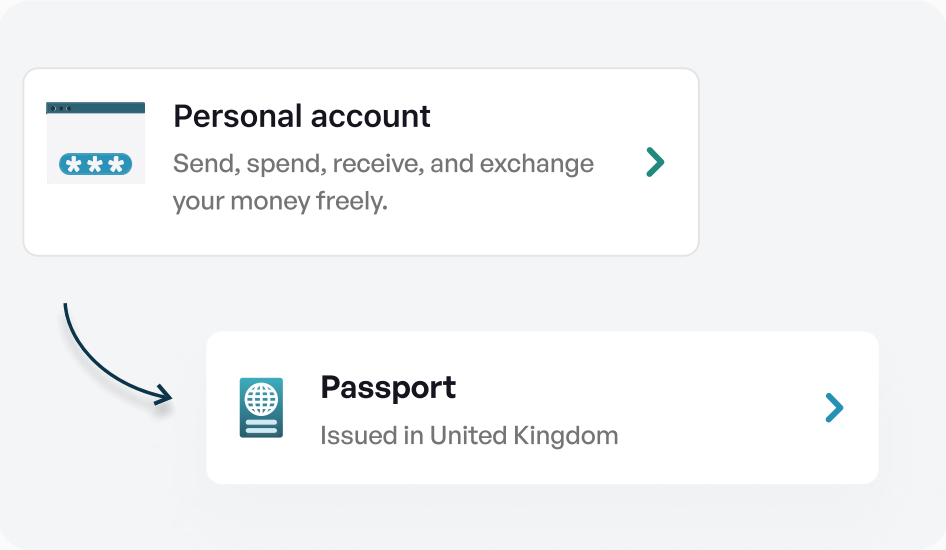
Verify your identity

1. Title Container

Let's copy one from our "Components" page and place it on our frame for our title container. Position 16px from the left and 96px from the top. Choose the one without the back button; there's nowhere to move back to in our example. Update the text, and we're done here.

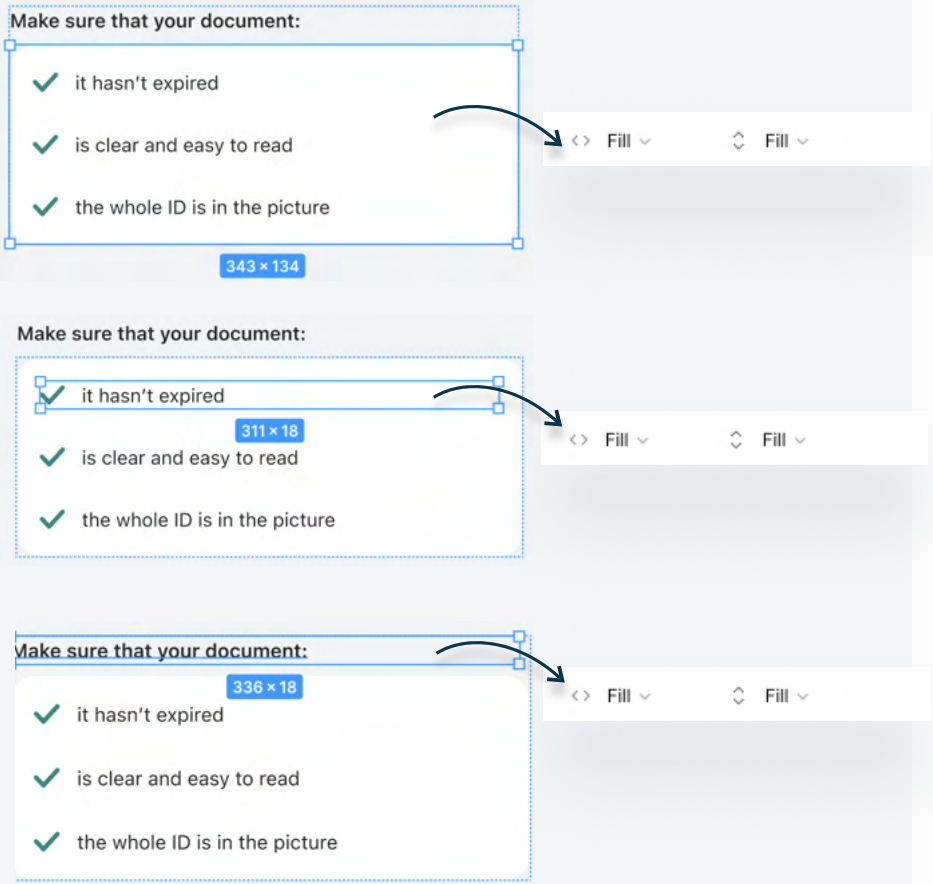
2. Cards Container

We can reuse one of our previous components used in the "Choose your account" frame for our cards. Adjust the images and text. Resize your frame to be 80px tall and go into each selection within the container to change the resizing properties - look at the reference file on the left and replicate what I did.



3. Checklist container

Icon lists are the best components you can practice your auto layout with. Create a text layer, apply auto layout, paste your checkmark and position it 12 px from the text. Select both and apply layout. Copy twice, select all three layers and use auto layout again. Check the resizing properties on all. Compare with reference file :)

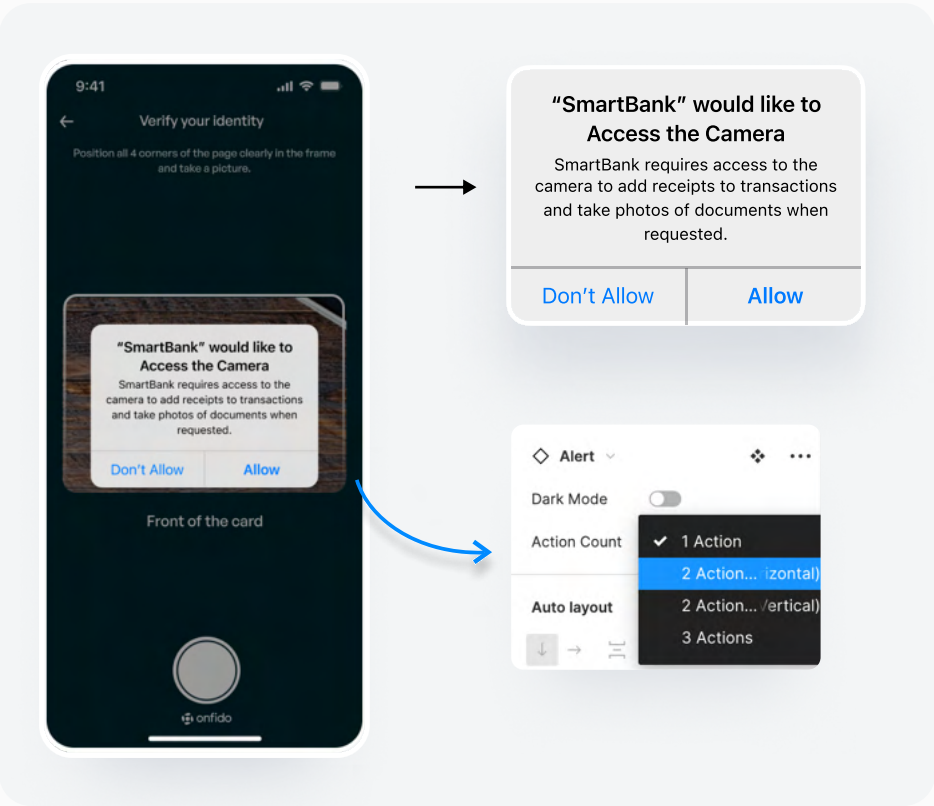


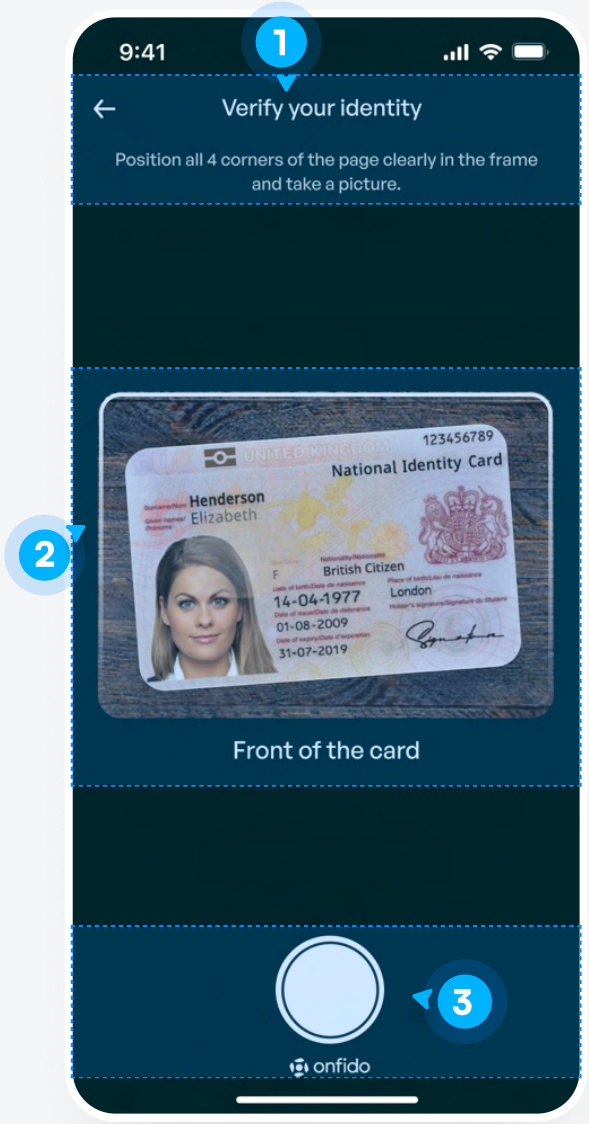
Allow camera access

We skipped the splash screen design here. Your homework is not to find a frame with a similar design and replicate it yourself. Hint - we designed it in the previous chapter.

Once you're done with that, let's move on to design our Verify your identity micro-flow.

First up, we have the alert, but we need to step back and design our main background first.





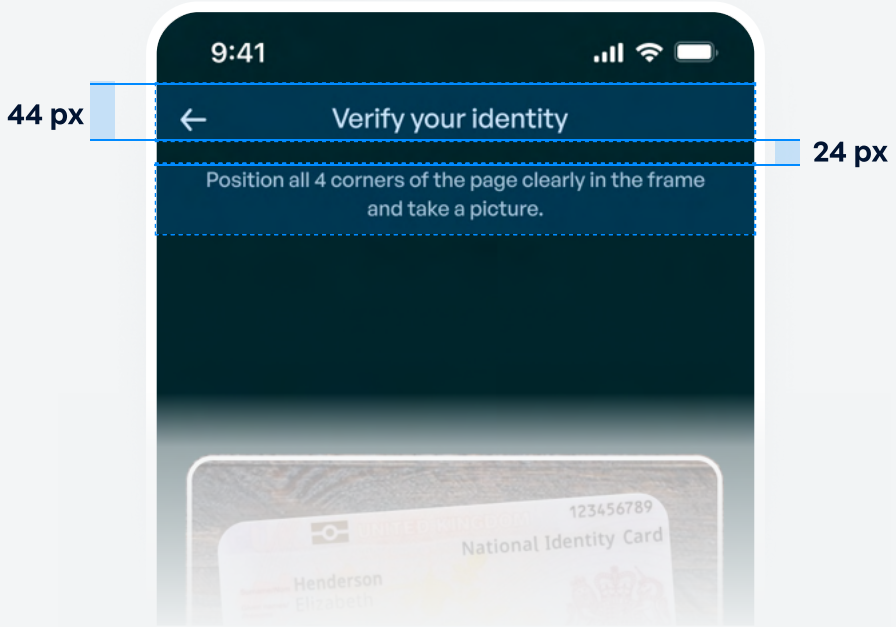
1.6.4b - Verify Identity 2-5

Verify your identity

We have a couple of building blocks that we're already familiar with so let's break them down below.

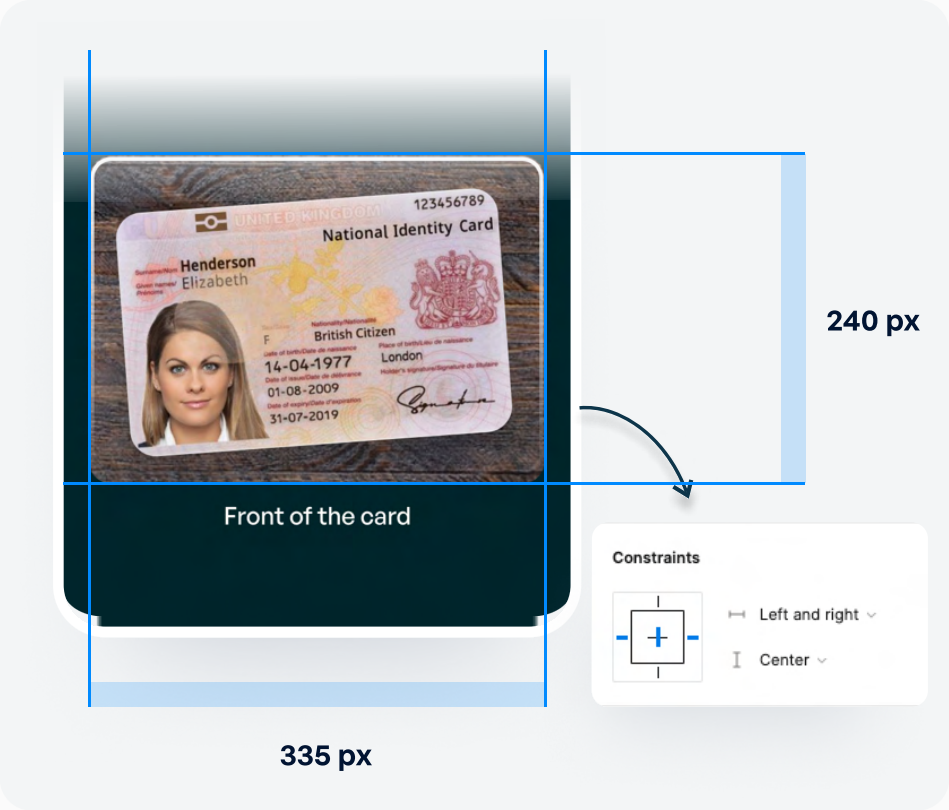
1. Centered title container

Use the header design from our previous frames - "Home address centered." Replace the text and change its color. Add a new text layer - make it Footnote 13 px in style and position it 24 px from our page title. You can group it up with our label and apply auto-layout, but you don't have to do it. They will probably be nested in separate containers.



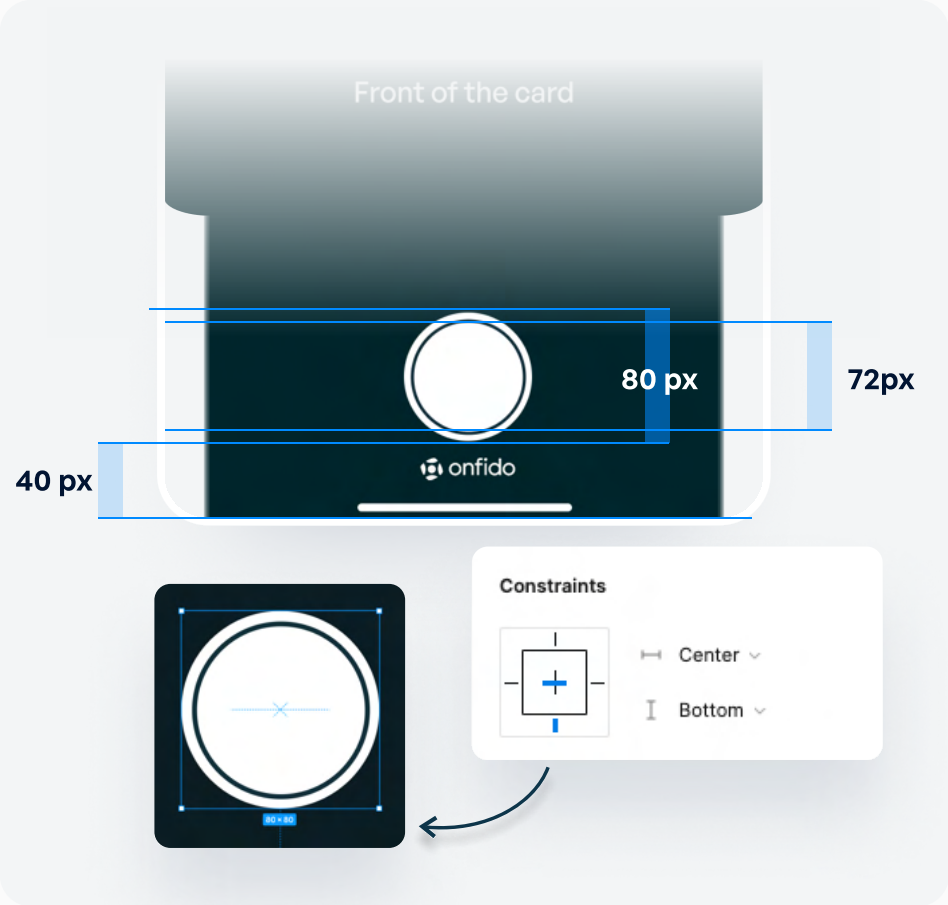
2. Camera preview

Create a rectangle with 335 px width and 20px margins on both sides, add a white border, and round the corners to 16 px. Position it at the center of your frame, both horizontally and vertically. Add a title below to confirm that this is, in fact, the front or back of an ID card. Apply constraints “Left and Right” and “Center” since we want our image to stay right in the center no matter what the resolution we resize it to.



3. Camera button

We need to create two circles for our button, one being 80 px wide and the second being 72 px with a 2 px black border. Place the bigger one behind the smaller one and group them both. Rename the layer to the camera button. Now the critical part. To retain the same ratio of a circle, we need to apply “Center” and “Bottom” constraints on all these buttons along with their group container.



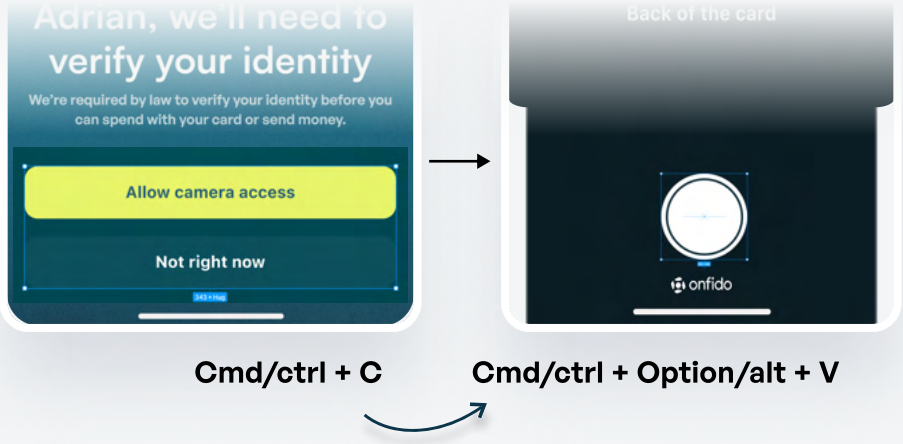
Verify your identity - Confirm screen

We'll only have to replace our camera button with two CTAs that we've already used in one of our previous examples for our next screen. Let's find a frame we duplicate it from.

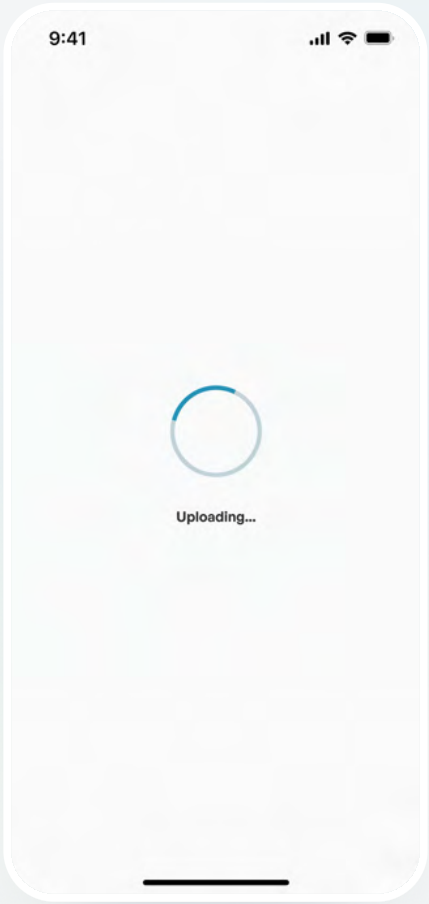
We didn't have to look long. Let's copy one directly from our Camera access splash. Now here's a little trick to do it faster. Select your button group and click "Cmd/Ctrl + C." Now go to your frame with the camera button, and select it.

Having your camera button group selected, click "Shift + Cmd/Ctrl + Option/alt + V." Your buttons should replace the camera button and retain the same distance from the bottom of the frame.

Now you only need to adjust the text of these buttons, and we're all set! Let's move on to our last screen from this flow.



Verify your identity - Uploading screen



1.6.5 - Loading screen

For our upload screen we just need to add two circles, one filled out partially to indicate a loading motion and a simple text layer titled "Uploading".

How to create a loader

1. Create a circle

2. Remove fill and add a 5px border, apply color



3. Select the shape and zoom in enough to see the “Arc” text appear.



4. Click on “Arc” and drag it out to sides to change the angle



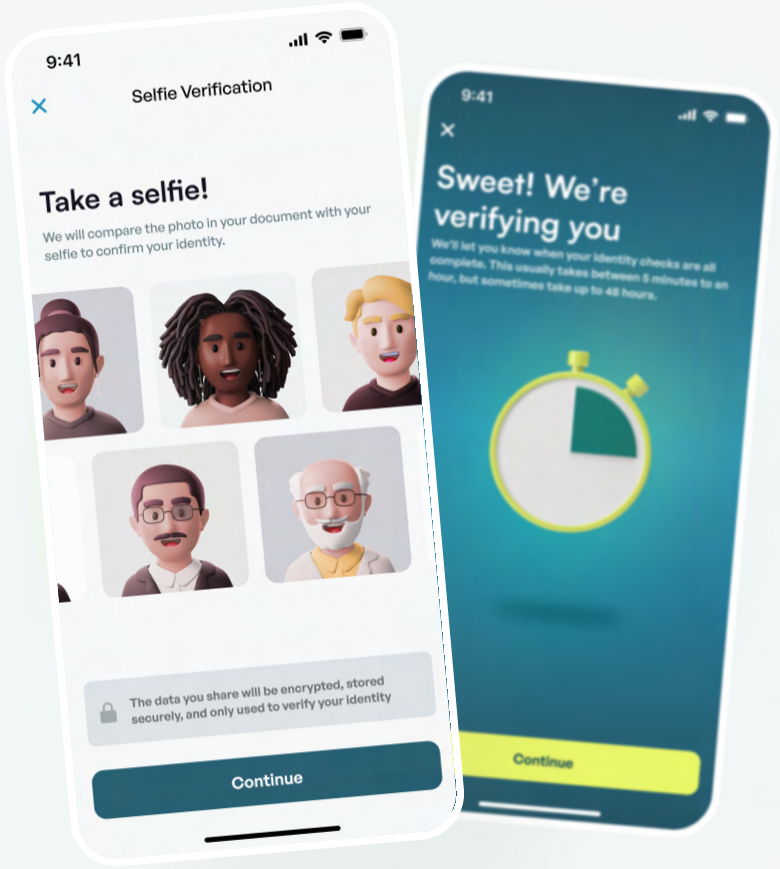
5. Double-click the shape and click on the anchor point in the middle. Remove it by clicking delete



6. Change the color and position your full outline circle behind. That’s it!

Chapter recap

- Verifying account holders is crucial for protecting user data and the business’s reputation.
- We want to guide our users through the onboarding process visually.
- Use simple calls to action throughout to guide them along the way.
- Necessary permissions (the app cannot work without those permissions) should be requested up-front, while secondary permissions may be requested in context.
- Create short, logical button titles. Aim for one- or two-word titles that describe the result of selecting the button. Prefer verbs and verb phrases that relate directly to the alert title and message
- Place buttons where people expect them. In a typical two-button alert, people are most likely to tap the button on the right side, and the Cancel button is on the left side.



Selfie verification

THE INTRO

Selfie verification - the theory

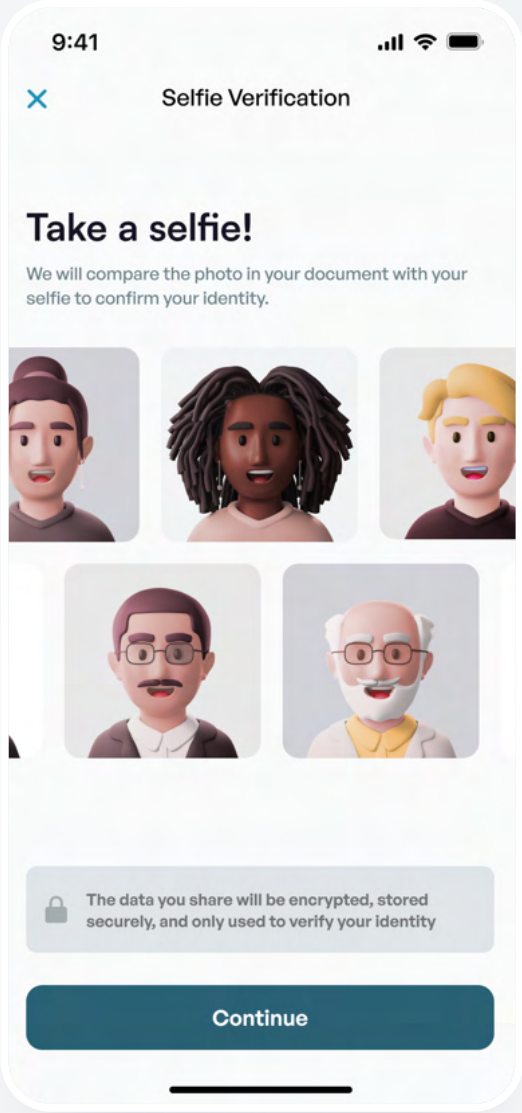
You want to open a new bank account or make a large online payment. First of all, your identity must be verified by the bank. Would you be satisfied if the bank merely required you to send a selfie as identification? You wouldn't, would you? A single, simple selfie isn't a safe way of identification. Images of our faces can be easily found and used on the internet.

How can we avoid this situation?

To fight this, we'll employ a more advanced facial verification method created by Onfido. We want to avoid fraud while keeping a good user experience for all users. Instead of static selfies, we ask individuals to record live videos with this new updated method. To prove they are real humans, they need to record themselves moving their heads and speaking aloud.

Before that happens, though, remember what we spoke about guiding our users through the onboarding process visually? Yep, it's time for another splash screen.

Let's prepare them for the next step and invite them with an excellent bold title, a fun 3d illustration, and a security badge to ensure that the data that's about to be shared is encrypted and securely stored. It's only used for verification purposes.



1.6.5 - Loading screen

Video selfie



1.6.7a - Live camera preview

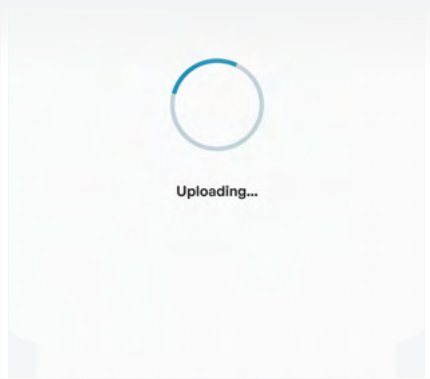
We'll be taking a video selfie now so the system can compare it with the user ID we provided in the previous steps. Onfido system looks for distinguishing biological traits, such as face geometry, and directly compares the ID and a picture of the user's face.

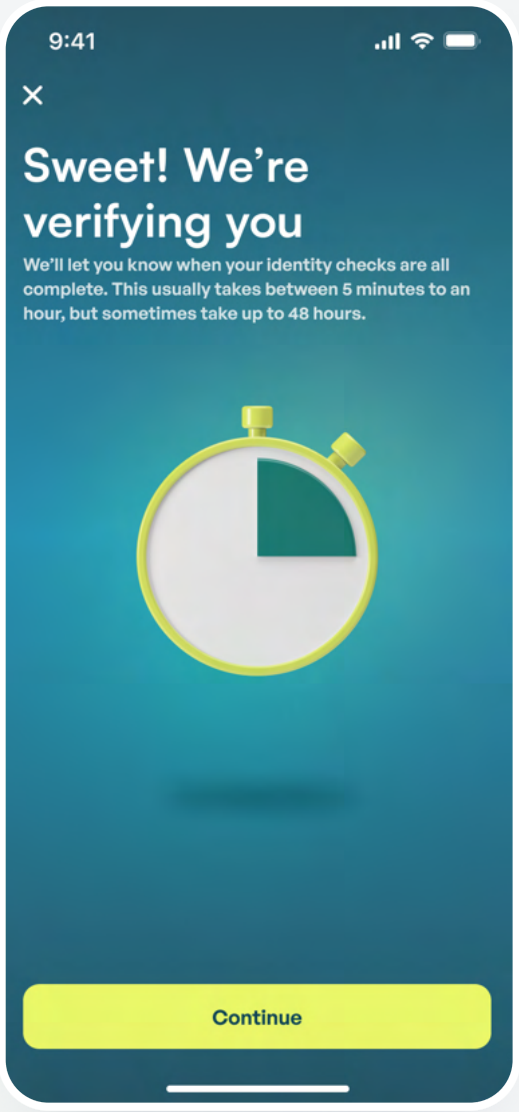
In some cases, it also requires moving the face in different directions and saying a few words to the camera to analyze multiple frames from a user-facing camera to determine that a human is the right person and is, in fact, a natural person.

Uploading - confirmation screen

After the successful verification we'll play a click "Uploading" animation to confirm the photo upload and then we'll take the user to the next screen which is a confirmation screen. Even though we can automate the verification process we might still need some manual checks.

Let them know that the process might take longer than a few minutes, and inform them however long it might take. In our case let's say our system may take up to 48 hours to verify the identity. We'll send them a confirmation as soon as that's done.



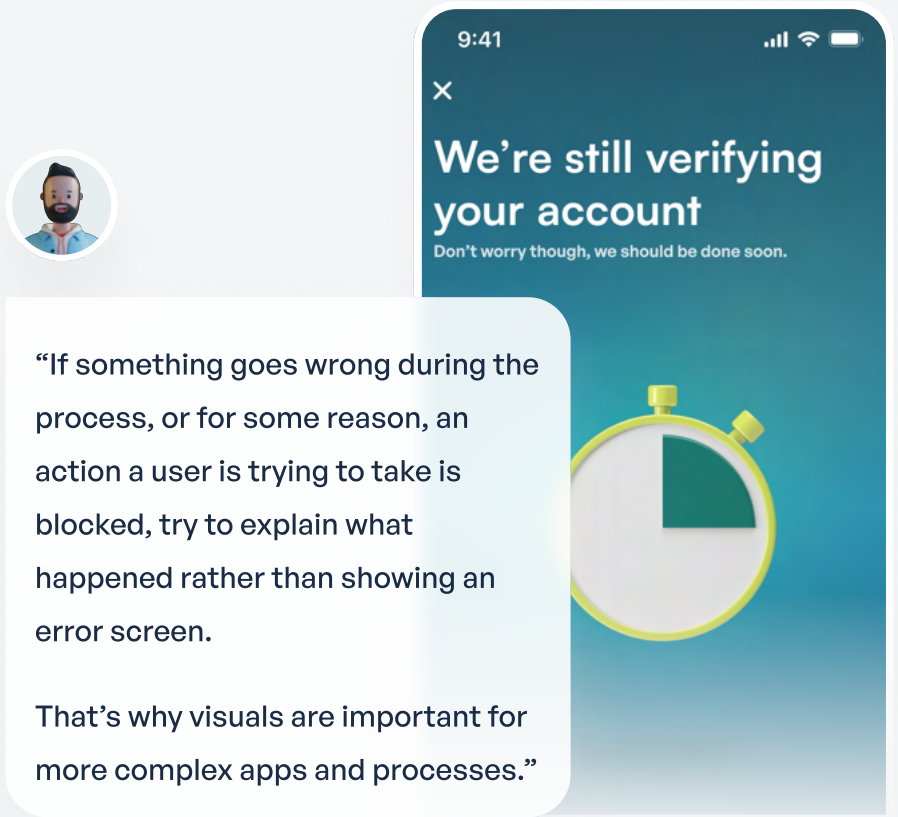


1.6.8a - Verify Identity

Verification in progress

While we're at this point, we might as well design a couple of screens for different states of the verification process.

If we're not verified, we shouldn't be able to send, receive and transfer money. In that case, we should show a splash screen with appropriate information - that we're still verifying your account and that it'll be done soon.



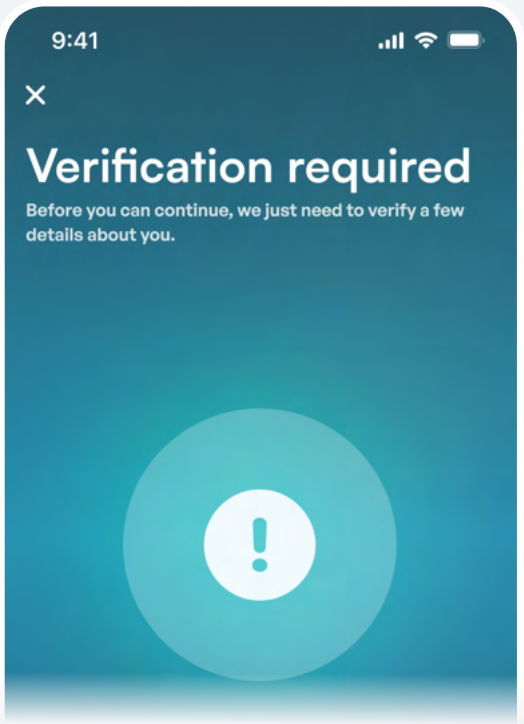
"If something goes wrong during the process, or for some reason, an action a user is trying to take is blocked, try to explain what happened rather than showing an error screen.

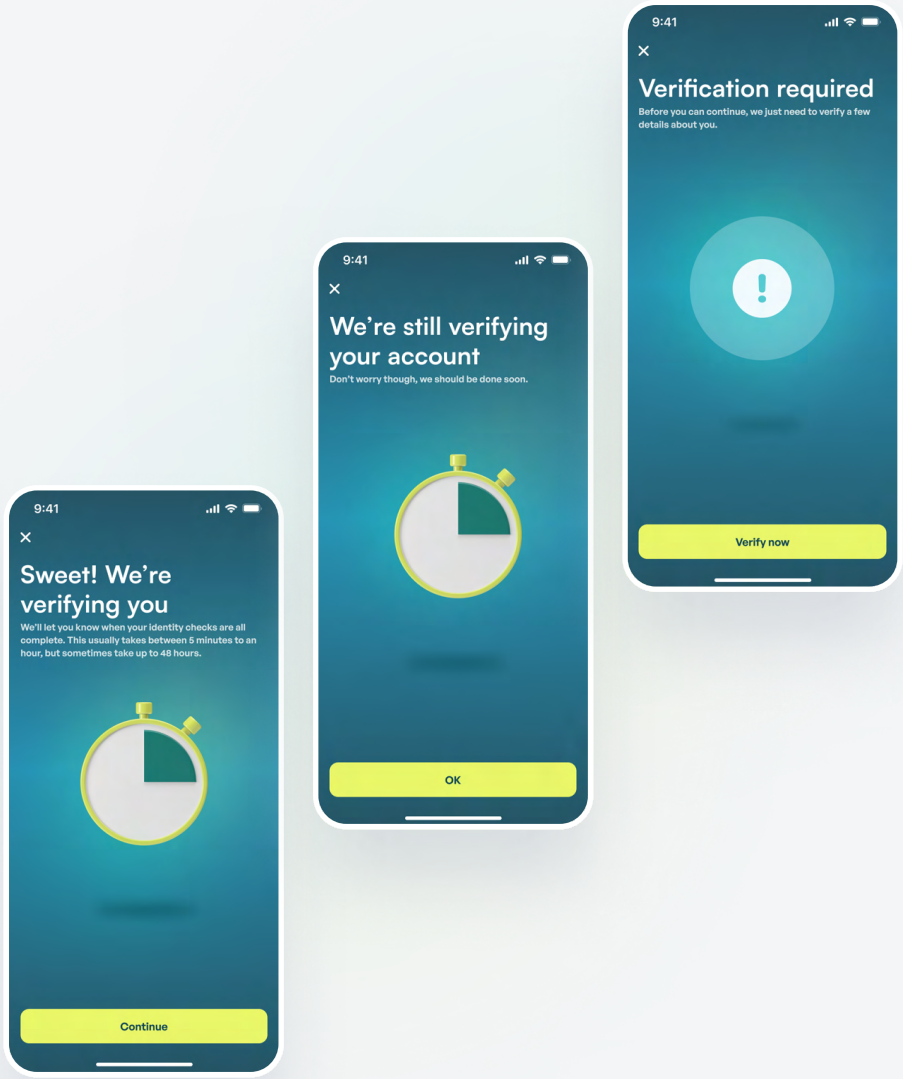
That's why visuals are important for more complex apps and processes."

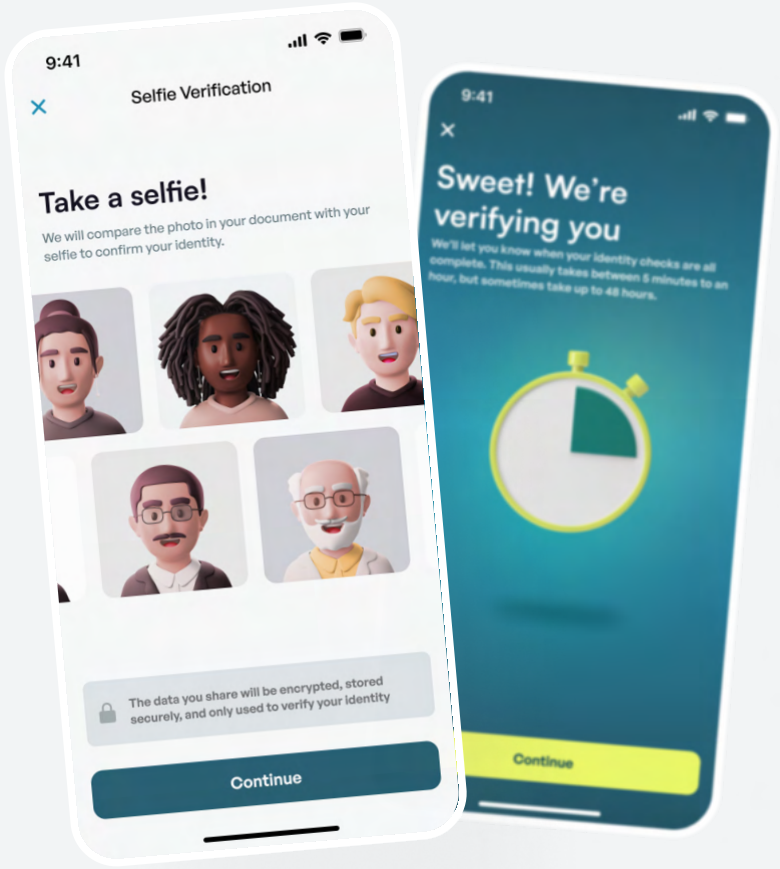
Verification required

If we skipped the verification process in our current journey and wanted to open up a new account, order a card, or request money, we would see the following screen - Verification required.

Unfortunately, this is a step everyone needs to complete, but we allow them to do it at their own pace, and we're not trying to lock anyone in the process.

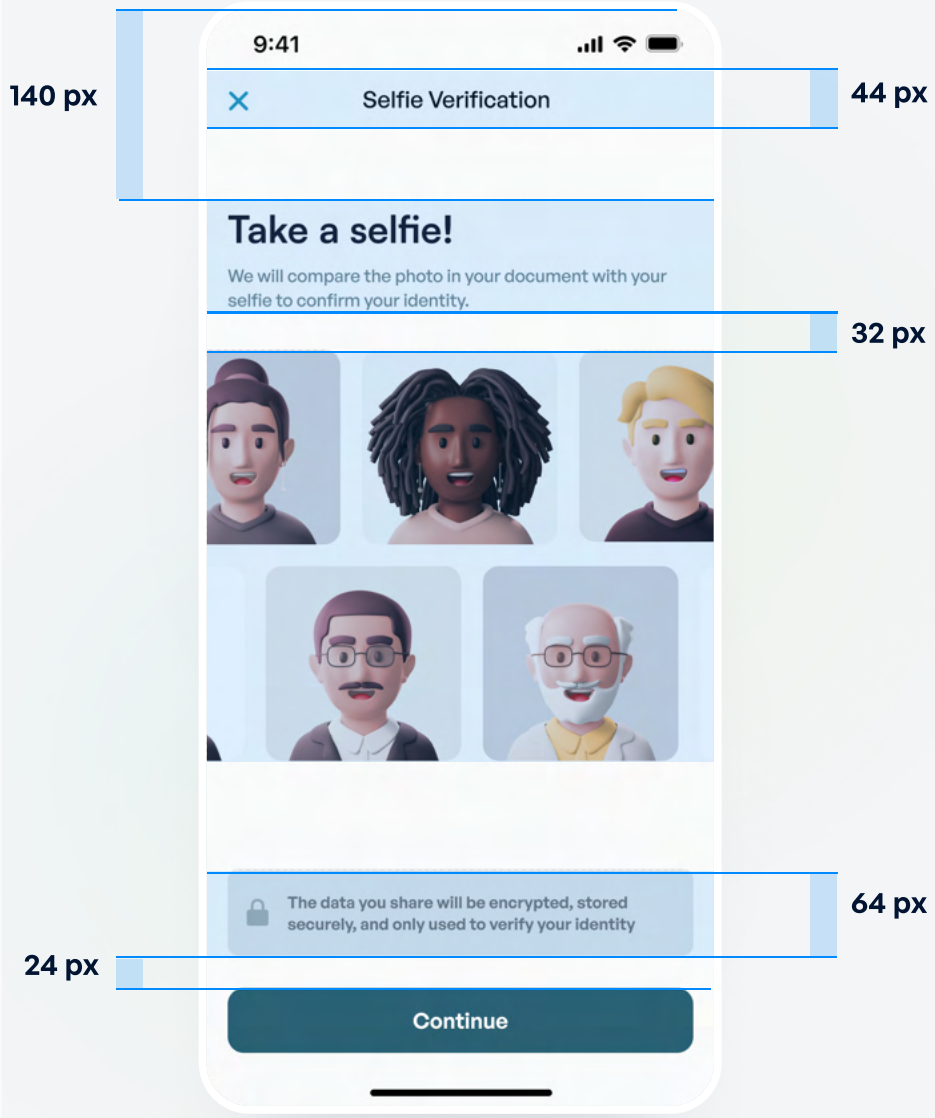






Selfie verification

THE STRUCTURE



1.6.6 - Take a selfie

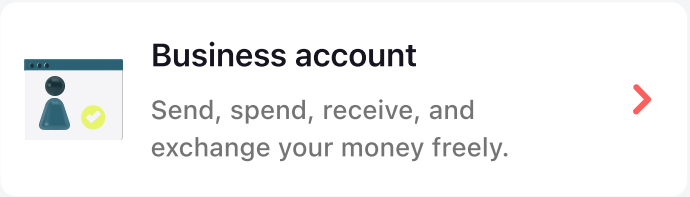
Selfie - the structure

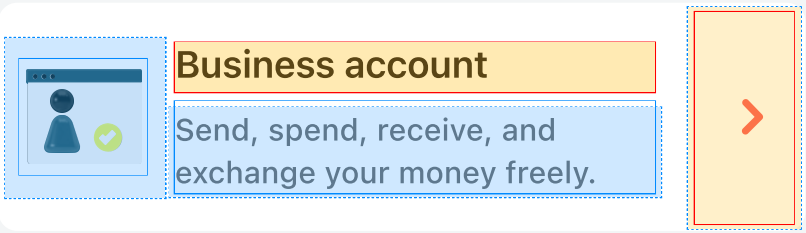
Let's recreate our Selfie screen now. What patterns can you recognize already? Let me help you.

Verify your identity looks familiar, doesn't it? Copy our top centered label from the "Verify your identity - ID Camera" and invert the colors. We have our top section ready.

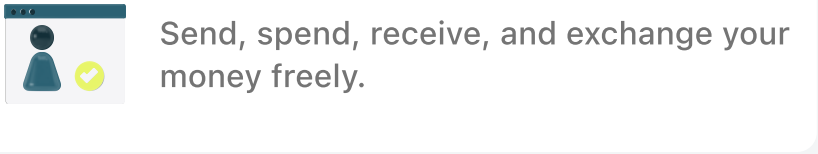
Next, let's use our Title Container but this time, move it around 56 px below the "Verify your identity" title. Update the text. Paste the image from our Assets page, center it in the frame, and position 32 px from the title.

For the last part, create a container box with a lock as an image and a description text on the right. Do you recognize this pattern already? Let's look for a box we can use to replicate it real quick. I found one I think could work.

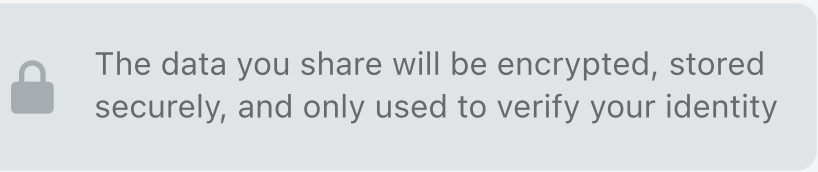




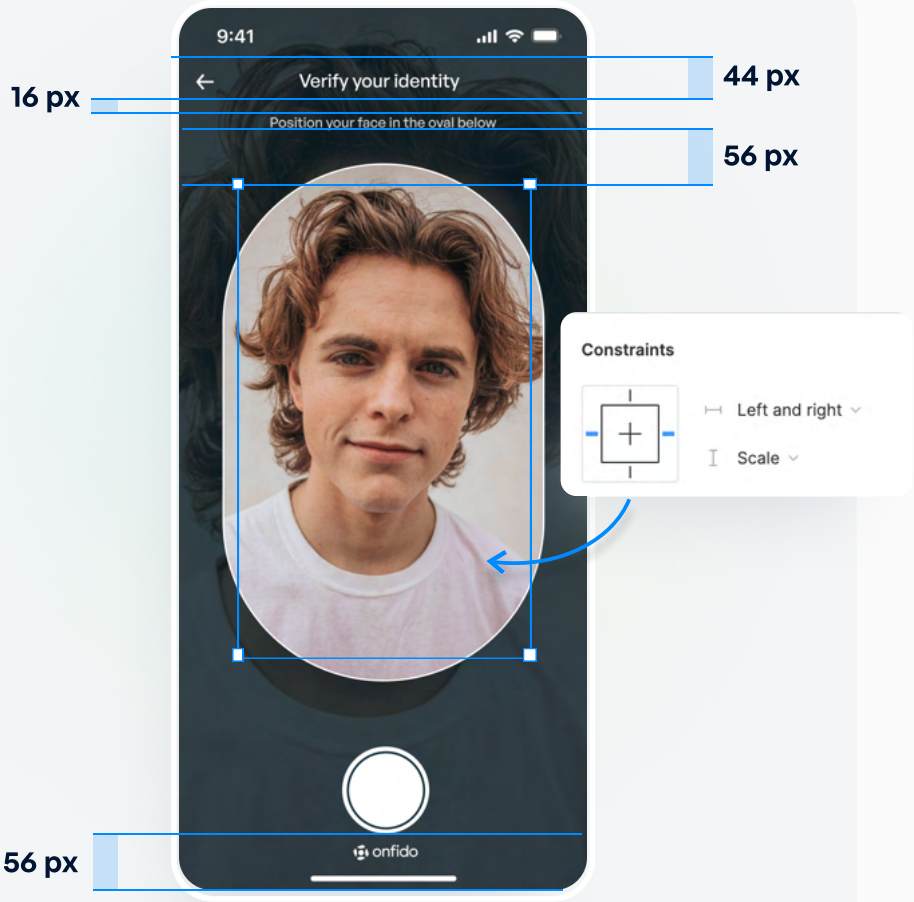
Delete the “Title layer” and the “Chevron container”. We should have something looking like this.



Now replace the image with our lock (you can copy it directly from the left reference frame), and change the color of our background box.



Resize the box to fit our frame - it should be 343 x 64px



1.6.7a - Live camera preview

For this one, we could just replicate what we had on the “Verify your identity - ID Camera” screen. The only thing we need to change is the center image. So draw a rectangle and round the corners to a high value - 500px - we want to be sure our image will be rounded when resized on higher resolutions. Set the constraints of the image to “Left and Right” and “Scale.” **Why “Scale” and not “Center”?**

Because setting it to “Scale” will retain the same ratio as we have on the smaller resolution. Setting it to Center, Bottom, or Top would stretch out our rectangle, and we don’t want that. See below for what it would look like.

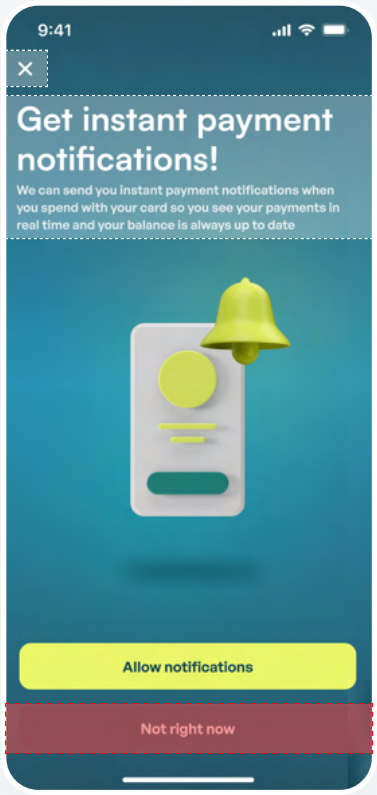


“Scale”

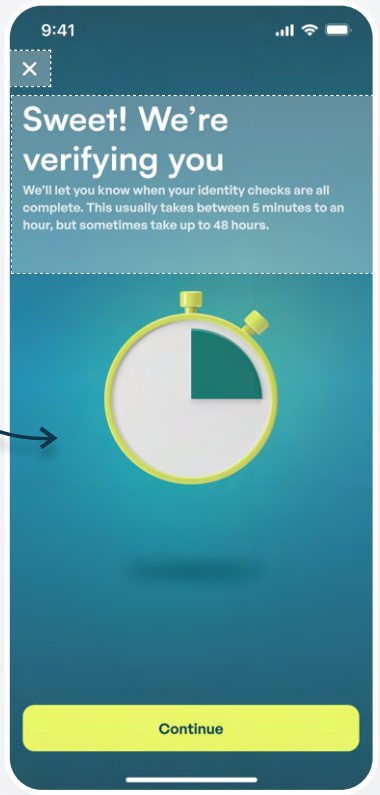


“Center”

Do you see the difference now? Both screens are set to iPhone 13 Pro Max resolution.



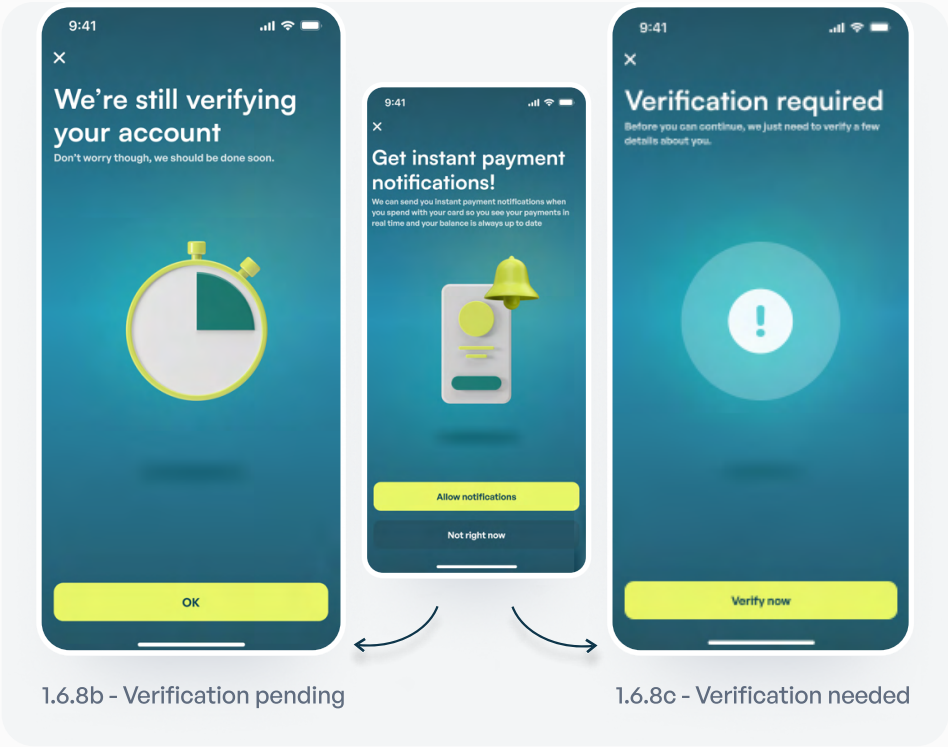
1.5.1a - Notifications on 1-2



1.6.8a - Verify Identity

We’re verifying your identity now

Here we can directly replicate the “Get instant payment notifications” screen. Update the copy, replace the image, and delete the Secondary button. Replace the text on the button. That was fast wasn’t it? That’s how you should be working as a designer. Recognize design patterns and use components to speed up your workflow.



1.6.8b - Verification pending

1.6.8c - Verification needed

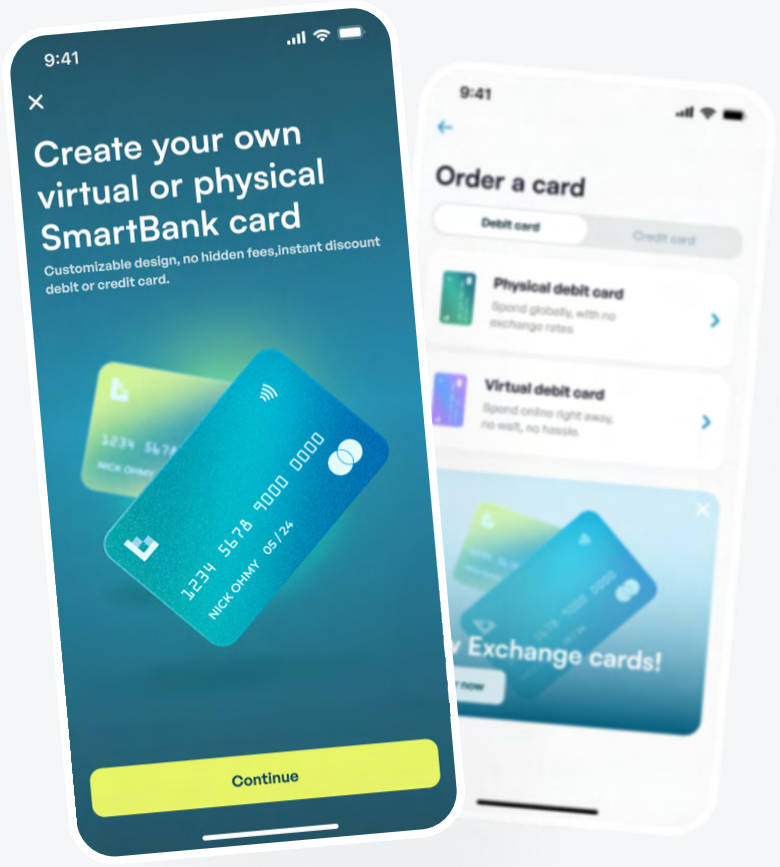
Verification states

The same thing goes for the two above. You can simply copy your entire "Welcome to SmartBank" screen, update the images, replace the text, invert the colors, and update your button style. You will only need to add a "Close" button in the left corner to cancel the action on one screen. Look for one on the previous screens and paste it here.

We're done with the Verification flow - we're ready to finally create our cards and start using the app!

Chapter recap

- Instead of static selfies, we ask individuals to record live videos with the help of the Onfido verification system.
- You should design multiple screens for different states of the verification process.
- If something goes wrong during the process, or for some reason, an action a user is trying to take is blocked, try to explain what happened rather than showing an error screen.
- Setting a shape to "Scale" will retain the same ratio as we have on smaller resolutions.
- Setting a shape to Center, Bottom, or Top would stretch it out instead of maintaining the same properties.
- You should always look for similar design patterns and use components to speed up your workflow.



Order a card

THE INTRO

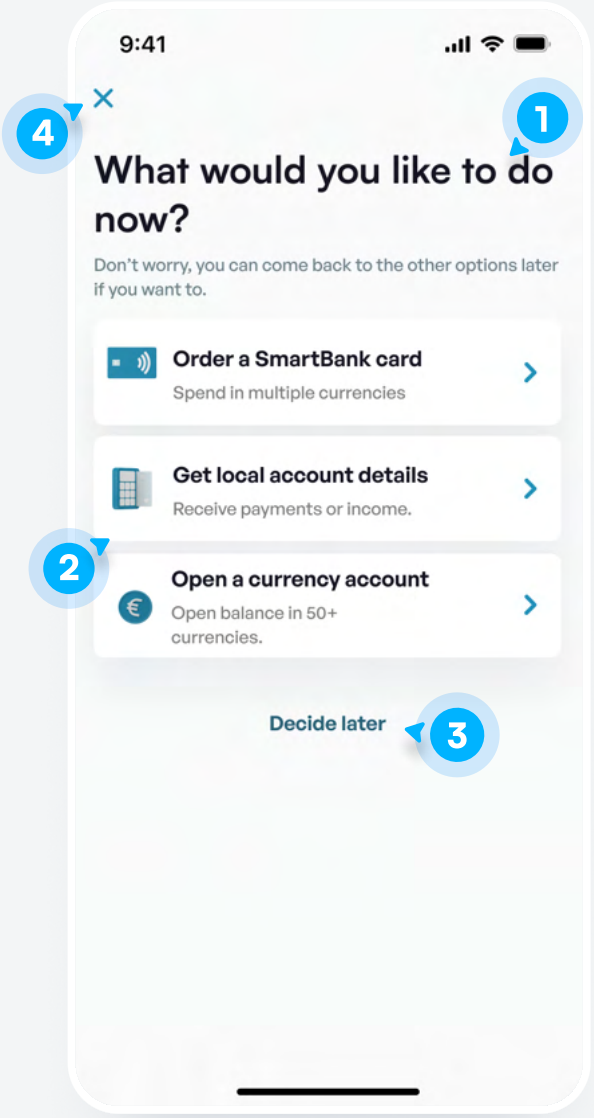
Cards - the theory

The next step is a personalization wizard; now that we have all the details we need to verify the users' identity, we can click out of the wizard and go straight into the app or choose one of the following actions. Smartbank will help us set everything up. What's the first thing we need to start paying for our commodities? A credit card. But we don't have an account yet, so how can we order a card? SmartBank will automatically open up an account in your name and link your credit card to that account if you choose the "SmartBank card" option.

First things first, though, let's break down our screen.

The building blocks of a wizard screen

1. Title to explain the actionable steps
2. A list of actions we can take to make the setup much easier and more streamlined
3. A button to skip the personalization process
4. A close button to return to the home screen



1.7.1 - What would you like to do?

Create your virtual or physical card splash

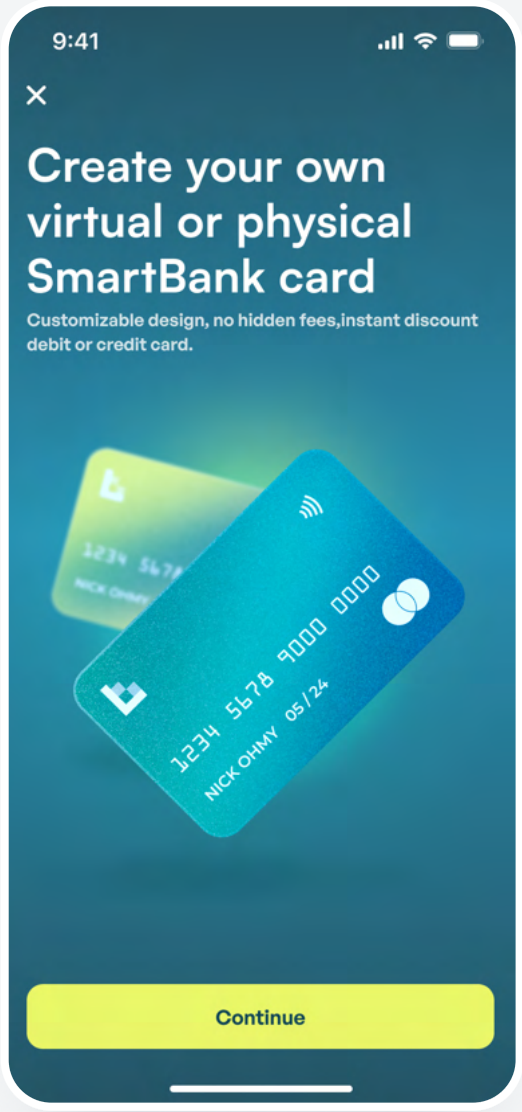
Okay, we clicked on the first option, and now we're taken into a "Create your card" flow. As we've established with our previous screens, we guide our users visually through the whole onboarding experience. We want them to identify our brand with a sense of security, clarity, and a visual style specific to our brand and easily recognizable.

Having said that, we decided we wanted to create a new card. In this case, we want to follow up with an intro screen that explains what's about to happen.

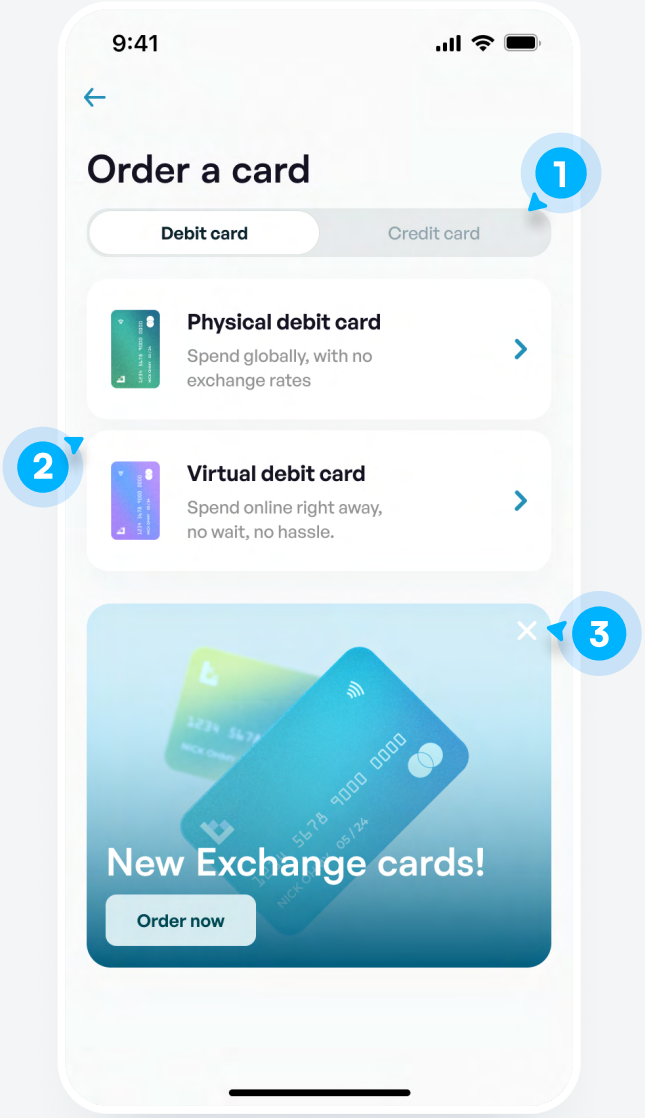
Looking at our intro, I can already see that we are able to create virtual and physical cards. Cards are customizable, so you're free to choose whatever design you like - a nice touch to keep it personal. There are no hidden fees, and everything is clear and transparent, showing that we have nothing to hide.

Lastly, we get instant access to the virtual cards, so there's no waiting time. Sweet.

All looks great; I'm ready to proceed; let's continue to another step.



1.7.2 - Create your SmartBank card

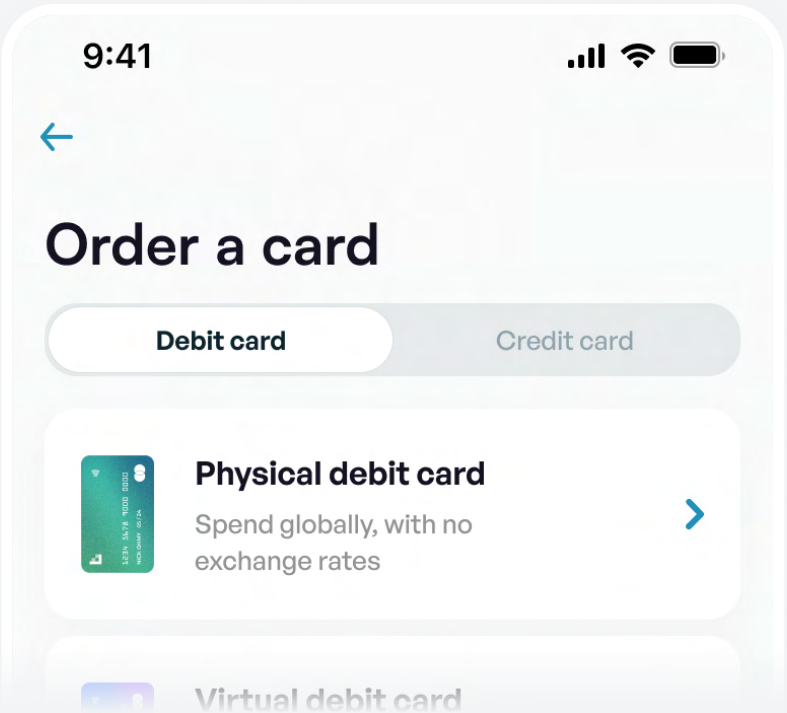


1.7.3a - Order a card

1. Type of card

The following switcher allows us to select the type of card we want to use. Aside from the physical and virtual cards, we are free to decide between debit or credit.

This way we make sure to choose the most convenient option for us.



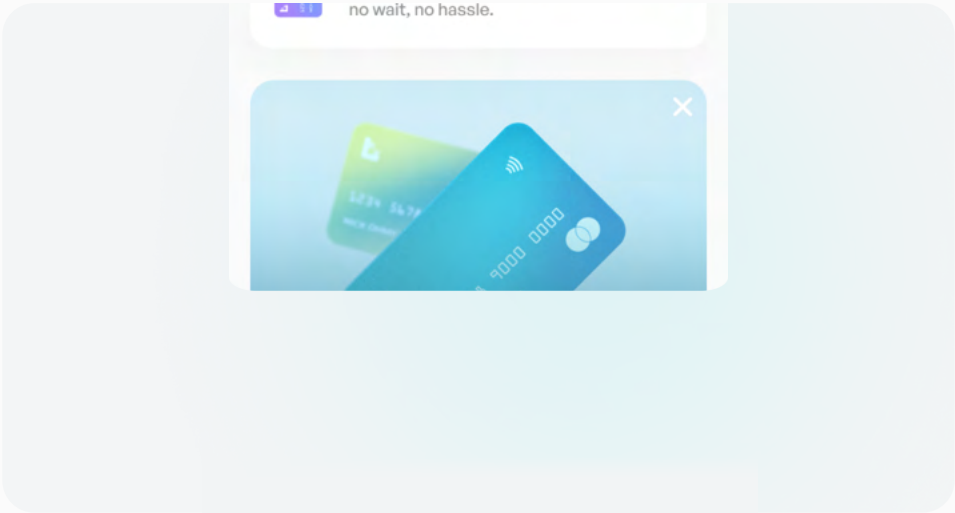
2. Physical vs. virtual cards

Virtual cards are used only for online purchases, while physical cards can be used in person and online. That might be the only advantage physical cards have over virtual ones.

The only disadvantage is that you have to wait for the delivery and with virtual you're ready to go right away.

3. Promotional banner

We want to showcase our marketing material in relevant places. We can show a promotional banner right before our users start a new flow by ordering a new card using this opportunity. However, this is optional, and we don't want to make it too intrusive, so let's offer a close button to close it quickly or ignore it while moving forward to the next step.



Customize your card



1.7.3b -Your virtual credit card

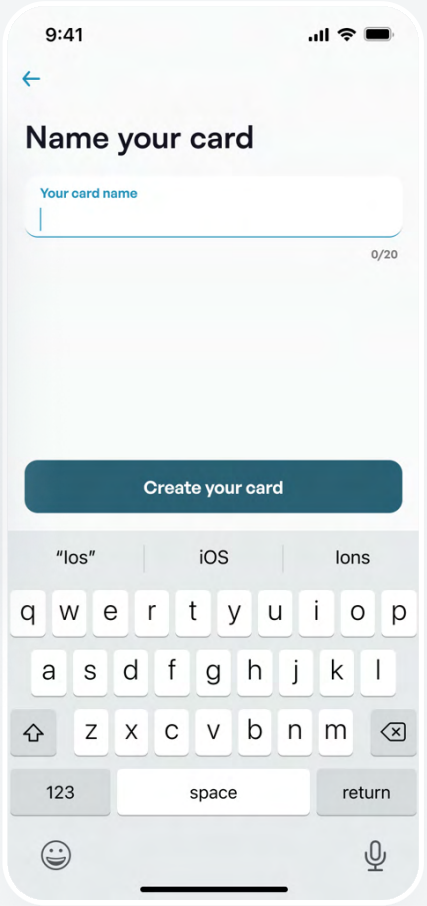
Any authorized cardholder can get a personalized credit card. We can choose a design upon card activation. It's this little personalization gimmick that might just add to the overall experience. Choose from several designs so your credit card matches your unique style.

Name your card

Apart from offering custom card design, we'll also add the ability to name our cards accordingly. Whether it's just a "Shopping card," "Utility card," "Wife's card," or whatever you want to use it for, name it as you see it and keep a better track of your spending.

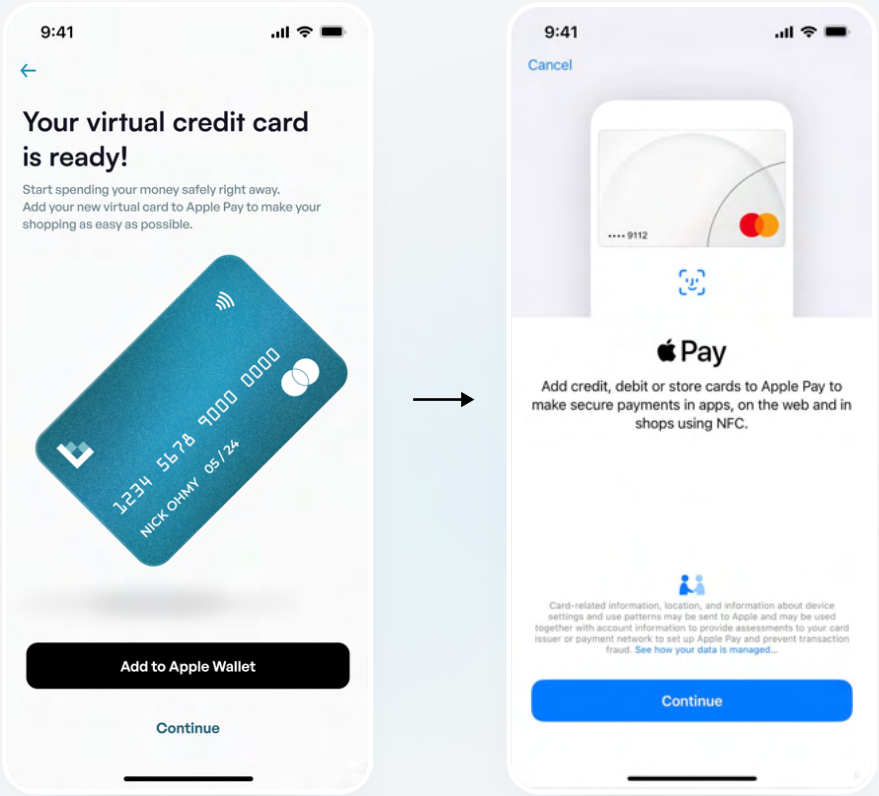
"While all these could be randomized or defaulted to one option, offering some kind of personalization might help increase your brand loyalty by appearing more user-focused rather than focused on yourself."

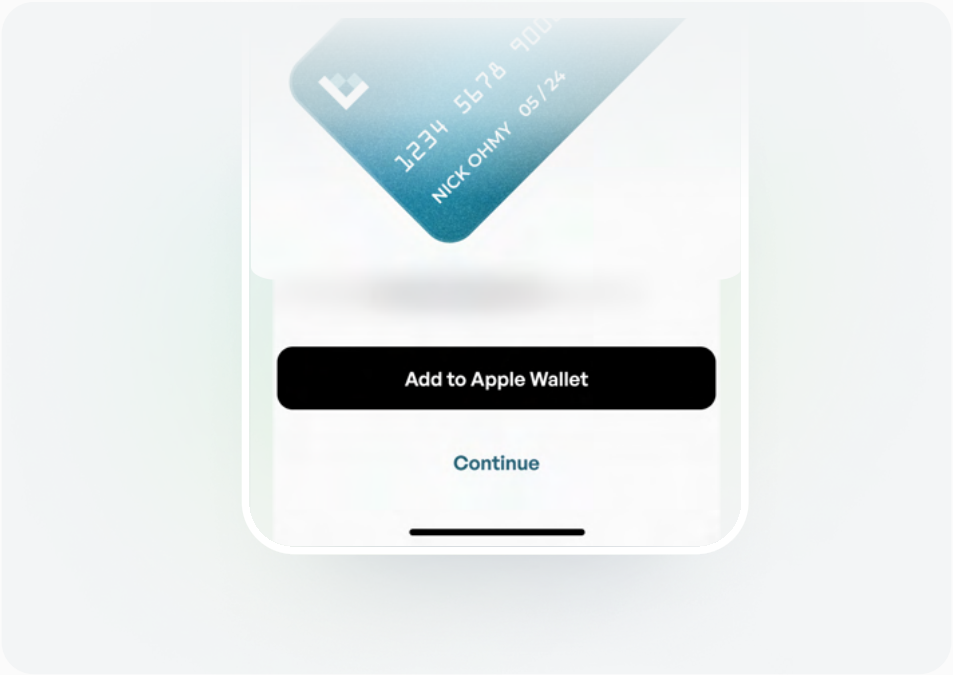




Virtual card is ready

As the last step of our card setup, let's add an option to add our new card to the Apple Wallet to make purchases even more effortless. It will automatically copy your new card details so you won't have to type them yourself. Once you click on the "Add to Apple Wallet" button, you'll open a new flow within the Apple Wallet app, and when that's done, you'll return to SmartBank.

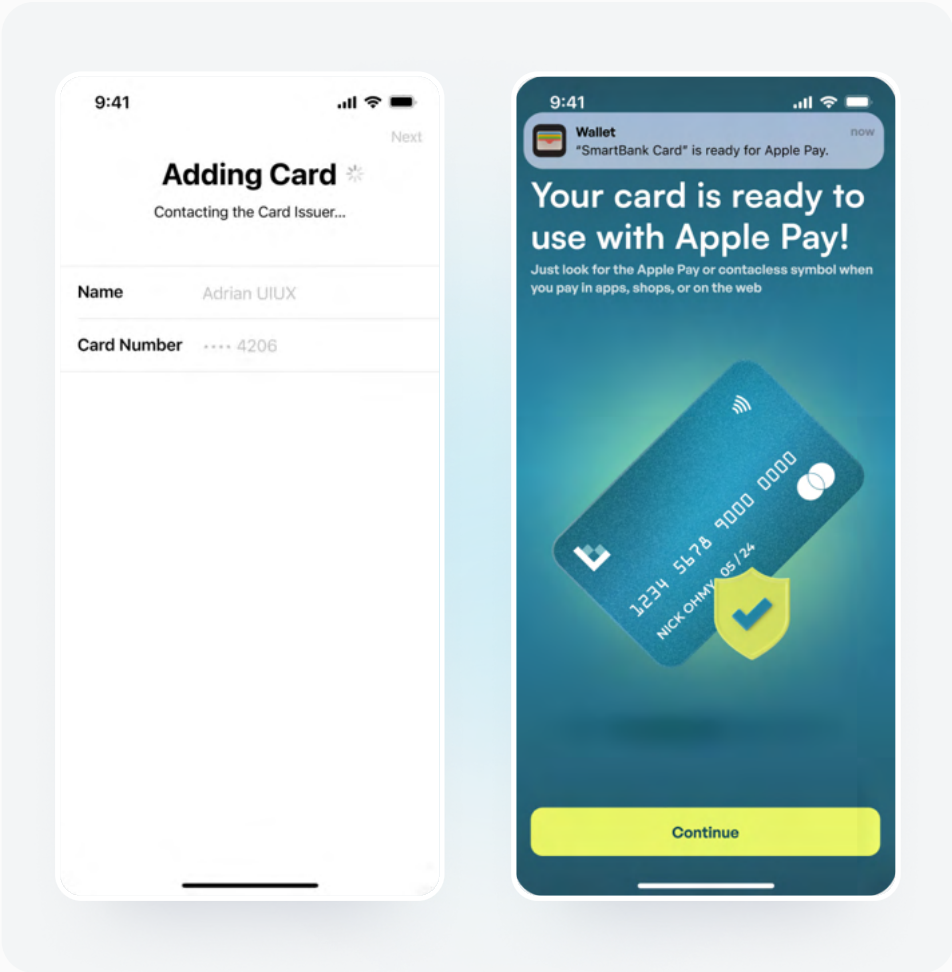




Why Apple Wallet?

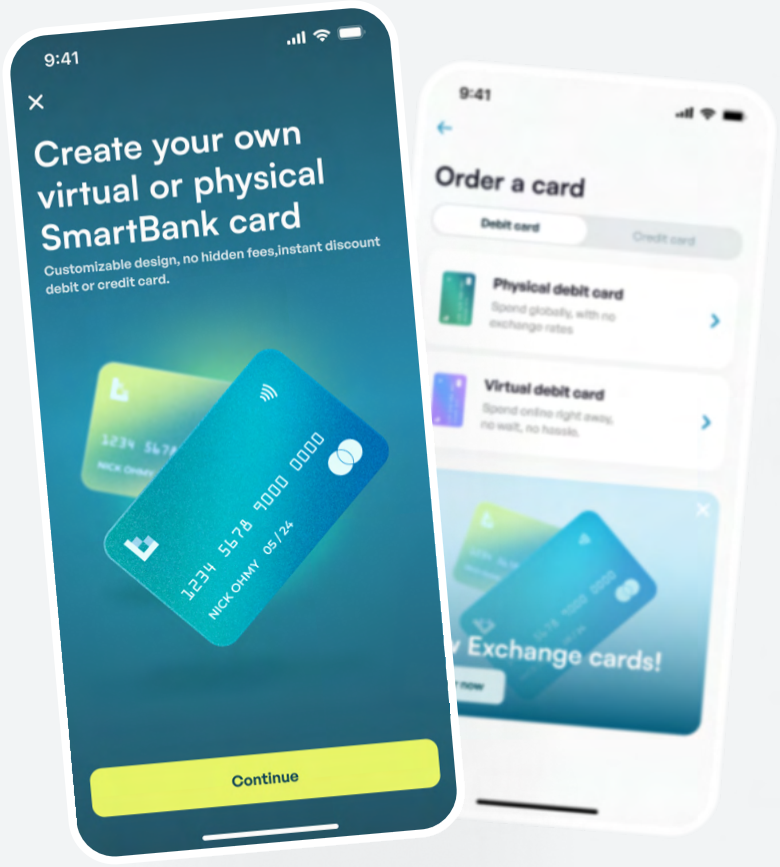
Apple Pay is more convenient and safer than using a physical credit, debit, or prepaid card. Face ID, Touch ID, or passcode are required for purchases on your iPhone, Apple Watch, Mac, or iPad. Your card number and identity aren't shared with merchants, and your actual card numbers aren't stored on your device or on Apple servers.

Even though Apple Wallet is installed on every iPhone device, we should still give users an option to continue the process without linking any new cards with Apple Pay.



Apple Pay integration

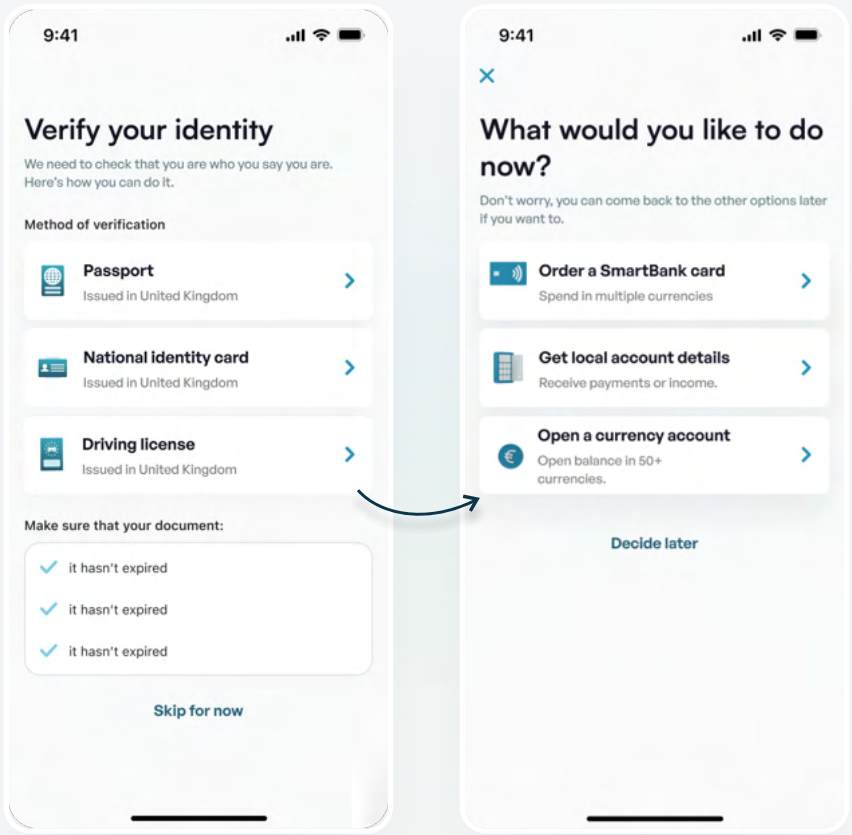
This flow is pretty much the same for all Apple Pay card integrations so we're just going to design our last tutorial screen with the confirmation message and a push notification from Apple Wallet confirming the successful card integration.



Order a card

THE STRUCTURE

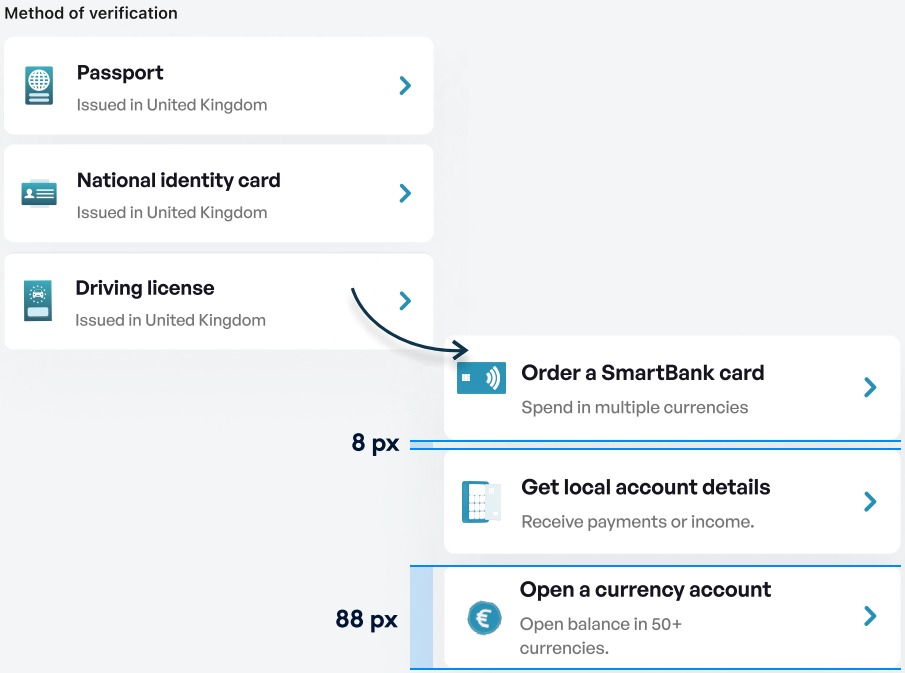
Personal details - the structure



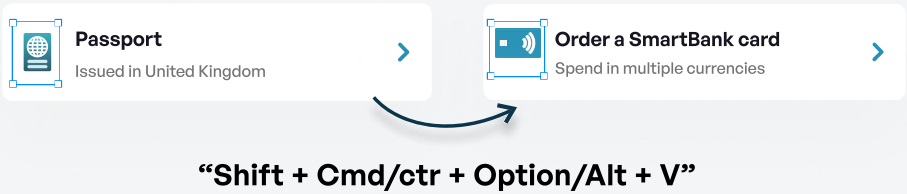
1.7.1 - What would you like to do?

1.6.2 - Verify identity type

Before moving on to design our screen, let's see how can we make our task easier and look for a similar design pattern. I think that "Verify your identity" is just what we need.

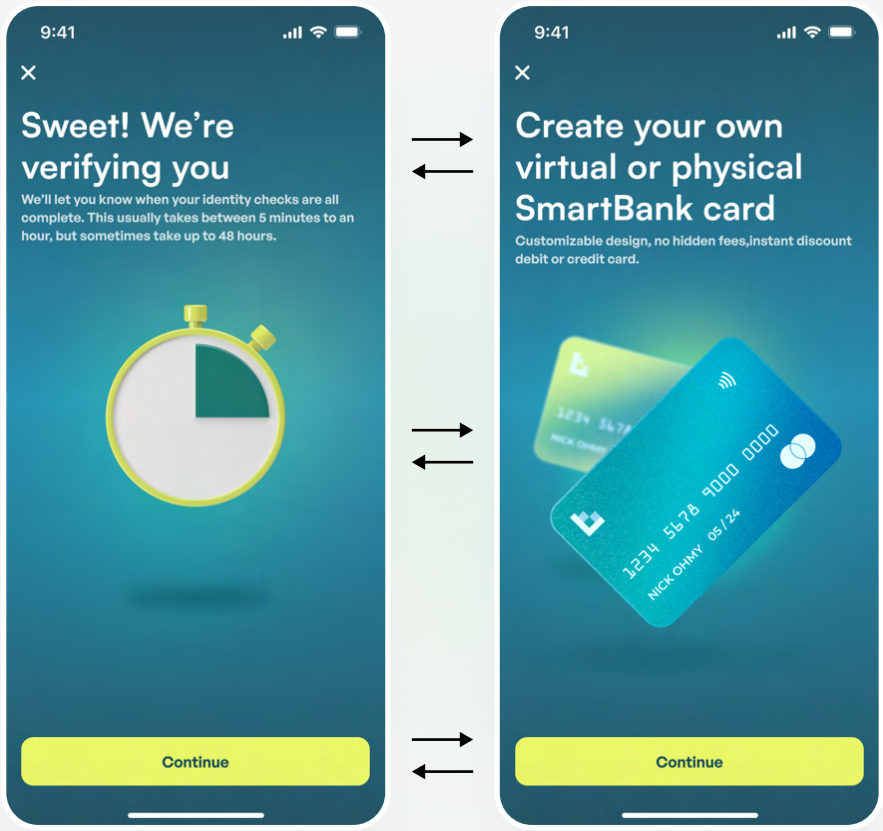


We can copy-paste the same component on our new screen and start adjusting the design. Replace the images. **Remember the shortcut?** Copy the image from our reference frame and select the old one in our container. Click **"Shift + Cmd/ctr + Option/Alt + V"** to replace it.



Create your own card

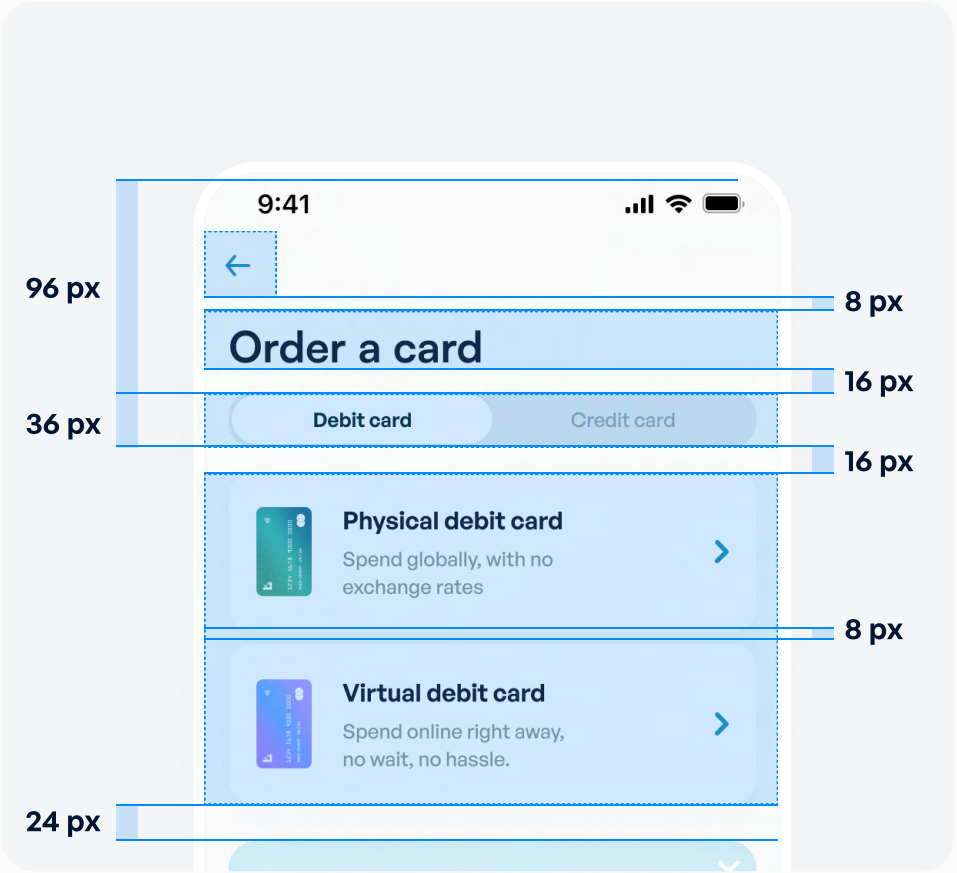
We can replicate the “We’re verifying your identity” screen with our next intro screen. Update the copy, replace the image, leave the rest as it is, and move on to the next one!



1.6.8a - Verify Identity

1.7.2 - Create your SmartBank card

Order a card



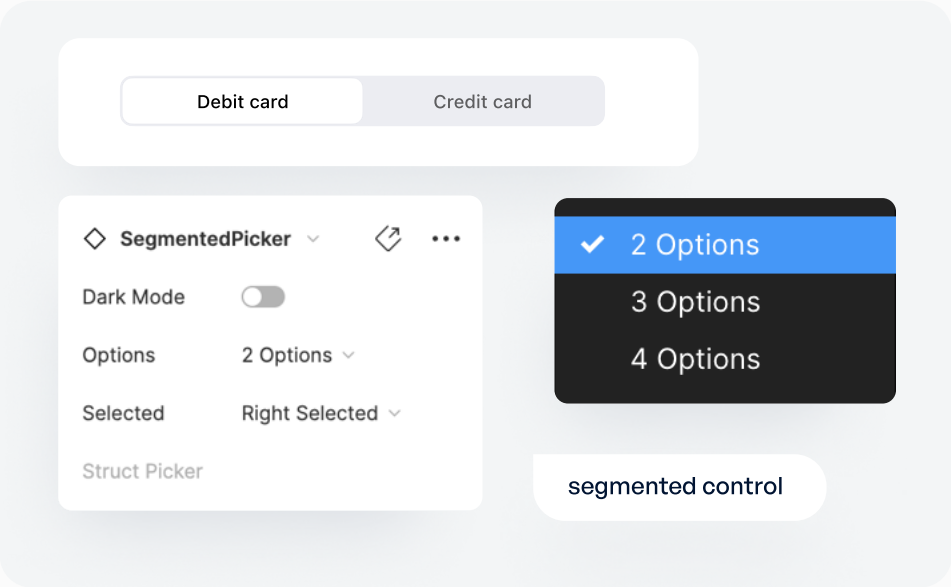
Let’s look at the screen and analyze it. What do we need to replicate it? First off, we’ll need a title container - that’s easy.

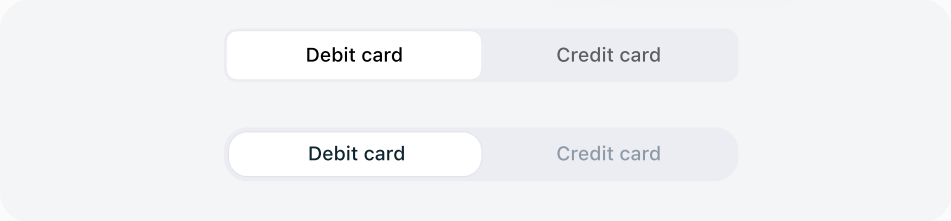
Secondly, we’ll need an iOS segmented picker - we’ll cover it on the next page. And lastly, we’ll need a card container to create our card boxes. We’ve successfully used them for a few screens before, so it shouldn’t be a problem now.

iOS Segmented Picker

We can copy our segmented picker directly from the iOS 15 UI Kit. Make sure it's 343px wide and 36px tall and position it 16px from the sides, and 24 px from our title container.

If you ever need to add more options to the switcher - select it and look at the right-hand-side panel to find the "Options" property. Click on it, and you'll see several pre-made variants. Adjust the style by changing the background color, and that's it.

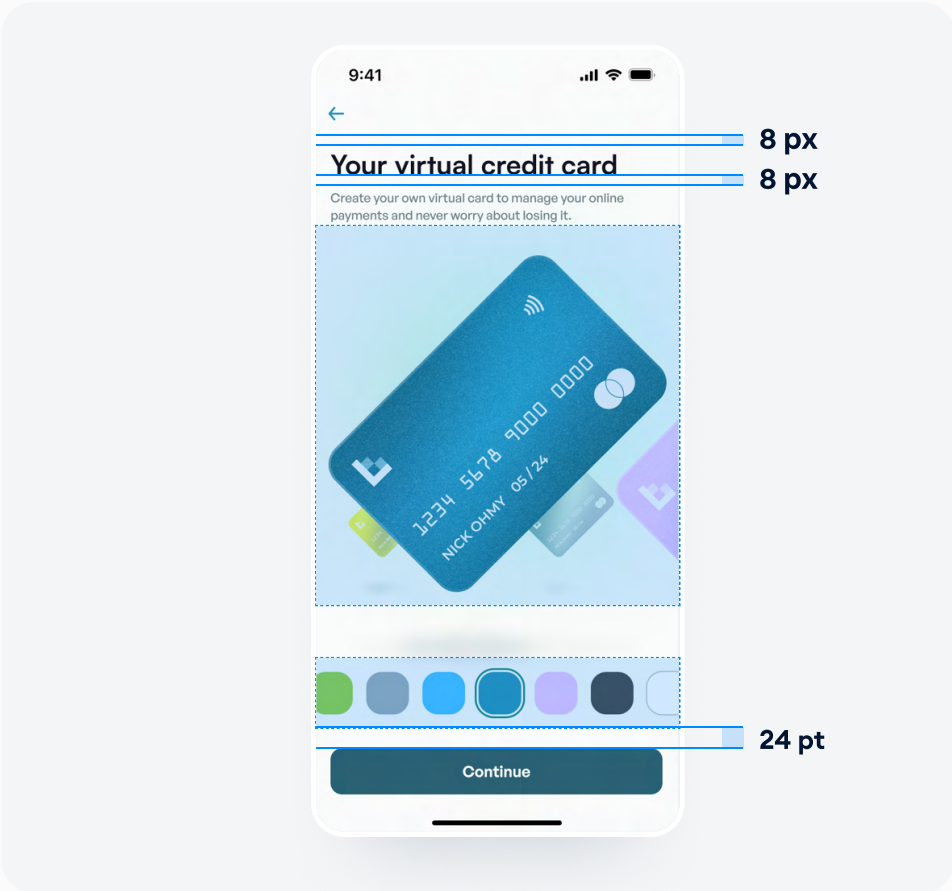




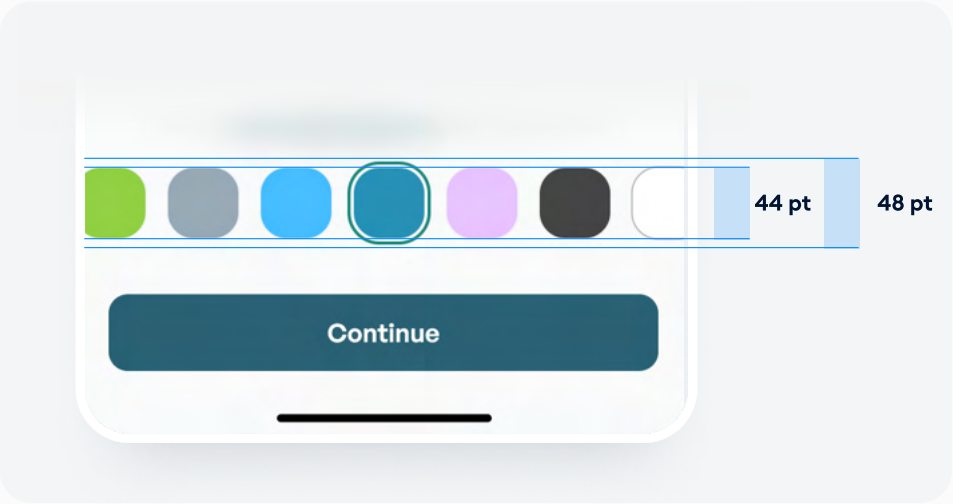
Your virtual credit card

To design the next screen, we'll need:

- A title container
- An image for the middle section
- A container with a number of rounded rectangles to imitate the scrollable color customizer.
- A CTA button



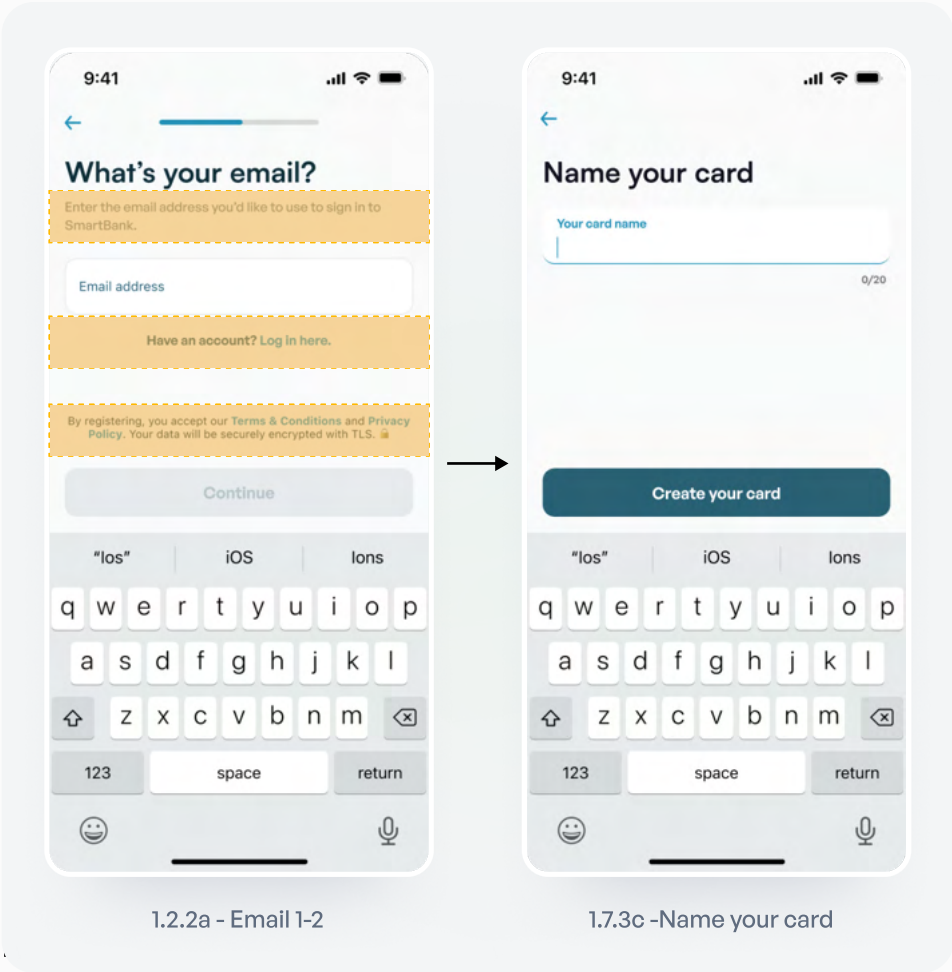
We can copy three of these components directly from one of our previous frames or from our components page. That is the “Title container,” “Image container,” and the “CTA button.” Here’s what we need to do for the color picker.



Auto-layout shortcuts

To quickly create a number of rectangles - add the first one, duplicate it, select them both and apply auto-layout (“Shift + A”). Resize your container to fit the entire width of the frame, select one of the rectangles and start duplicating it “Cmd/Ctrl + D.”

Change the colors and add a second rectangle 48x48 in the middle of your frame, remove the fill, add a border of 2 px and position it directly behind to create our active selection state.

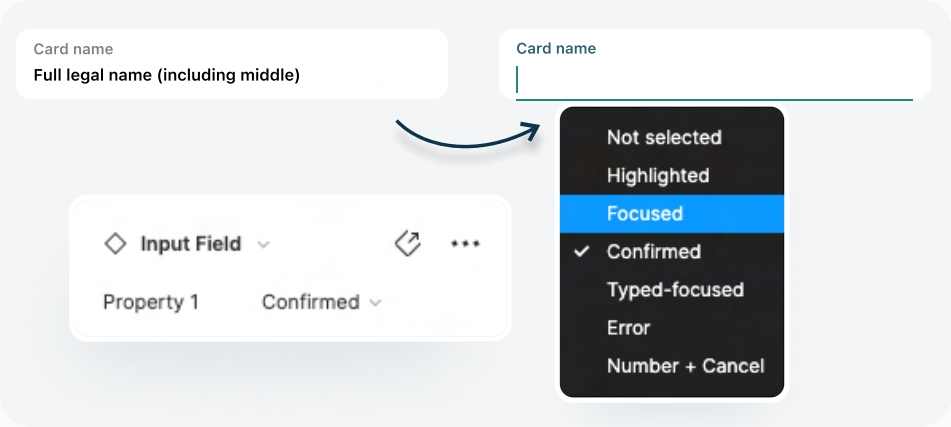


1.2.2a - Email 1-2

1.7.3c -Name your card

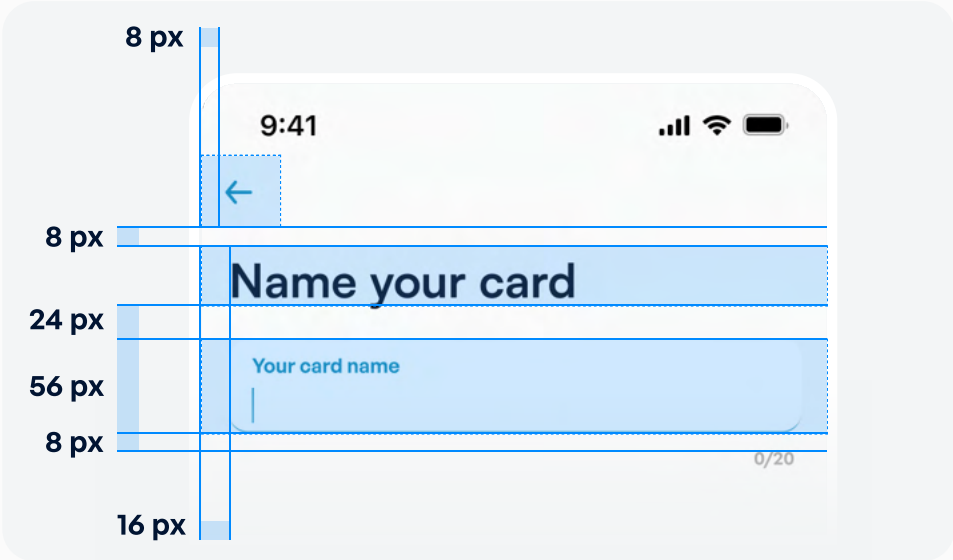
We can simply copy the “What’s your email” frame for the next screen, remove the description, the link button below the input field, and the acceptance text above the button.

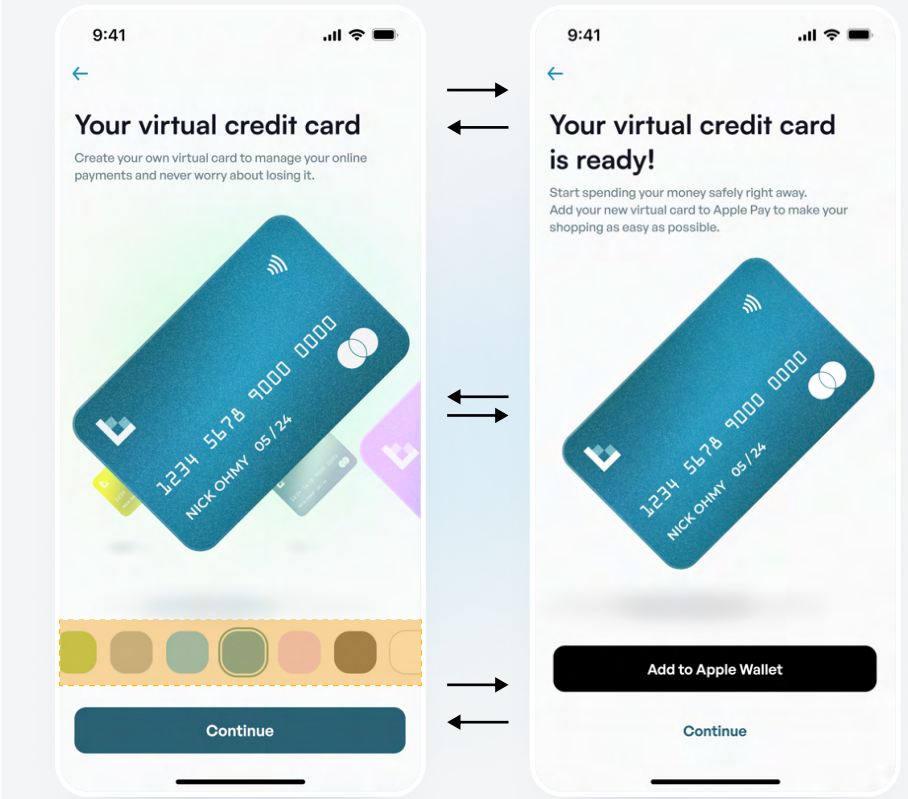
Reposition your elements so that the input field is 24 px from the title container. Update the text and change the input field from “Confirmed” to “Focused.”



Here's the thing - you're free to use your own padding and margin values, but there are specific patterns you might borrow from other iOS apps, especially the ones designed by Apple.

We used similar values that you can find in Apple Music and App Store. All are built on an 8pt grid.





1.7.3b -Your virtual credit card

1.7.3d -Virtual card is ready

Your virtual card is ready

For our next screen, we need to replace the image, remove the color customizer, and replace our button with a button group from the Components page or one of our previous screens. Change the button styles to Apple Wallet and Primary. Update the text for our title and description and that's it! We're ready to start the "Top up" flow now.

A few words about the UX behind this project

We're halfway through our Design Workbook, and I feel this needs to be repeated. All of the design decisions from the UX standpoint are based on the most popular fintech apps on the market and heuristic evaluations that are not backed by any serious research or data. Given the limited resources, we wanted this ebook to focus more on the design, as you've probably noticed, rather than the UX itself.

We are trying to make the best possible choices and place ourselves in our user's shoes as much as possible, but in a real-world environment, we would test every single approach and make adjustments where needed.

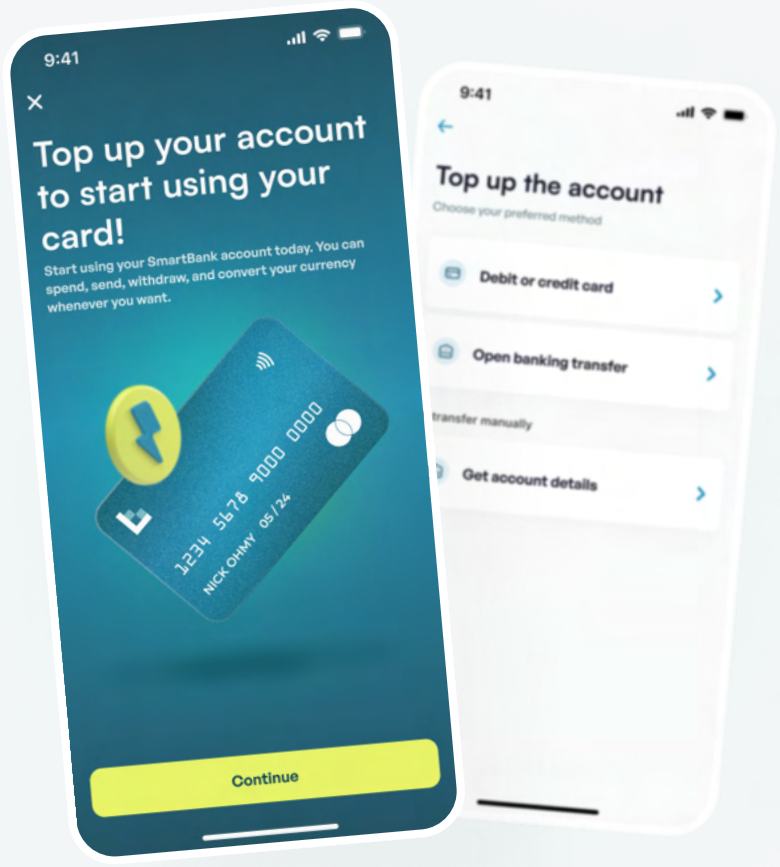
Treat this project as an MVP that's been released to the market for a closed group of testers, and our job is to find the right solutions by analyzing the data collected and improving.

While the UX of this app is probably above average, there are many things we could improve, especially when looking at the account setup, but at the same time, we wanted to design more screens for you to practice with.

As I mentioned many times before, by replicating these screens, you will get better by following the best design practices for iOS design and app design in general while developing a better eye for design and a solid understanding of Figma.

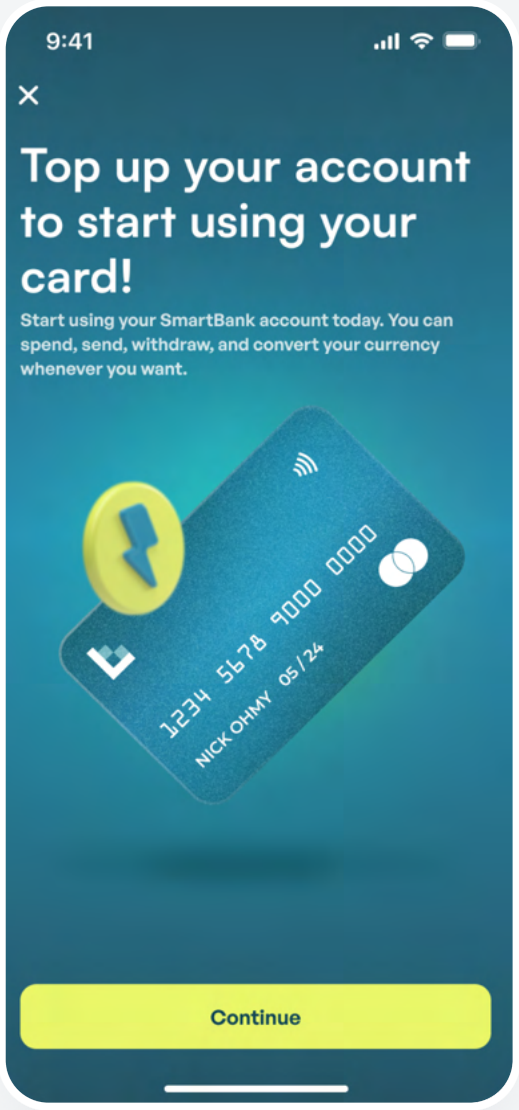
Chapter recap

- As we've established with our previous screens we are guiding our users visually through the whole onboarding experience.
- Apple Pay is more convenient and safer than using a physical credit, debit, or prepaid card.
- Your card number and identity aren't shared with merchants, and your actual card numbers aren't stored on your device or on Apple servers
- If you ever need to add more options to the switcher - select it and look at the right hand-side panel to find the "Options" property.
- Feel free to use your own values for padding and margin. If in doubt, look at other iOS apps, especially the ones designed by Apple.
- You will get better by following the best design practices for iOS design, and app design in general, as well as develop a better understanding of Figma.



Top-up account

THE INTRO



1.7.4 - Top up account

Top up account - the theory

Opening and running an account costs money, and while usually, fintech products are much cheaper in regular fees, they require some kind of deposit to start using their services.

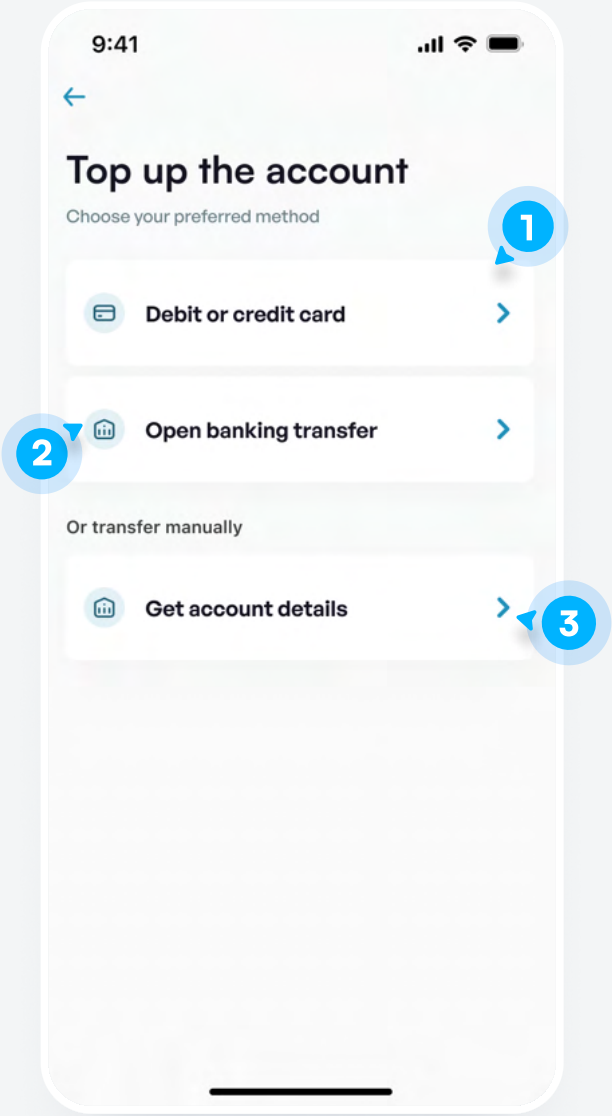
Naturally, we'll need to top up the account if we want to start using the card. This is always optional, so let's leave that close button visible in a place that our users should be used to seeing by now.

We can use our customized card as a focal point for our splash screen visual to enhance the experience.

The basic and most commonly used method to fund the account is by using one of our existing debit or credit cards, an online bank transfer (open banking), or less commonly, using the account details to transfer them directly from one of our existing bank accounts.

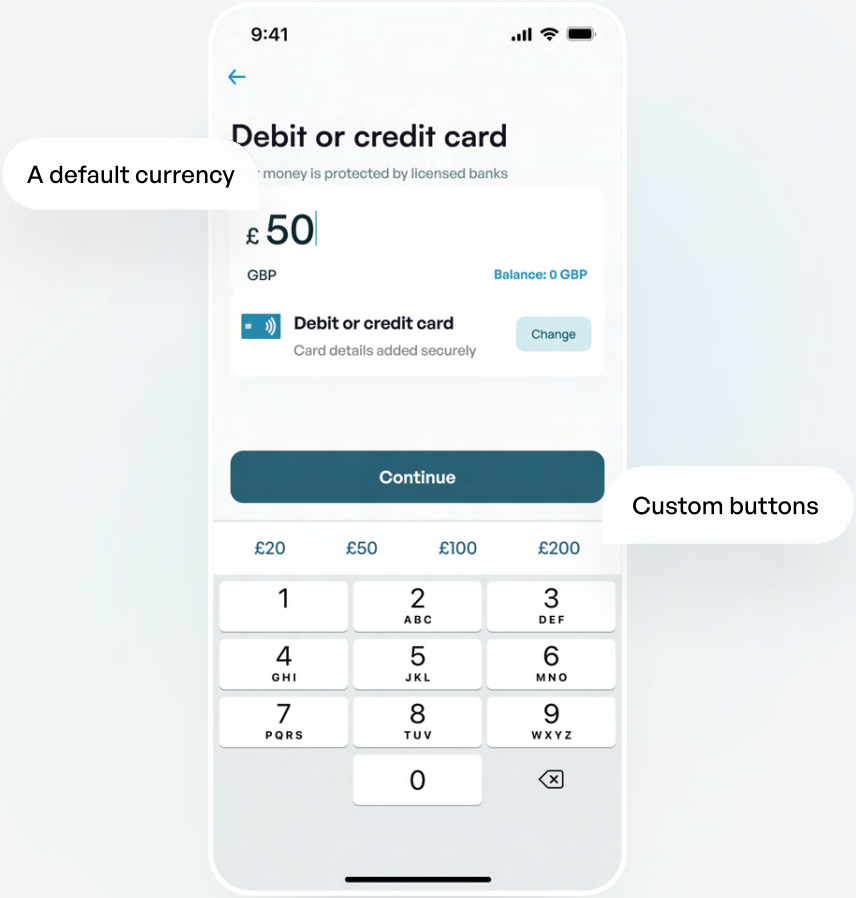
Depending on the chosen top-up method, it may take some time for the top-up to appear in your account. While card top-ups appear in your account right away, the others may take up between 1-3 days.

Let’s see what kind of methods we can use to top-up the account.



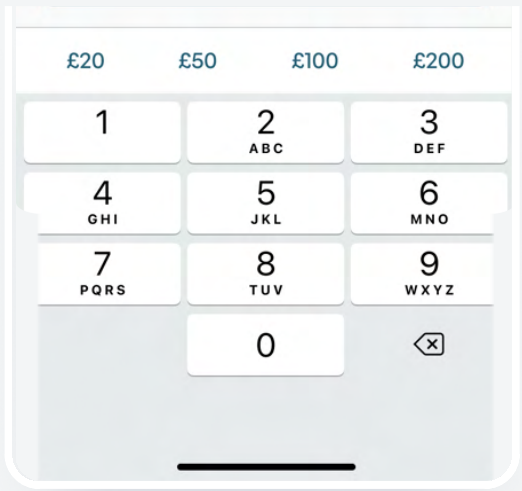
1.7.6a - Top the account

1. Debit or credit card



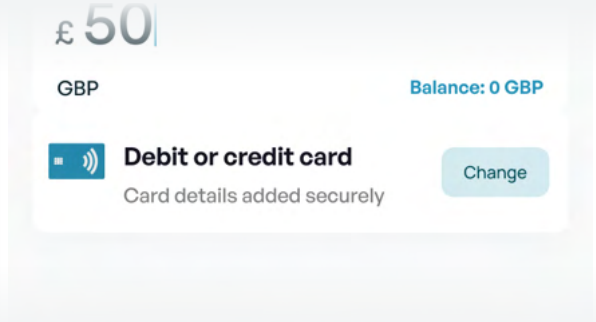
1.7.6b - Debit or credit card

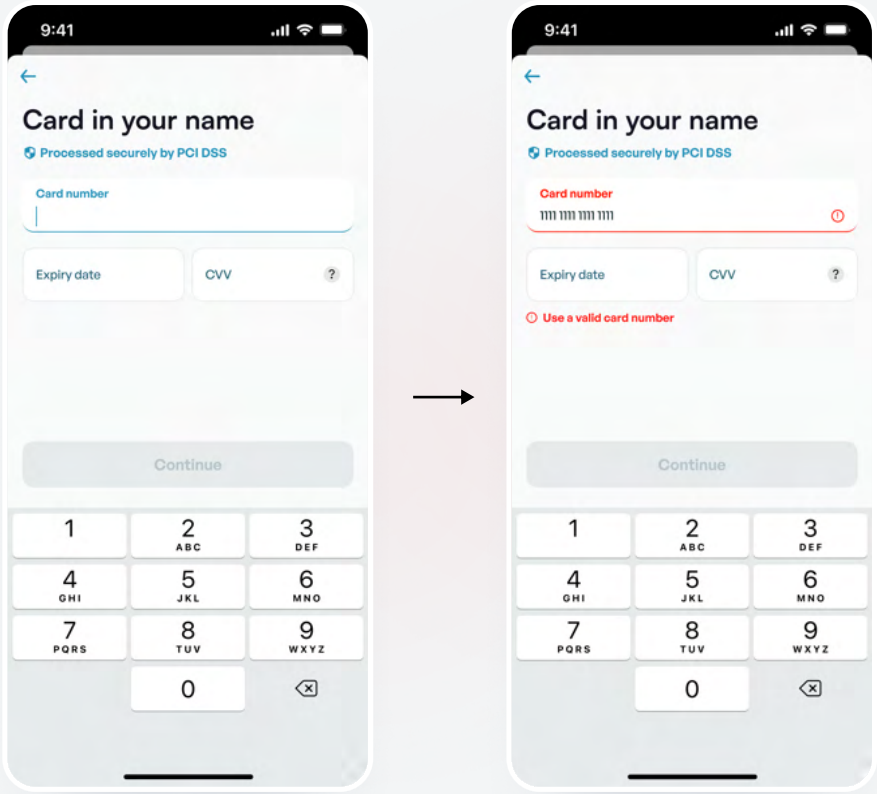
Let's break down our first option. What we'll need is an input field with our default currency that's based on our country of residence. Right below that we've placed a card to change the current top-up option, as well as a keypad with suggested top-up values.



We've used a couple of autocomplete suggestions for standard top-up amounts for the strip above the iOS keypad. We want to incentivize our users to complete the top-up.

I've also included the action button to switch to a different top-up method without going a step back. Make your users' life as simple as possible with autosuggestions and quick action buttons.





1.7.7a - Card in your name

1.7.7b - Card in your name - error

Once we input our desired top-up amount, a modal stack with card details appears to ask us to fill out the rest of the information.

The details, of course, need to be true because otherwise, we'll see an error state with the improper input outlined in red and a descriptive text that explains the error. We should always describe the mistakes causing the errors in our flow.

2. Open banking transfers

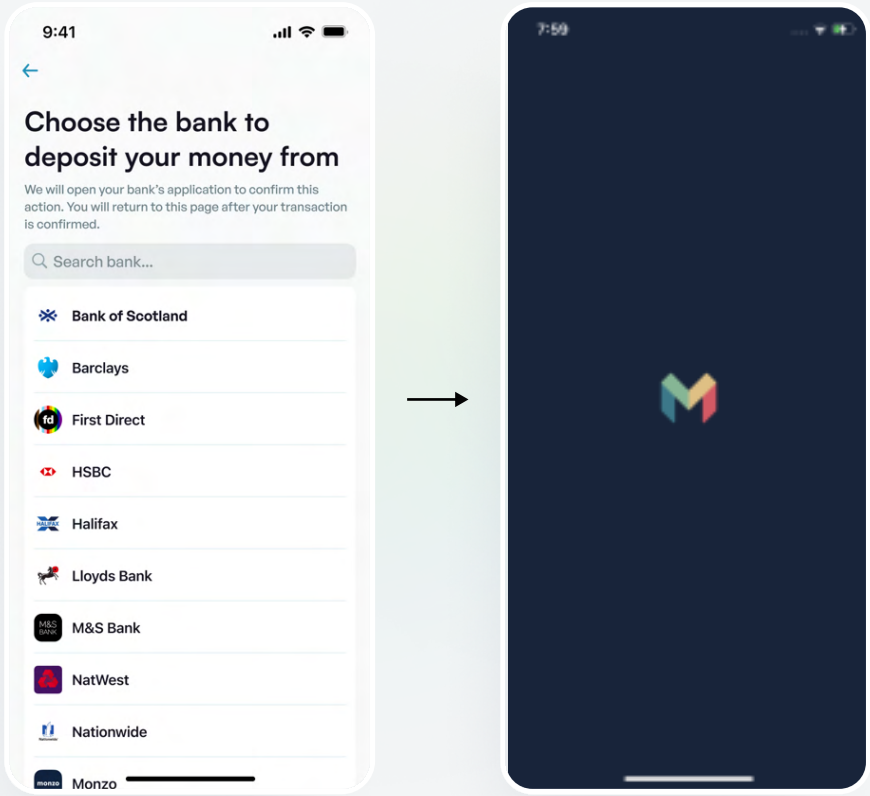
While Europe has a clearly defined framework, open banking in the US isn't so structured. There's no similar regulation to lay the groundwork for open banking, so you'd probably use it the same way we have designed it.

What's open banking anyway?

Open banking is also known as "open bank data." Open banking is a banking practice that provides third-party financial service providers open access to consumer banking, transaction, and other financial data from banks and non-bank financial institutions through application programming interfaces (APIs).

In layman's terms, it's used by banking apps to connect to other banks without breaking the in-app flow. You click on a bank you'd like to join to submit a transfer, and then you're being taken away from the app to complete the flow in another (banking) app, to then go back to SmartBank and finish the flow there.

Super helpful and convenient; not available in every country, though, as it's regulated by different financial institutions in different countries and regions, may be also called differently.

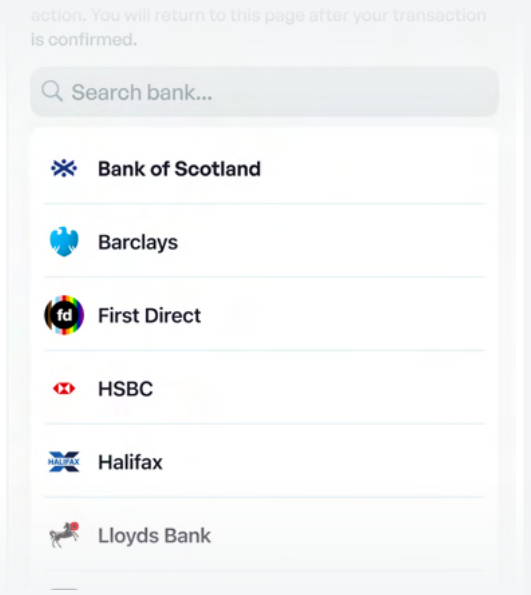


1.8.1 - Open banking

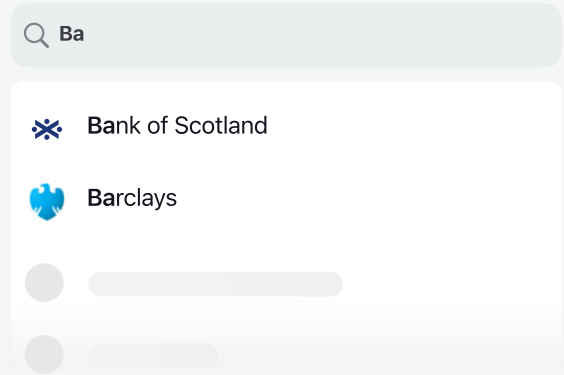
Open banking flow begins

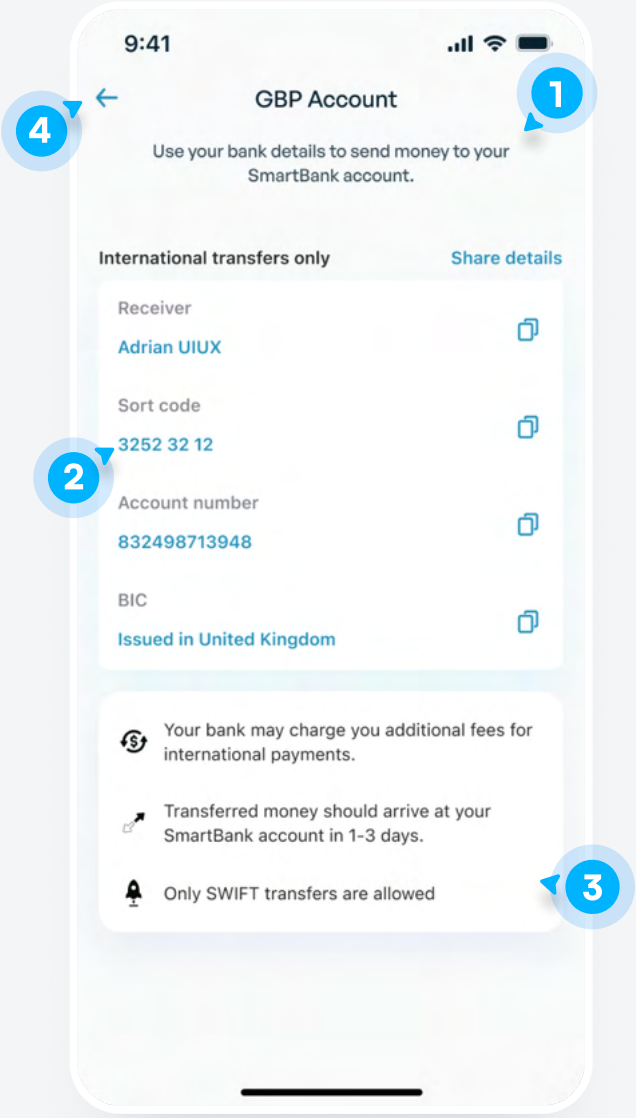
The list should include all eligible banks for open banking, which should be the majority of the biggest banks available in your country.

Once you click on any particular bank, it should redirect you to an app on your phone where you should be able to finish the transfer.



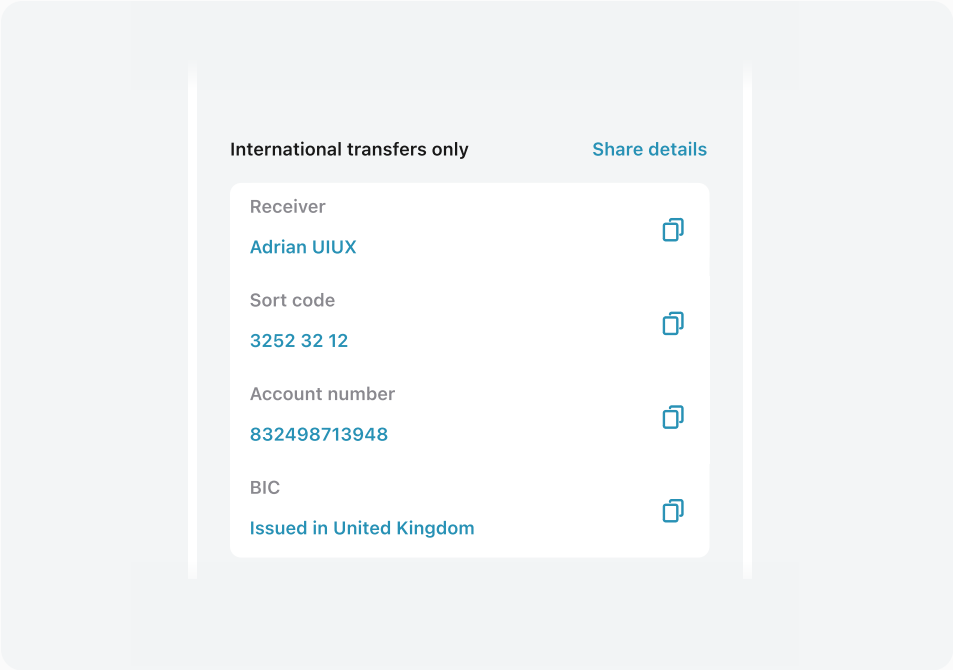
As you might remember from one of our previous pages, we should consider using a search input for lists longer than 6-8 items. It's much easier to just start typing the name and select a suggested bank instead of endlessly scrolling through a list of 100 different providers.





1.8.2 - Account details

Last and the most basic method of a top-up - copy the account details and manually transfer using your current bank account.

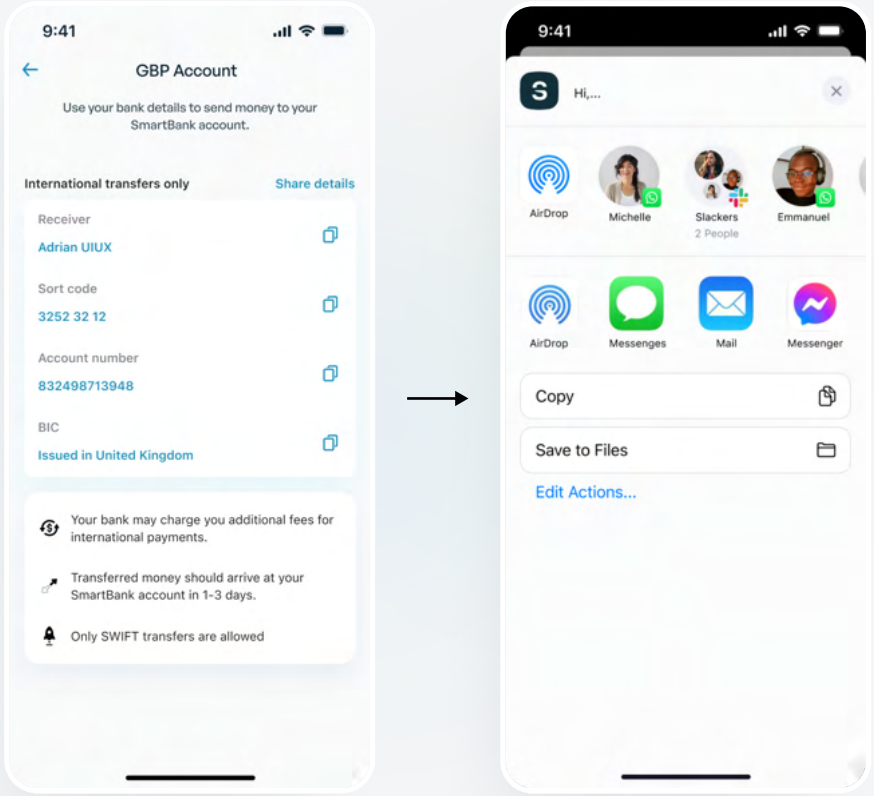


Share details

We're presenting a couple of options here. Copy the details with one click and share them outside our app using Gmail, Messenger, Whatsapp, Messages, etc. Clicking on it would open up a standard iOS share sheet with all the possible options.

We could also copy each detail one by one and send them manually. We want to leave this option open.

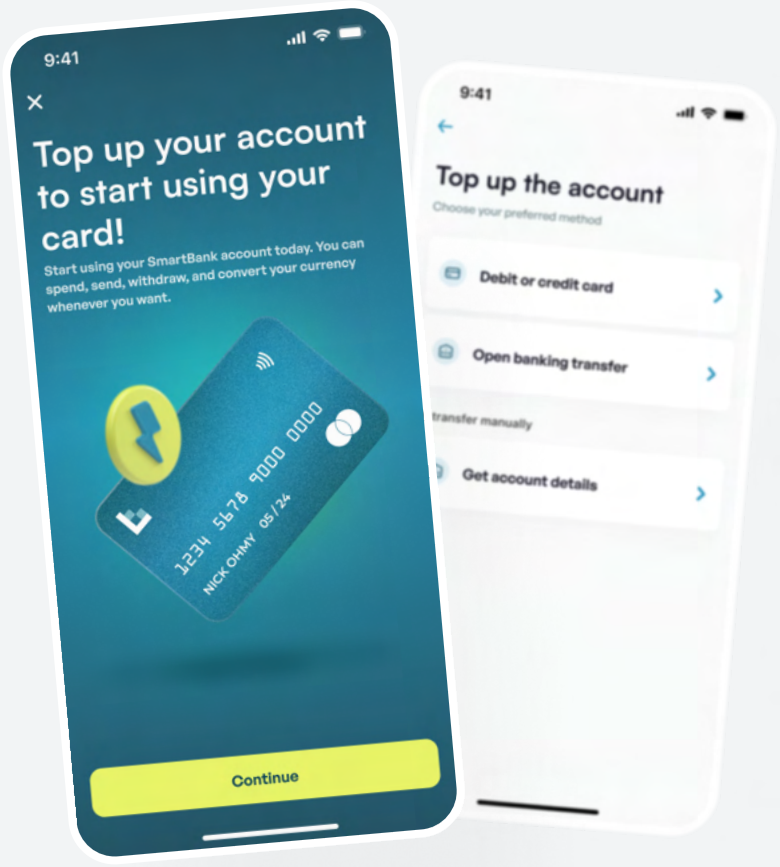
iOS Share Sheet



1.8.2 - Account details

1.8.3 - Share Sheet

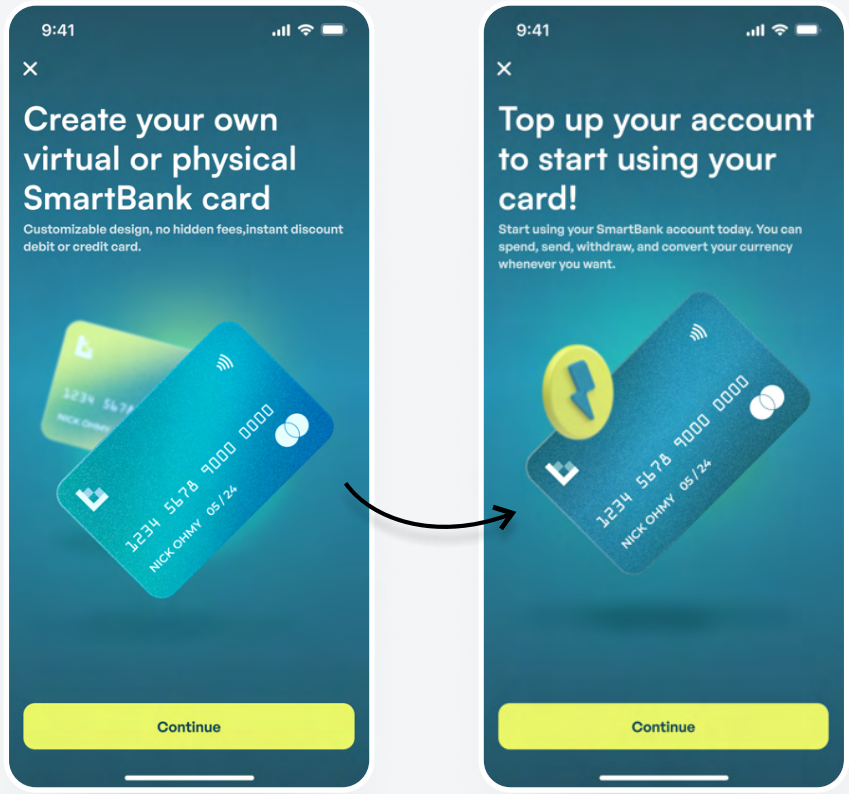
That's about it! We've covered the top-up flow. Now it's time to break down the structure of every frame and see what design patterns we've learned in this part.



Top up account

THE STRUCTURE

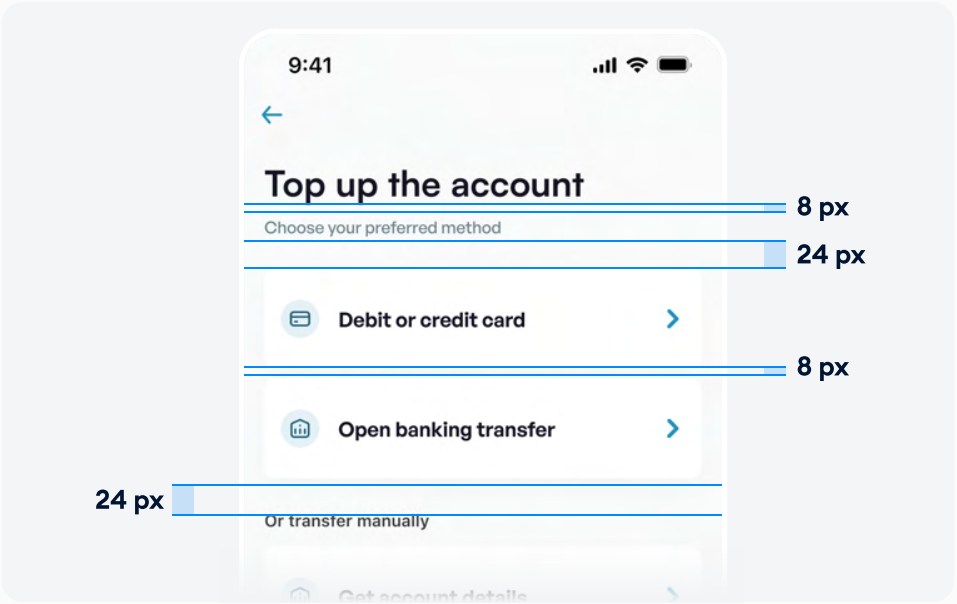
Personal details - the structure



1.7.2 - Create your SmartBank card

1.7.4 - Top up account

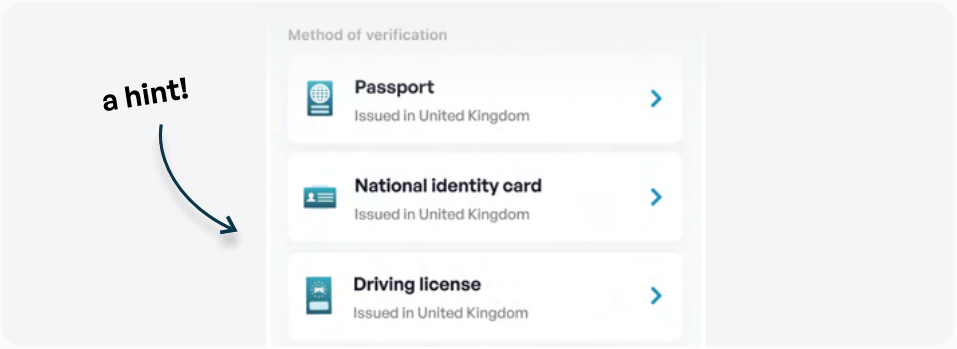
One glimpse at the “Create your card” tutorial screen, and we already know what frame to use as our reference to replicate. Start updating our copy and replace the images.

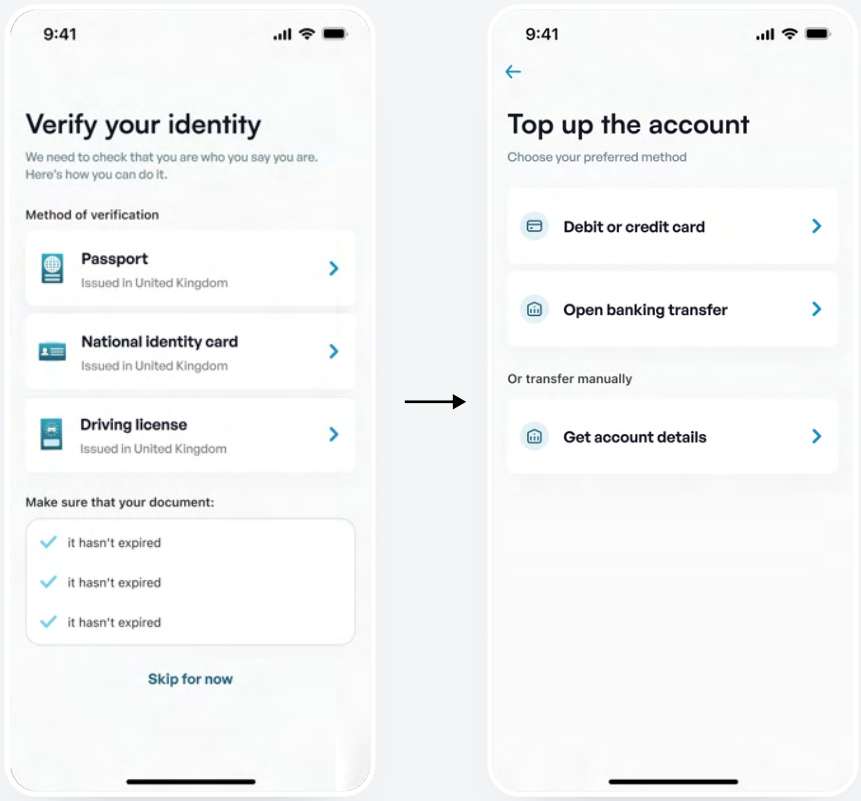


This part also looks familiar, doesn’t it?

For our top section, copy the Title Container from one of the previous screens and update the text.

Take note of how these cards are built, and let’s find a frame with a similar design pattern.





1.6.2 - Verify identity type

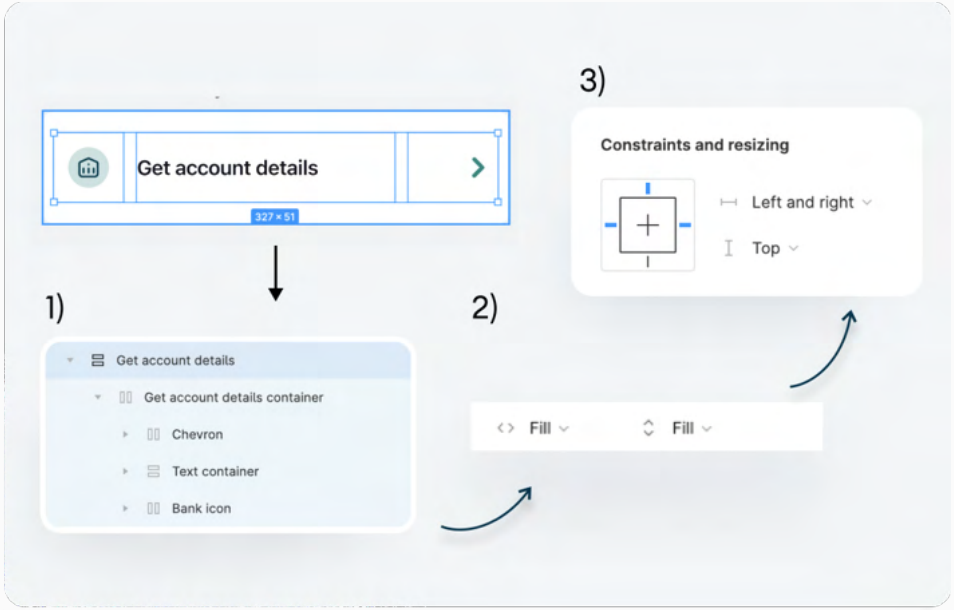
1.7.6a - Top the account

Let's use our "Verify your identity" frame to base our new screen. First off, update the text of the label and description.

Delete the descriptions under each label for the second section and update the text. Replace the images, and we're good to go!

But what if you wanted to recreate these cards yourself? Look at the next page for a more detailed breakdown.

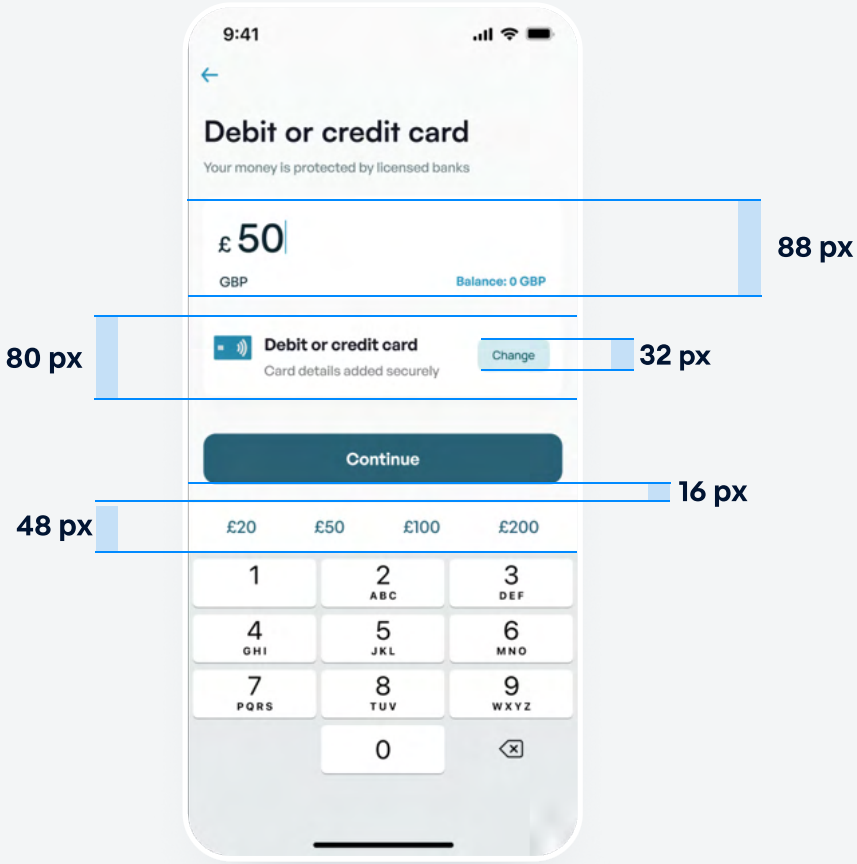
Top up account container



To understand it look at each one of the layers within the container and see how it's structured 1). Look at the right properties panel to see the auto-layout properties set 2) and then compare it with your design while trying to replicate it. Lastly, look at constraint 3) of the outer layer and check the responsiveness.

If it's not working correctly, then one of the layers has incorrect properties set.

Troubleshoot and re-adjust the settings. It takes some practice, but you'll get used to it.



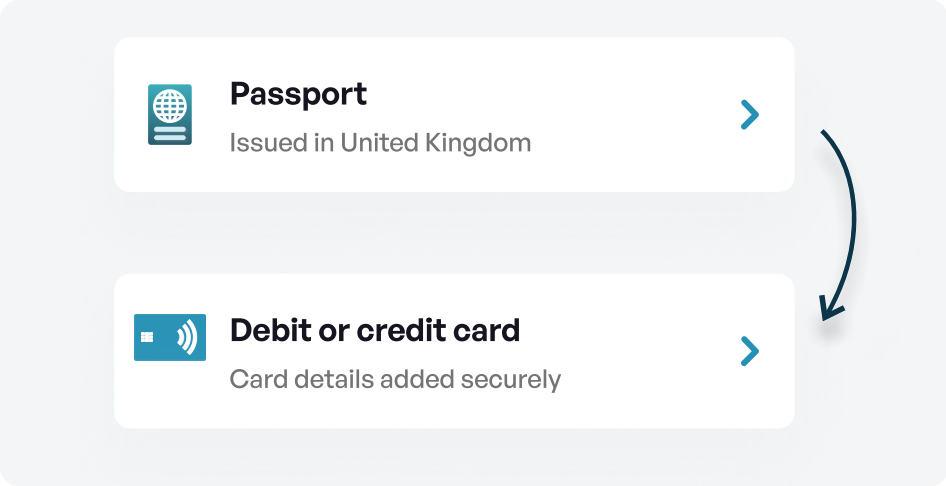
1.7.6b - Debit or credit card

We can use the input fields from our Components page for our next screen. Let's change the type of our input field to "Amount" and position it as 24 px from the top "Title container."

We'll need a little more work for the second card container, though, so let's tackle that next.

We don't need to start from scratch, though. If we look close enough, we'll find out that all the card containers we've covered are very similar structurally.

Let's use the one from our "Verify your identity" frame and adjust it slightly.

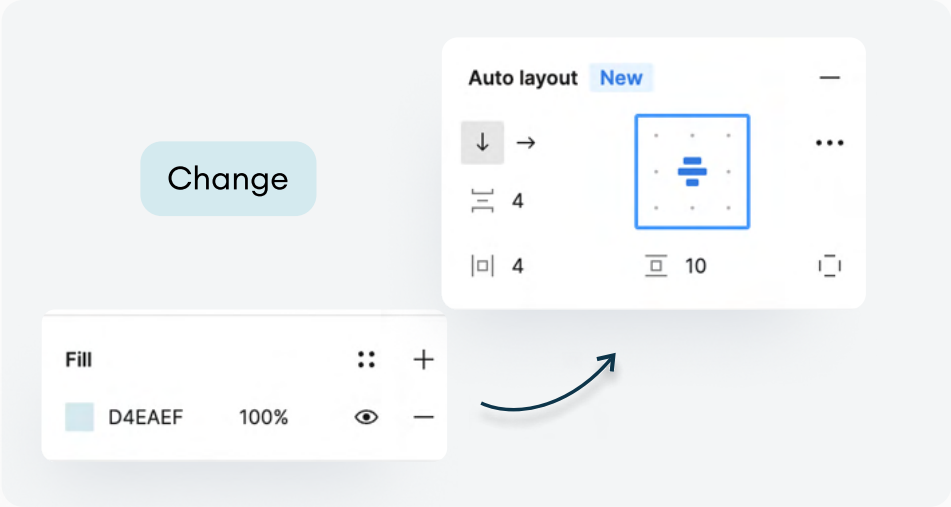


We need to create it before we can replace our chevron container with our button. Here's how you can do it.

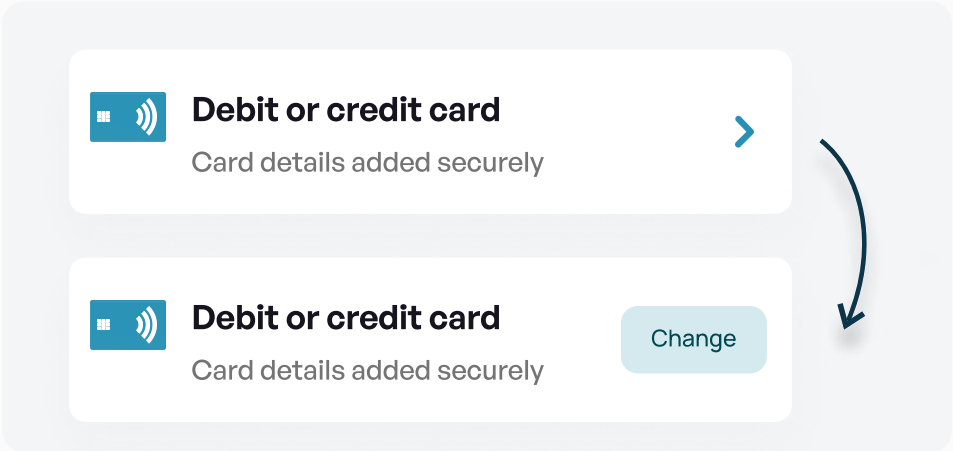


Create a text layer and write "Change". Apply auto-layout (Shift + A) and add border radius of 8 px. Add a fill of #D4EAEF.

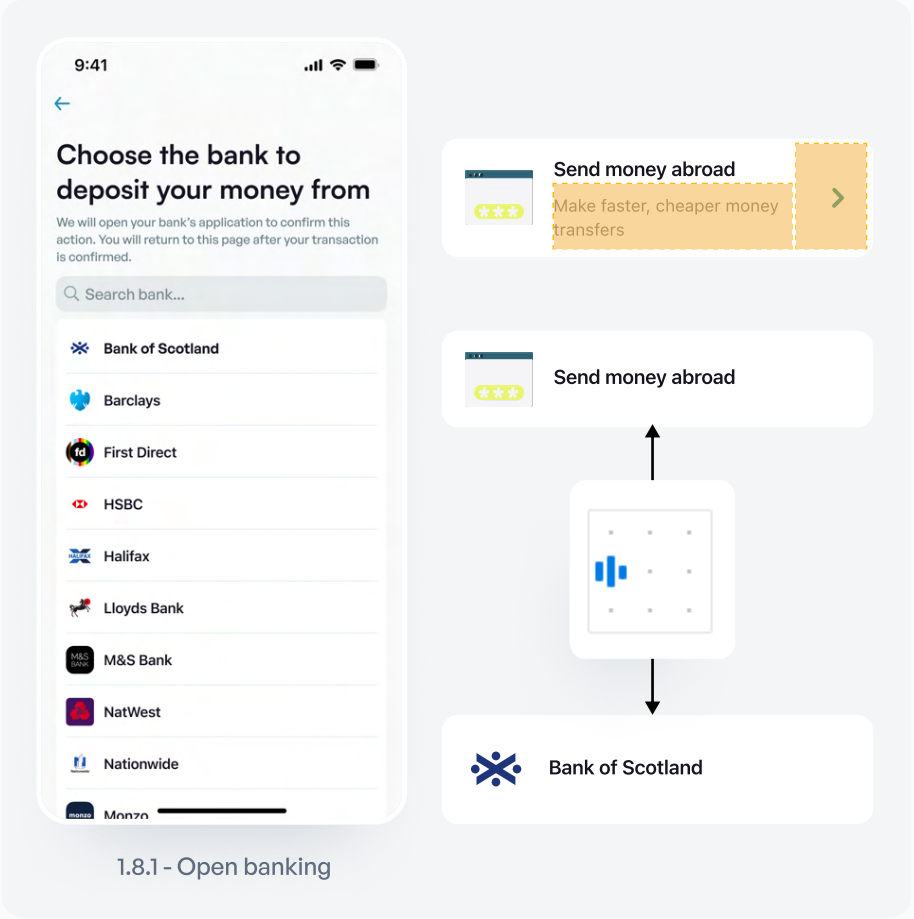
You'll find the exact auto layout properties on the next page.



Once you have the button set up correctly, select the frame and copy it by clicking “Cmd + C” on your keyboard.



Now select the Chevron container on our Card container and click “Shift + Cmd/Ctrl + Opt/alt + V” to paste the selection into the frame. It should look something like this.



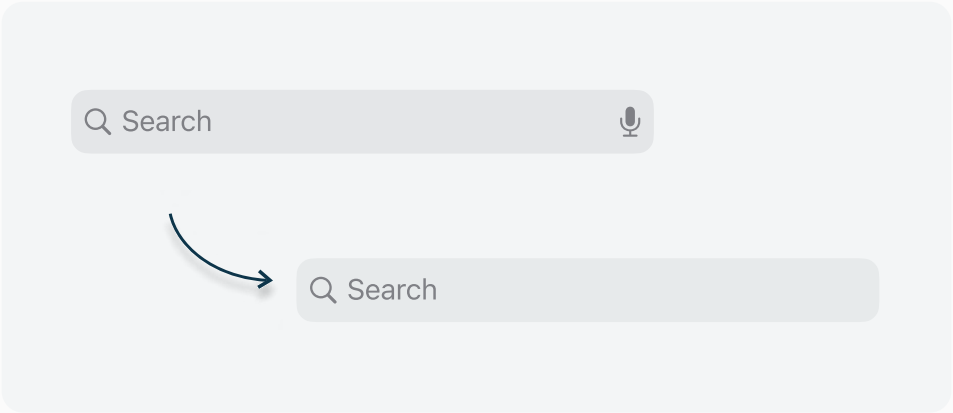
1.8.1 - Open banking

We can use the same card on our “Verify identity” frame. Just remove the text layer for the description and the container for the chevron. Ensure to remove the containers, not the layers nested inside, because the containers will still remain there.

When you remove the description, select your title container and make sure its resizing property is set to “middle” and “left” so it sticks to our bank image when it resizes to a higher resolution.

When you have one card created, click Shift + A to apply auto layout and nest our card within it. Then duplicate (cmd+d) it multiple times until you fill in the page. Replace the text and bank images, and of course, name your layers.

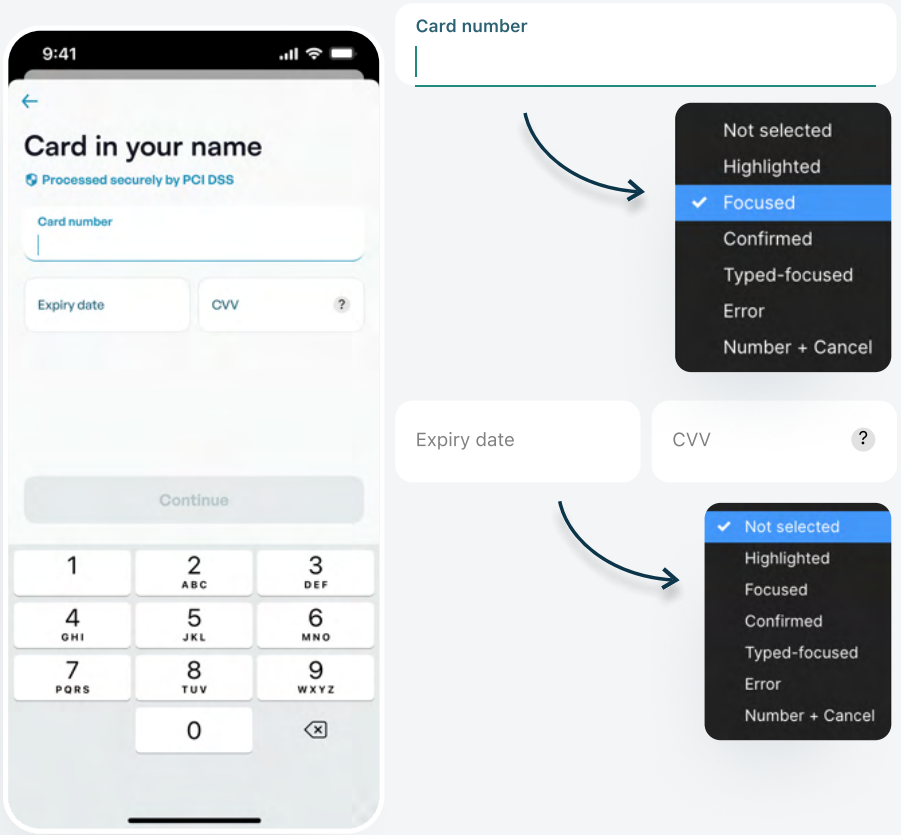
Let's move back up for a minute and create our search field. We can either copy one of the existing fields or find it in the iOS UI Kit and paste it onto our frame.



Let's adjust the style of our search field, so it matches our branding. We can also remove voice search as we probably won't have support for this functionality. Voice search tends to be a bit buggy in most apps, so let's say we voted it to be out for this MVP.

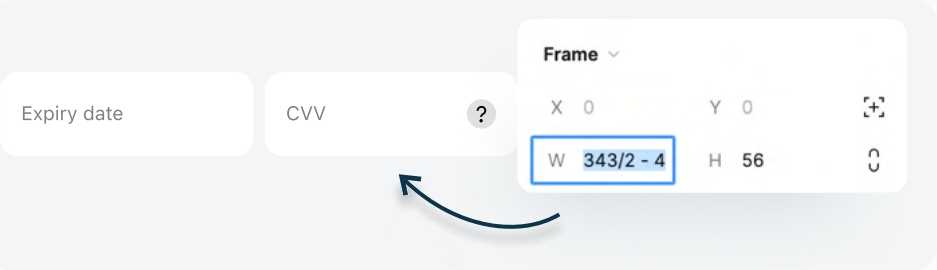
Card in your name

For the next screen, let's just use our standard input fields from the Components page, or copy them from our previous frames.



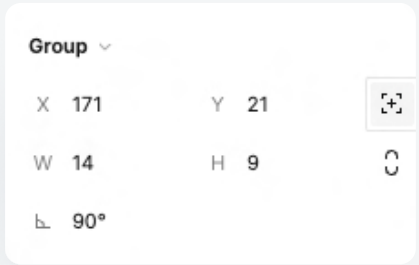
1.7.7a - Card in your name

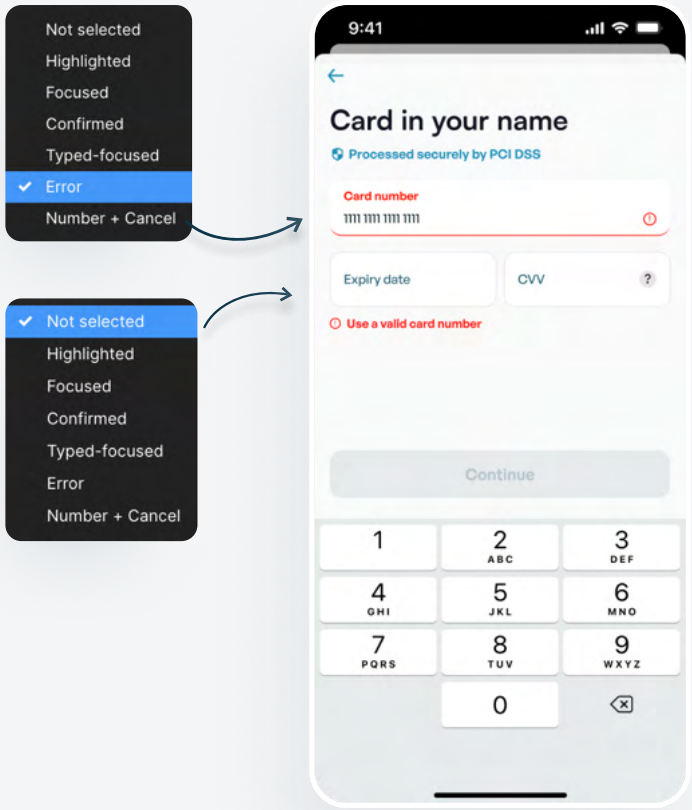
We need to adjust the width of the two bottom inputs. The easiest way would be to divide 343 in half and remove 4 px from that (we want to leave 8px between our two input fields).



To add the tooltip icon with the question mark, we simply need to add a text layer, write: “?”, create a circle, center aligns the two layers, and apply auto layout. Then drag the group into the input container.

It will automatically align vertically to the input container, but since the new Figma release, we can now set “Absolute” positioning on our elements within the auto layout containers. Click on the “?” group and navigate to your properties panel; click on the “Plus” icon next to the dimensions. This will let you set the position of your tooltip easily.

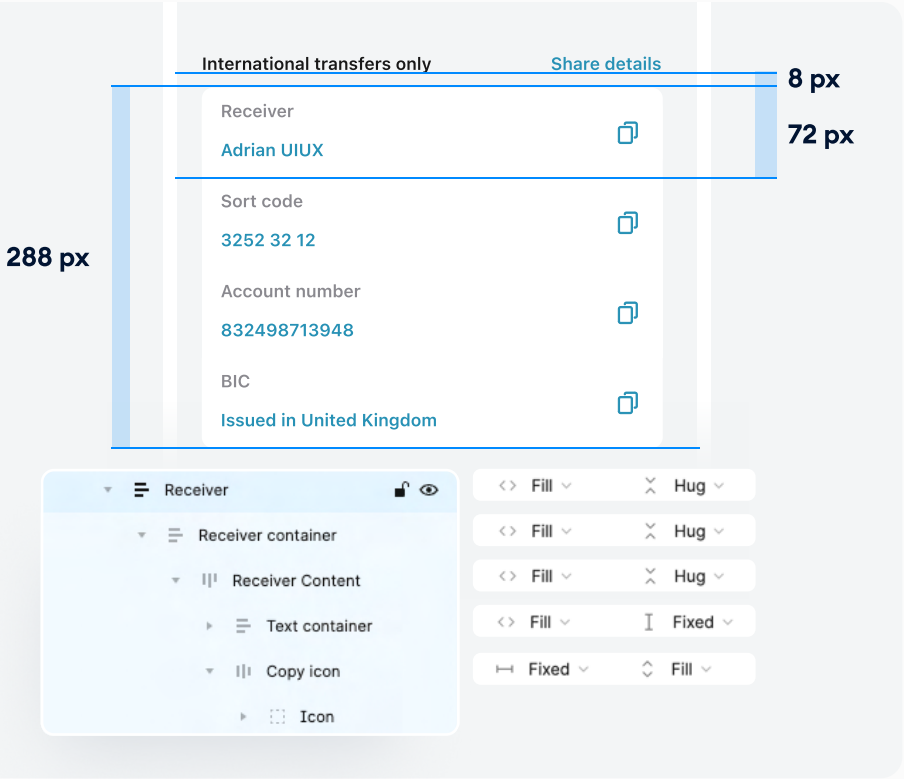




1.7.7b - Card in your name - error

Error states

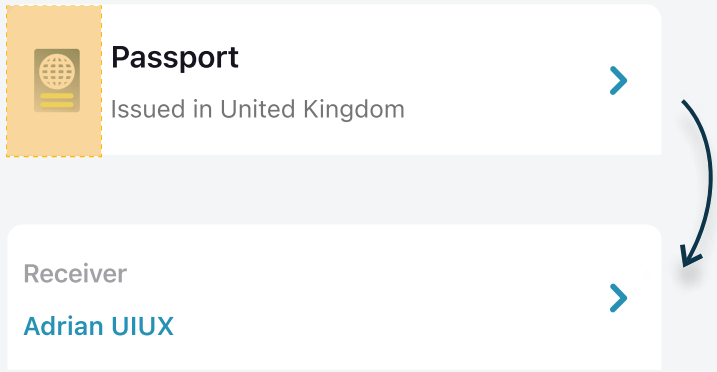
We just need to switch to an “Error” type input field for our error state and adjust the copy inside. If you feel confident enough, you can, of course, replicate the input fields yourself. If not, simply copy and paste it into our frame and analyze the properties of each element nested inside the container.



Get account details

Our “Account details” frame can be easily replicated using previous components. We can re-use the cards from our “Verify your account” frame. We need to adjust the style of our text layers and replace the chevron with our copy icon. We’ll also need to delete our icon container.

Always look at how auto layout containers are nested within each other and what are the resizing properties set on each of them. This will help you understand auto layout much better.

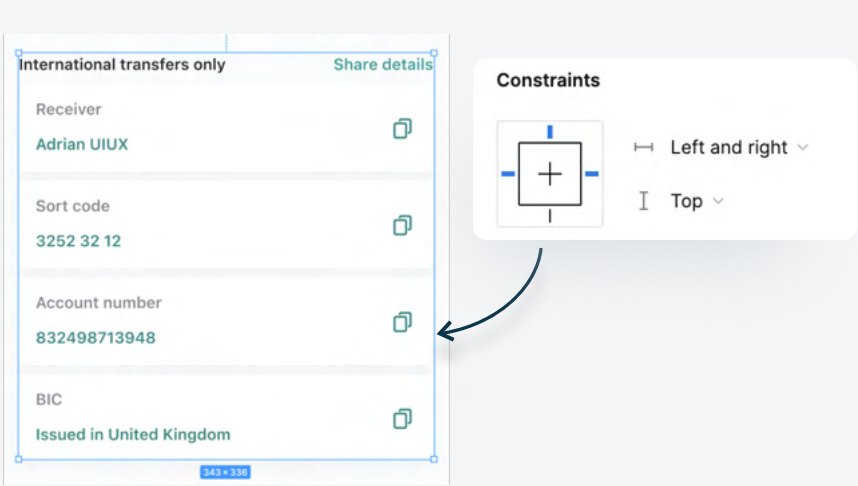


Make both of the text layer “Footnote 13px” style. Reduce the opacity of the label to 60% and change the color of the bottom text layer to our accent color #21897E and make it semibold.

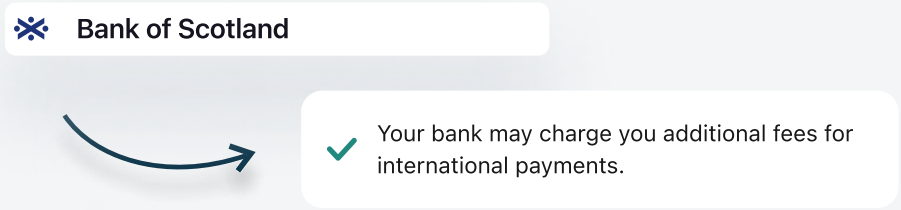
Next up, let’s replace our chevron icon with a copy details icon, and we’ve got our card container recreated.



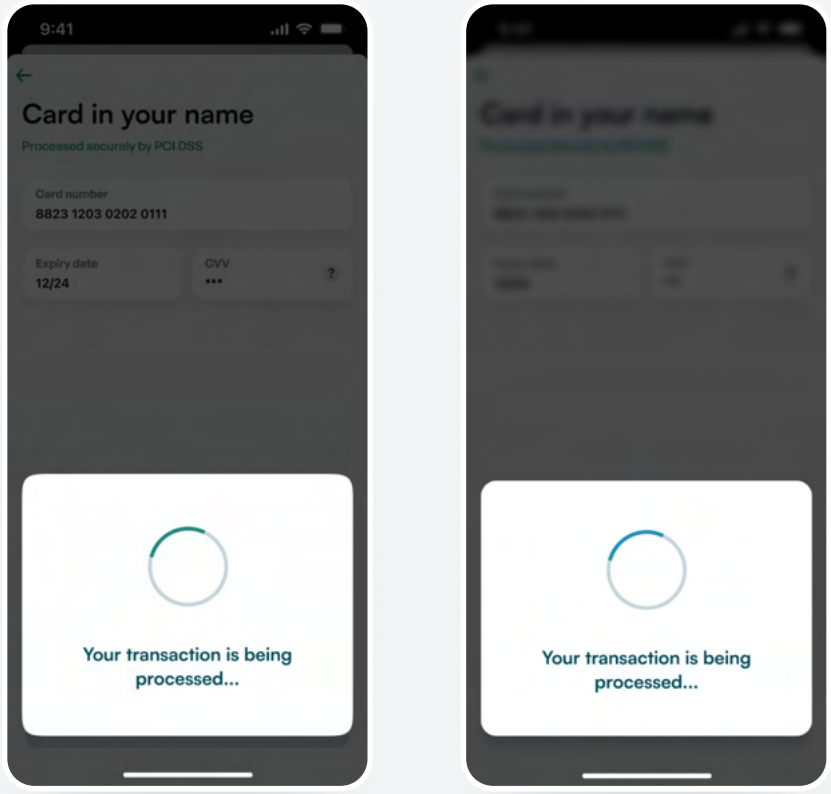
All we need to do now is apply an auto layout to our container and duplicate the card 4 times. Replace the text on all components and make sure we have our constraints set to “Left and Right” and “Top” to retain the same distance from the top of our frame.



Now it’s your time to practice. Try replicating our bottom section by modifying one of our previous frames (hint - you could use a container from the “Open banking” flow.



Transaction being processed

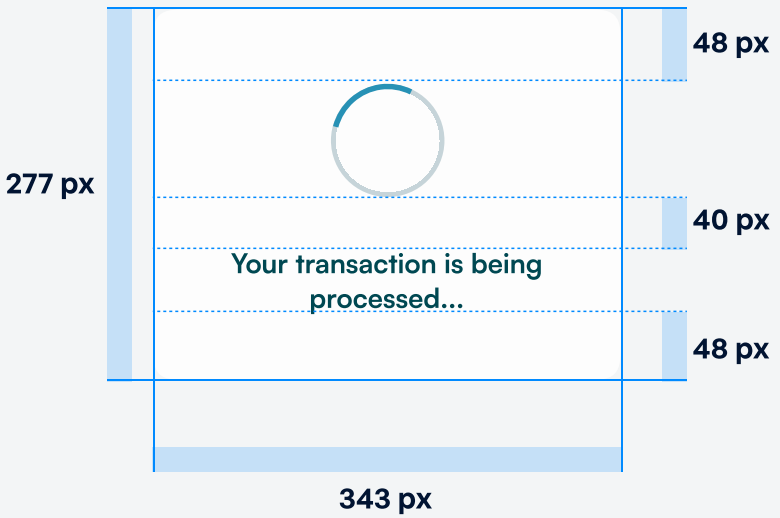


Without blur

With blur added

We just need to add a modal box with the dimensions you see above for our last screen. We can copy the Uploading circle from the “Verify your identity - confirmation” frame and add the text below.

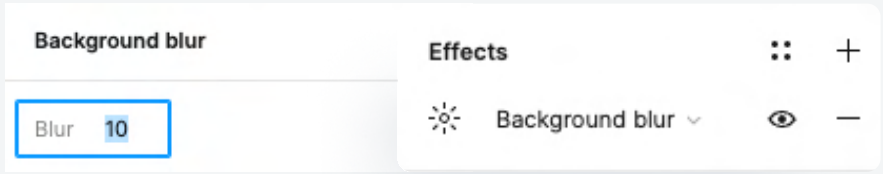
Let’s change our **input fields** to **“confirmed”** and update the text.



Change the style of the text to Heading 3 - 20 px and change its color to our dark green accent color. Remember to add a rectangle covering the entire screen. Set it to #000000 and change the opacity to 70%.

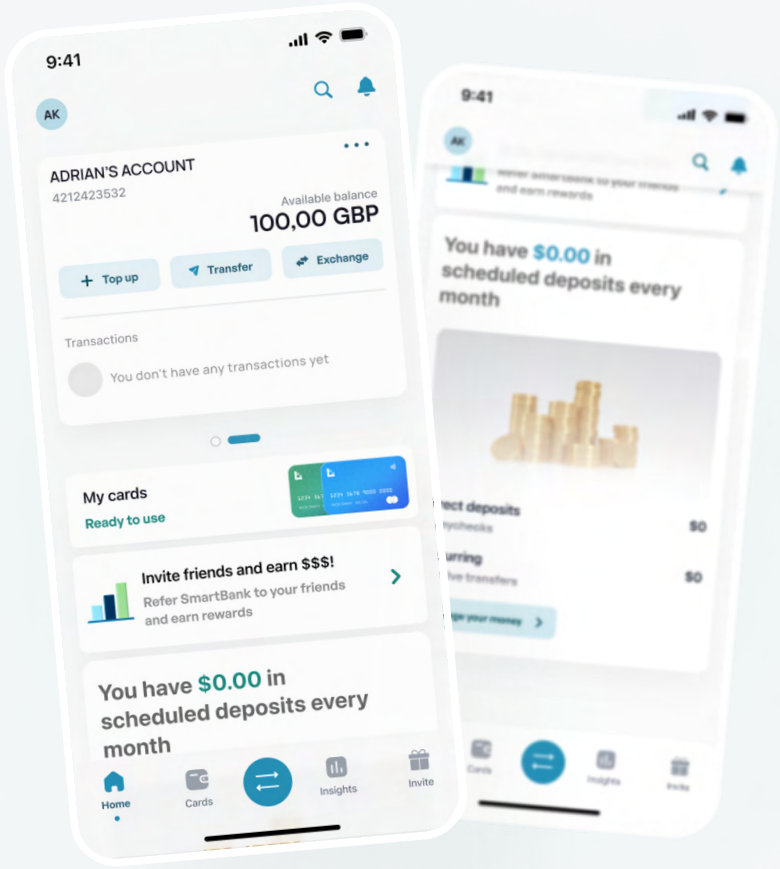
If you feel adventurous, you can also add a background blur to your overlay to create another layer of depth. Set it to 10.

Make sure it's below our modal box. That's it!



Chapter recap

- Open banking is a banking practice that provides third-party financial service providers open access to consumer banking through the use of application programming interfaces (APIs).
- If your components are not working properly then one of the layer has incorrect properties set. Troubleshoot and re-adjust the settings. It takes some practice but you'll get used to it.
- Always make sure contraints are set. Look at the left reference frames and check the settings there.
- Use iOS modal stack views for screens where there's a need to display an overlay modal instead of redirecting to a new screen.
- We should always describe the mistakes causing the errors in our flow.
- Absolute positioning allows you to exclude an element from the normal auto-layout flow without removing it from the frame itself. Pretty handy!



Home

THE INTRO

Home screen - the theory

We've done our research and decided which features our fintech app should have. It's easy to get overly ambitious and build a complete app with all the features you can think of straight away. Please, avoid the temptation to do so. It's super risky and will make the design and development process slow, complex, and clumsy.

Instead of going all-in on the functions, we focused on the core functionality for our "first release," and we'll gradually build outwards. This ensures a faster time to market and more time for testing and iterating. It also stops us from wasting time on extra features our audience might not even find valuable or need.

An excellent example of that approach is to just look at earlier releases of many popular apps to see how simple they've started out and still, throughout the past few years, grew exponentially to become real giants in the fintech industry or any industry for that matter.

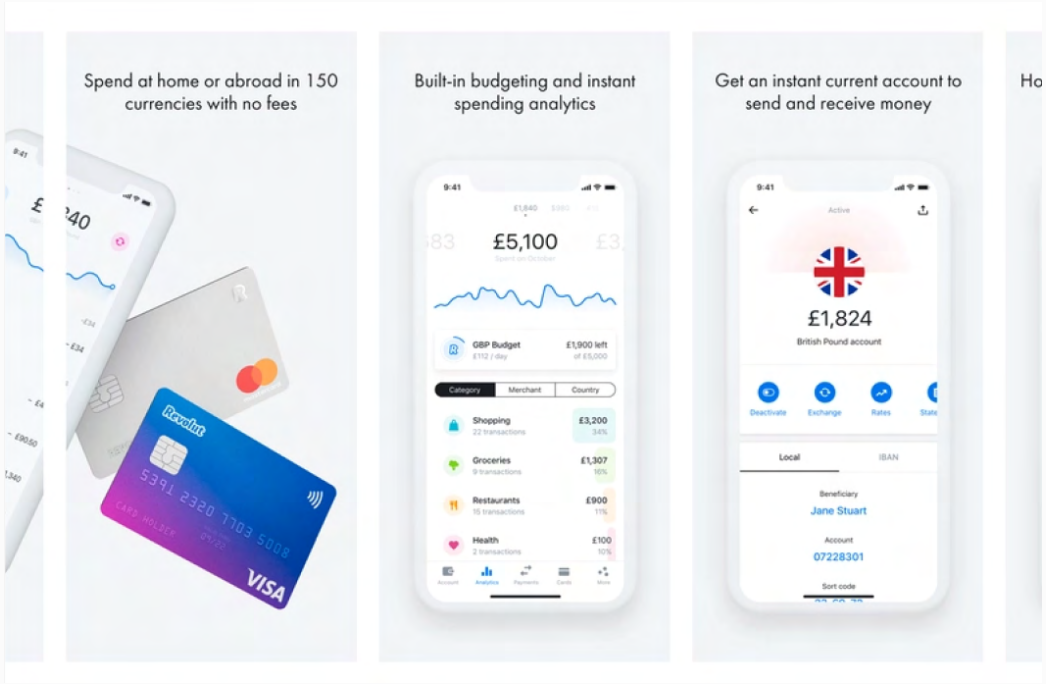
This ebook is like a ver 1.0 of a fintech product, and the following updates with new supplementary flows can be treated as new app version releases.

The evolution of Revolut

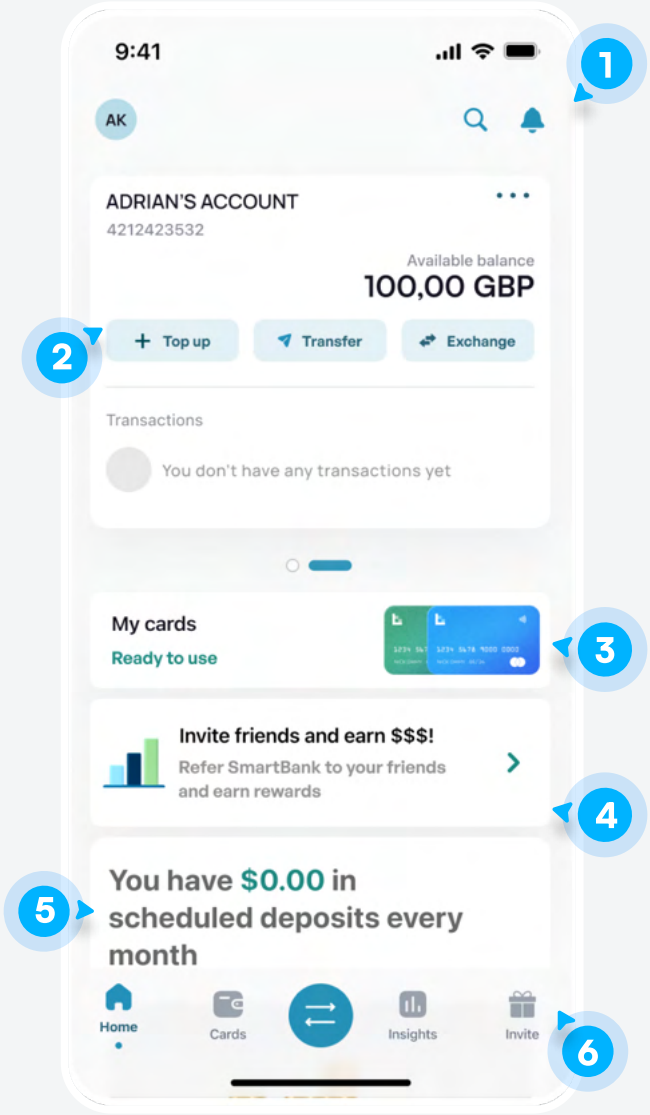
Look at how Revolut evolved in the past 7 years, growing from a user base of 100k people in their first year to over 15 million, £261 million revenue, and a £33B valuation in 2020.

With this kind of money, you're free to do as many tests and iterations on your product.

It's better to start slow and easy and develop as you go.

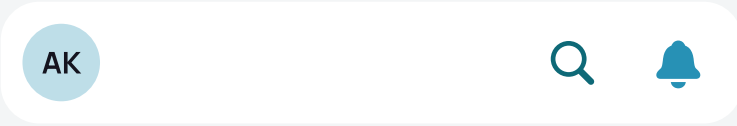


With this little intro out the way, let's cover our app's main screen and what it consists of. Let's go!



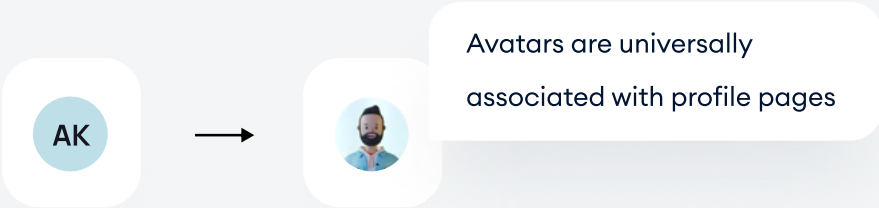
1.9.1d- Home screen - topped up account

1. Top Navigation bar

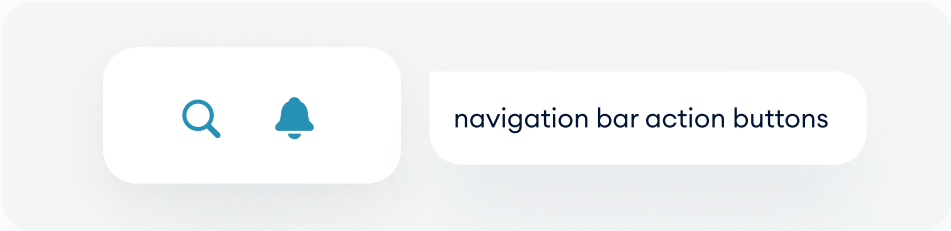


First up, we have our top navigation bar. As the name suggests, it provides an additional way of navigating the app. It's important not to overcrowd it and offer too many actions, though. We want it to help our users, not overwhelm them with too many navigation controls.

Remember that the top navigation bar provides content and actions related to the current screen. We can pull up all the personal details in our app by clicking on our avatar on the left. We have a search icon on the right-hand side to quickly pull up several results based on a current search query and lastly, we have in-app messages.



We've added user initials for empty avatars and leave the option open for users to upload their own image avatars if they want to.



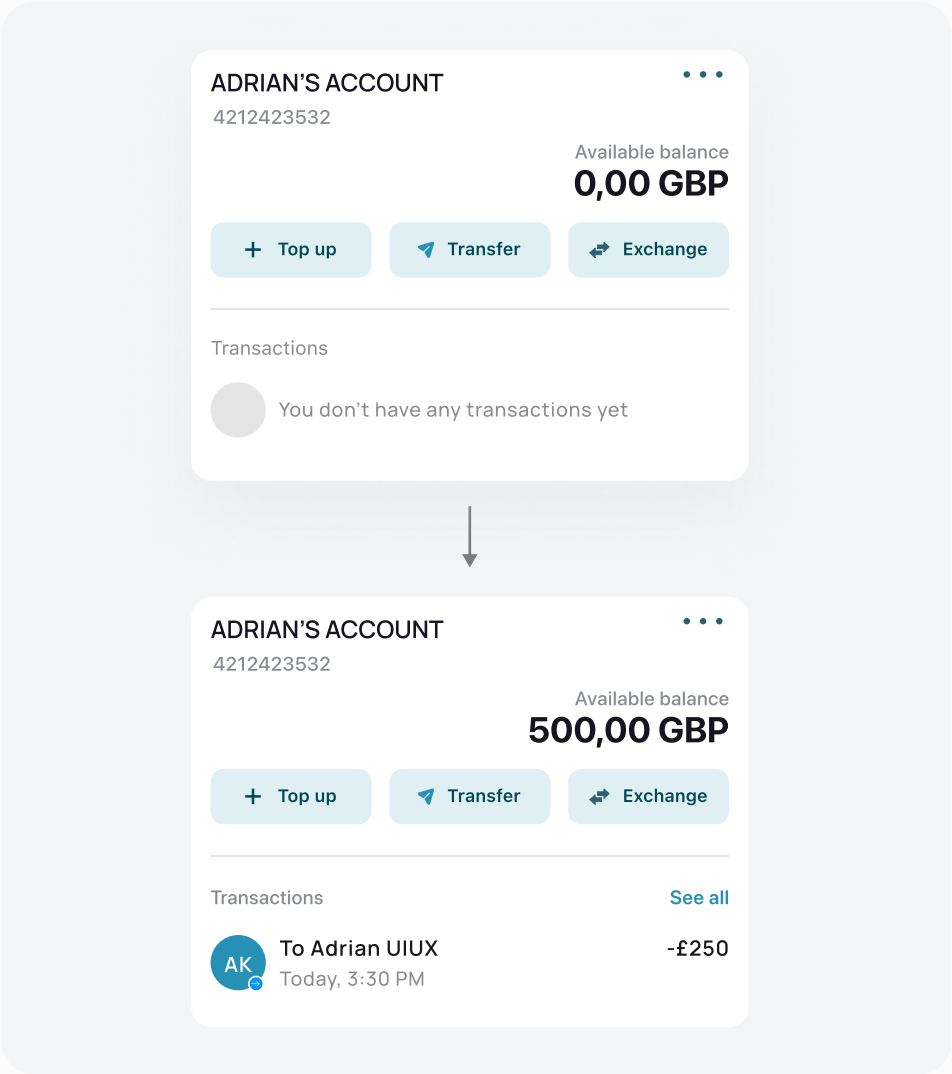
Buttons on the navigation bar shouldn't exceed two or at most three actions. We've considered adding the account insights here as well, but in the end, we decided to move it down to our main navigation element - the tab bar.



2. Account balance(s)

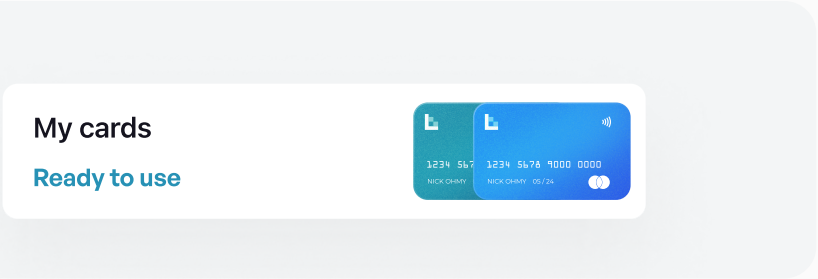
We'll be logging in to this app to check our account balances, and send, receive, and exchange money, so I wanted to make this super easy for all users.

Right after we land on our main screen, the first thing we see is the card with our account, quick action buttons like "Top-up," "Transfer," and "Exchange." Once we take any action on this account, we'll also start seeing "Transactions" appear right under our action buttons.

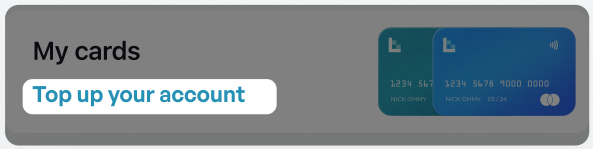


Once the transactions start rolling in, you'll be able to click on the link button opposite the label and open up a more detailed view of all the recent activity associated with that specific account. More on that coming later.

3. My Cards

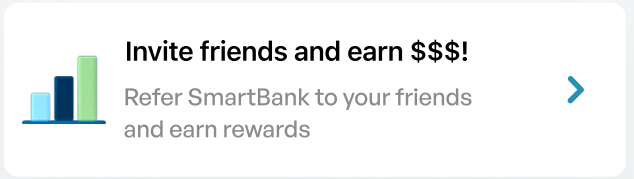


After clicking this card container, we will be taken to see all our current cards (virtual and physical). If we skipped this part of the personalization wizard, didn't open a bank account, and didn't top up our cards, we would see a different description, incentivizing us to top up the account to start using our cards. Use copy to your advantage, and always use action verbs for the call to action.



We're using clear and straightforward action verbs to incite action in this example. Clicking on this element would take us back to the "Top-up" flow. If it's a second time we're topping up, we will skip out on all the intro screens.

4. Invite Friends



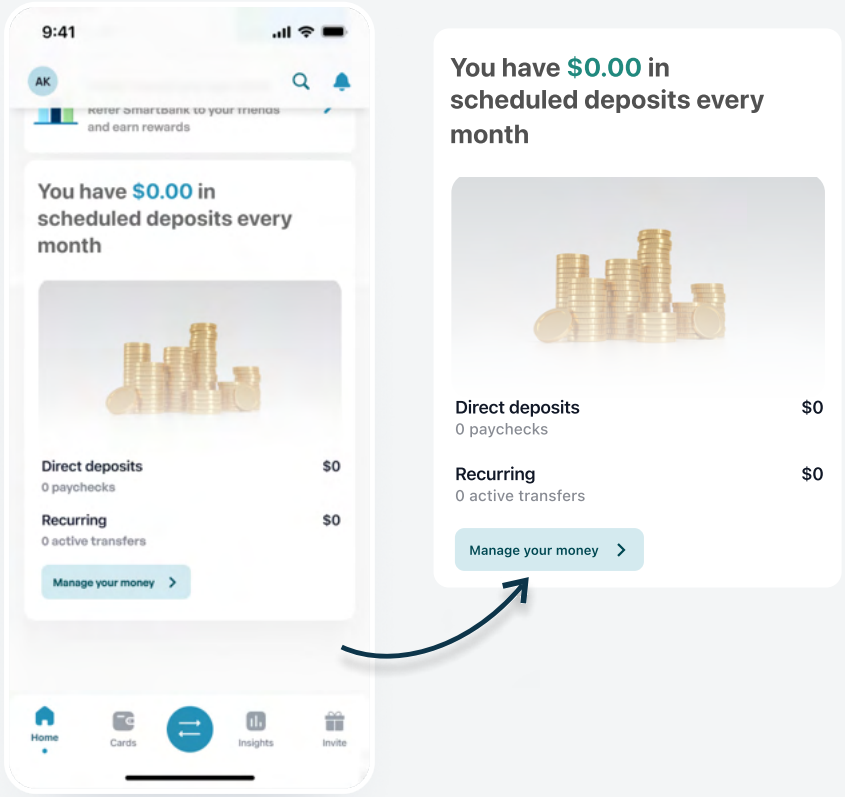
This is a little gamification play to attract new customers through one of the most powerful marketing methods - word of mouth. When you're already signed into the app, you'll get access to an affiliate program, where you may earn gifts, or reduce your fees (more common) by referring friends, family, or even strangers.

Clicking on it would take us to the "Invite your friends" screen which we'll discuss in later chapters.

A referral program is a marketing strategy that encourages current customers to share around the brand. Customers typically receive gift cards, discounts, or points in exchange for their efforts.



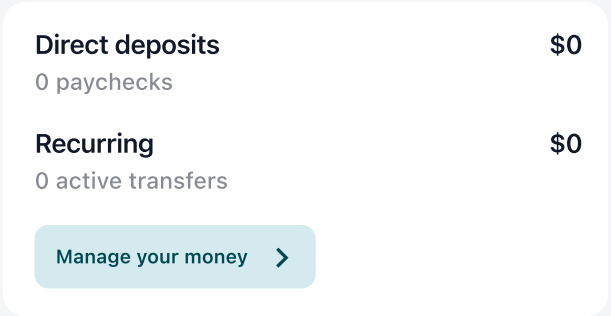
5. Scheduled Deposits - Insights



Scheduled deposits are a super handy feature for all users planning their future transfers and need an automated way to manage their balances and savings, and we've seen it being used

This option enables us to set recurring and/or fixed amount deposits into a deposit account every month.

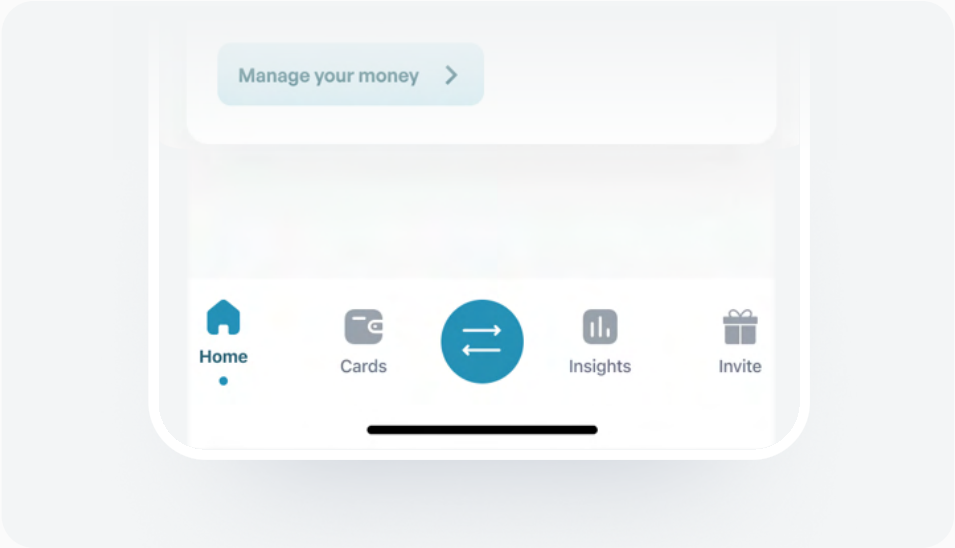
For some, it's a huge time saver, and it helps to keep everything in check. Customers permit the merchant to automatically withdraw funds from their accounts at regular intervals on an ongoing basis.



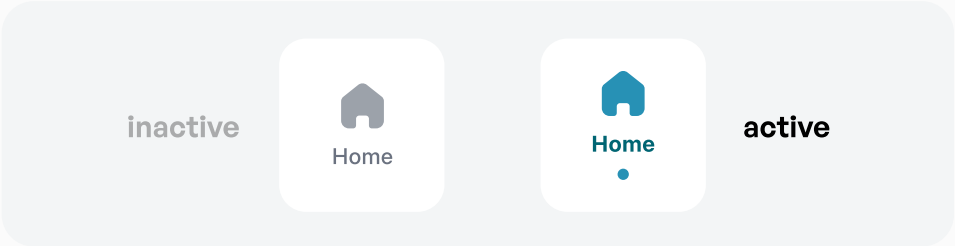
Clicking on "Manage your money" would ideally take us to the "Savings" flow, but since it's ver. 1, we'll open up an account view with the account insights and build this function up in later releases.

User convenience should be one of our top priorities when designing experiences and thinking about adding additional, functional elements to our designs. Although we shouldn't be making all decisions without thoroughly researching and testing them first.

6. Tab bar - bottom navigation

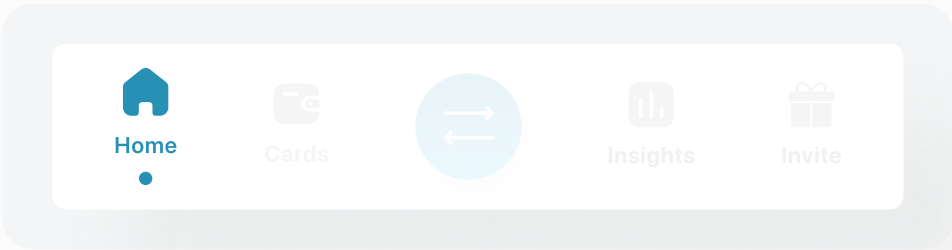


Active indicator



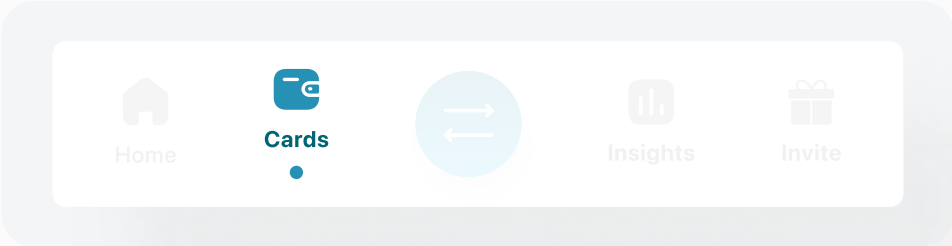
One of the most fundamental UX/UI design principles is the Visibility of system status. The active state indicator is one of the most prevalent types of feedback you can give your users apart from increasing the contrast of your icon, increasing the thickness of your type, or changing its active state color altogether. Let's cover our tab bar next.

Home



This is essentially the hub of our app, with every card and component taking us to the start of each respective flow we have designed. We want to treat this page as an overview of all the things happening with our accounts, cards, and balances to quickly and easily take the desired action.

Cards



Cards are where all our physical and virtual cards are located. We can manage our cards, see the details, edit the limits, see the transactions, and freeze them if needed, all from this page.

We'll cover it in detail in later chapters.

Transfer money



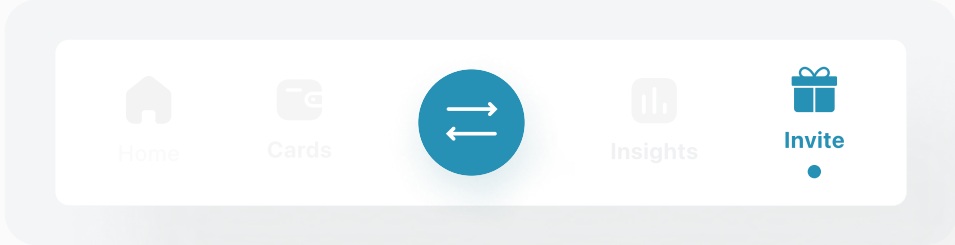
Right in the middle, we've got the transfer money feature which, most likely, will be the most frequently used out of all. That's why it also has the most prominence. The icons must be noticeable and recognizable just at first glance.

Insights



Banking Insights give you access to visual data to better picture your finances, earnings, spending, and savings. SmartBank will analyze your debit and credit card transactions and deliver personalized tips regularly to your feed in the form of in-app notifications and personalized promotional offers.

Invite



And lastly, "Invite," a one-page flow with your share link and a list of all the users you've personally invited to the app.

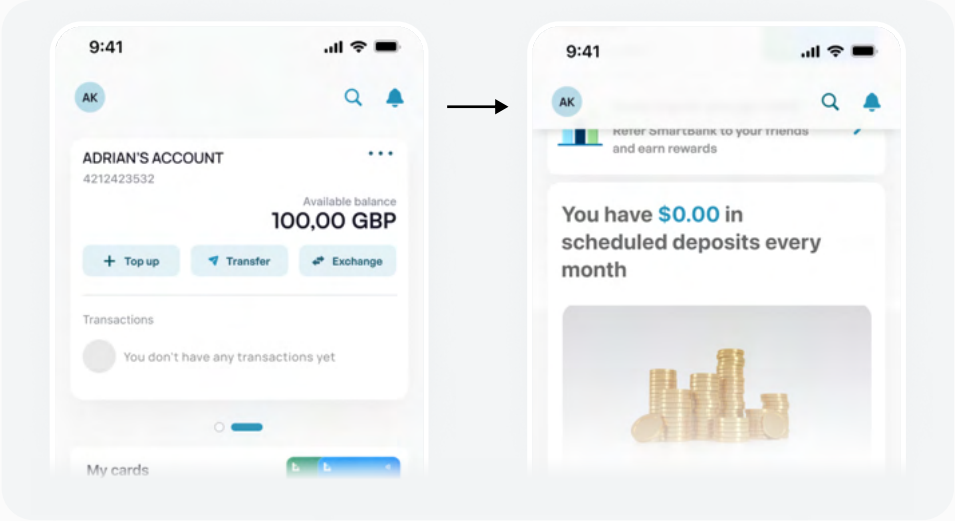
We'll probably remove this from our tab bar and move it to a home screen card as a separate call out in later releases in favor of other functionalities but for ver 1.0, where word of mouth marketing is one of our biggest strengths, let's leave it visible and easily accessible.

By default, iOS tab bars should display up to 5 tabs, with the last one being "..." or "more." If the tab bar holds more than 5 controllers (important pages or flows), consider hiding it in the "more" list. Try to leave the most important pages visible and hide secondary screens that are not that important.

In our example, we've hidden the "Profile settings" in an "Avatar" and "Search" + "Messages" in our top navigation bar icons leaving all the important controllers in the tab bar.



Top navigation - scroll



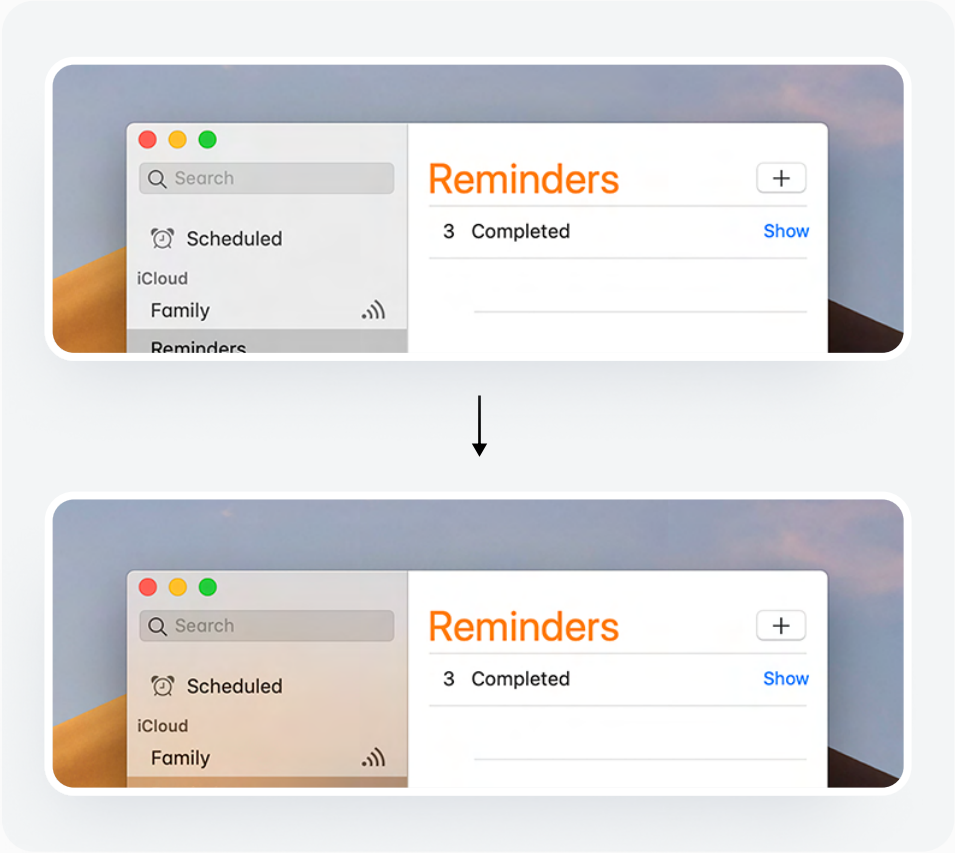
We have to mention one crucial micro-interaction that you won't be able to see just by looking at our designs without actually prototyping these interactions. It's the top navigation bar and how it behaves in relation to your scroll position.

When scrolling past the initial position, especially with rather long screens and important action items in the top navigation bar, we should consider making it sticky. Sticky, meaning its position will remain locked and follow us on-screen whenever we scroll down or up.

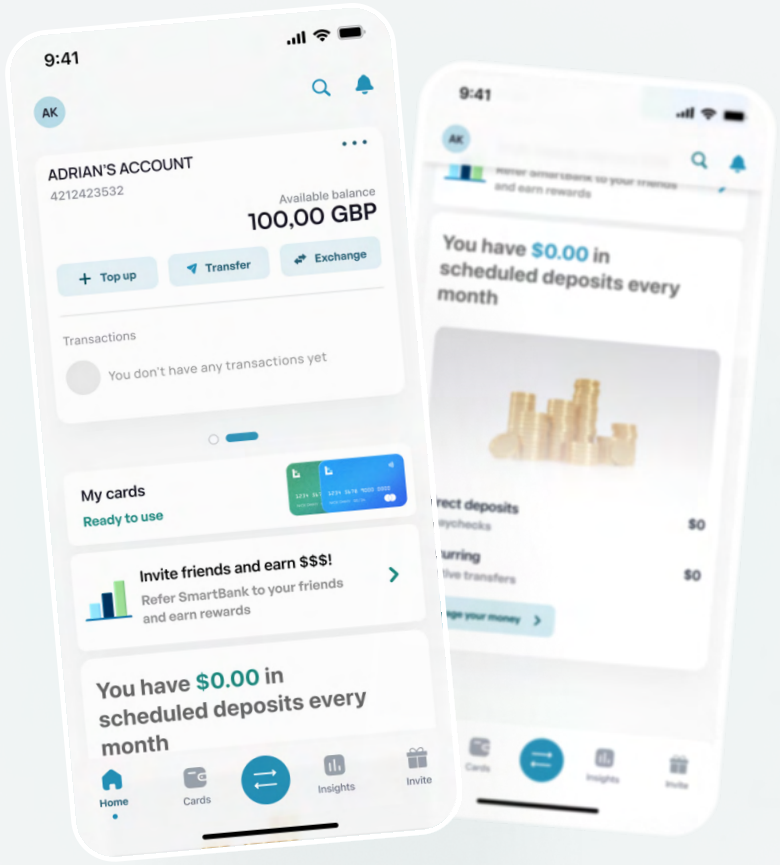
In iOS, you can also hide this navigation when you scroll down but slide it back out again when you start scrolling towards the top.

iOS Transparency and blur

Blur, transparency, and movement are effective ways to convey information. Some people find them annoying or uncomfortable, so we really want to make them subtle, almost not noticeable, and in line with iOS design patterns.

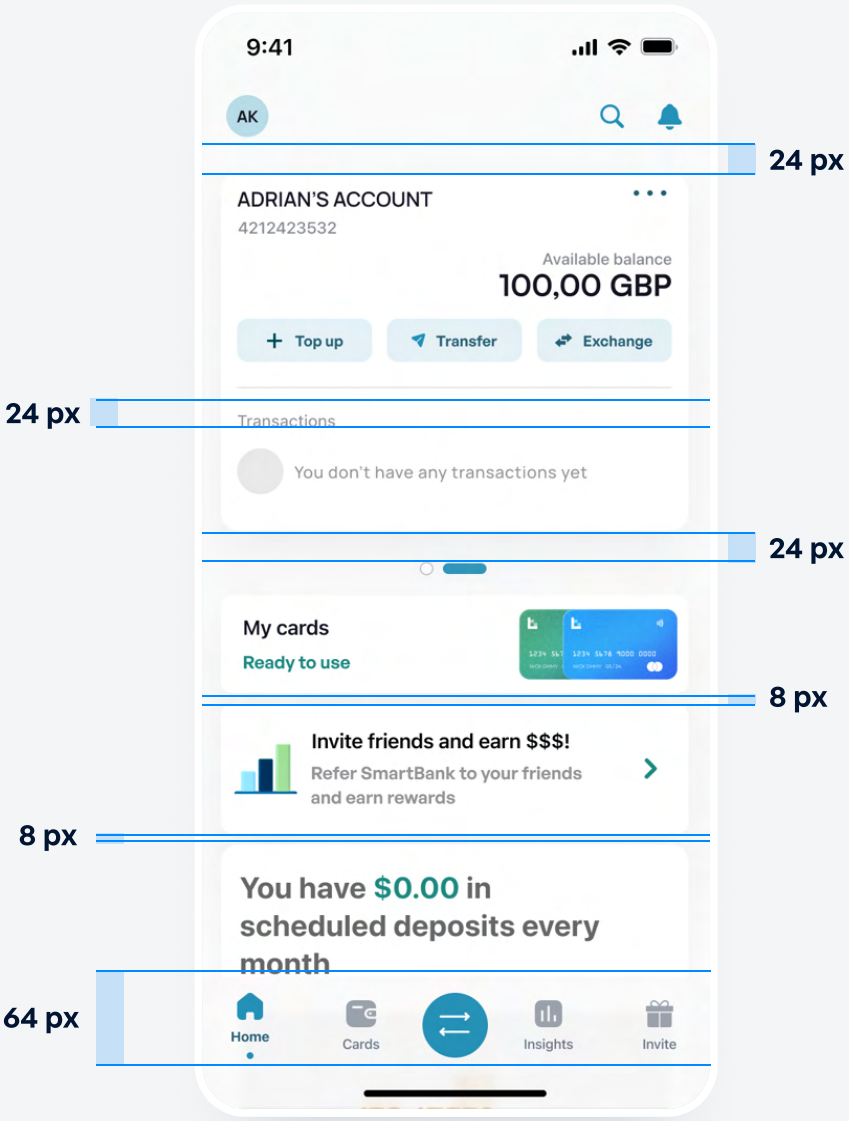


It's not a must, of course. Still, it adds to the visuals, and your users are already familiar with it interacting with other Apple devices, so they will even subconsciously recognize it as an iOS pattern.



Home

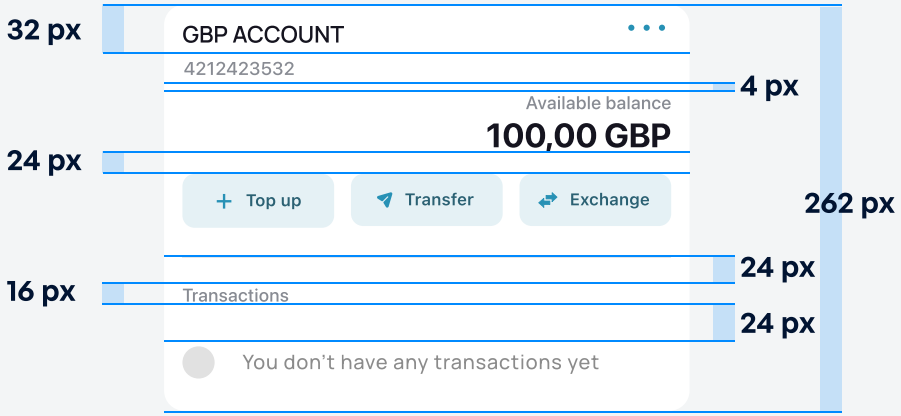
THE STRUCTURE



1.9.1d- Home screen - topped up account

Home Screen - the structure

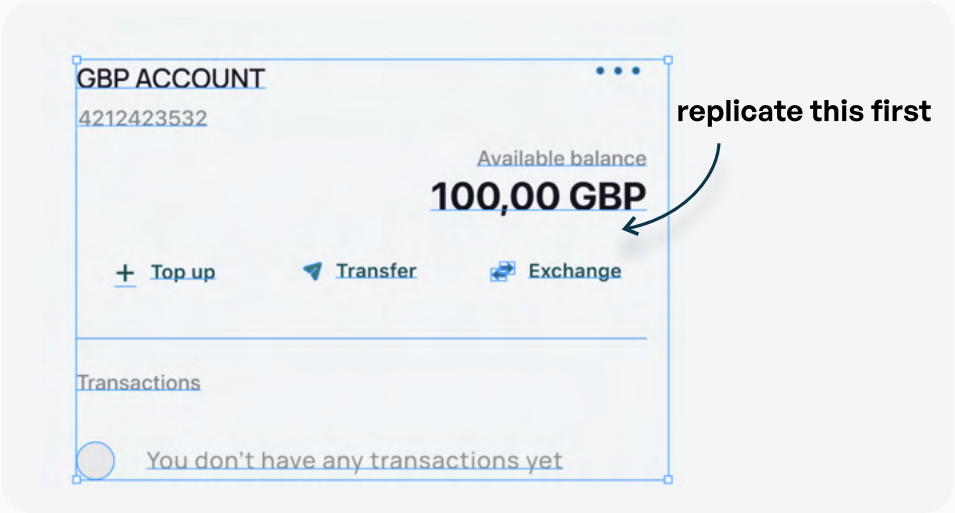
Account balance card



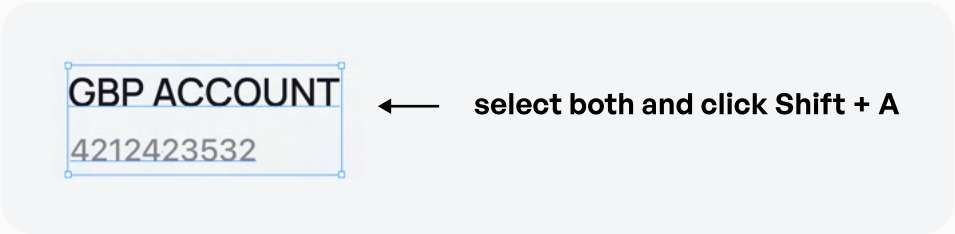
This container is probably the most complicated out of all we've covered thus far. Don't worry; it's not hard if you take it slow and analyze every container.

If I were to do it from scratch, I would start by just placing all the elements I see here on a practice frame without applying auto-layout anywhere. We'll use our containers later on. Let me show you what I mean by that.

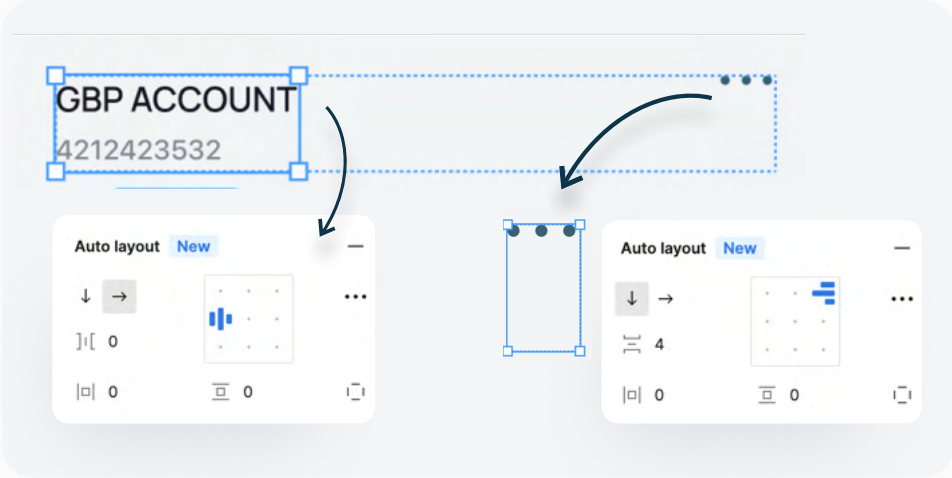
Account balance card



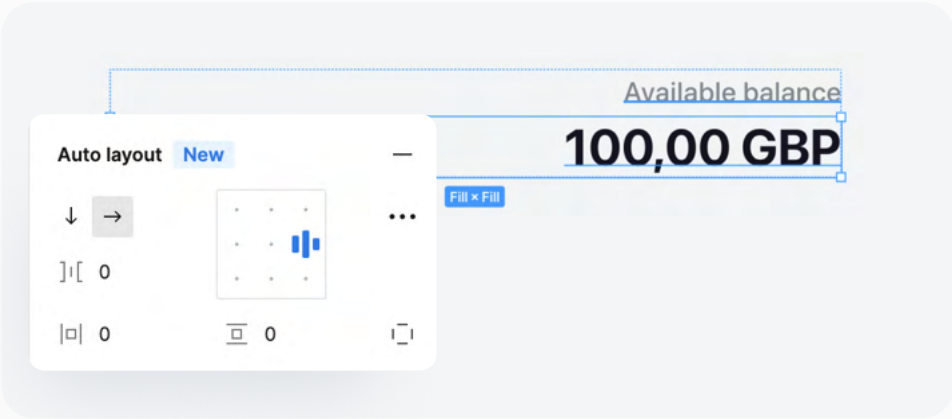
Replicate the layout manually, and then let's start grouping our layers into auto-layout containers. First, select both in our top text group and apply auto layout.



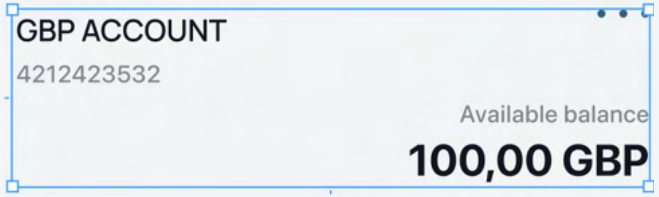
Next up, we need to group our text container with the meatballs menu. Select both and apply auto-layout; make sure your left selection is middle-left aligned, and your meatballs icon is set to fill container and top-right aligned.



Do the same with our bottom section, align text layers to the right, select both and apply auto-layout. Make sure our container is set to align to the right and middle.



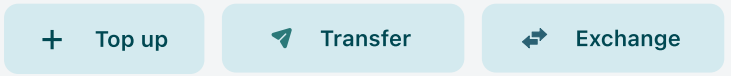
Now select both auto-layout containers and click "Shift + A" again to nest them into another container. Adjust the spacing between them if needed, but overall, they should look fine.



For the next section, we need to create three buttons. You've got three text layers made. Group them up with your icons and apply auto-layout to each of them. Change the padding values to 8px on top and bottom and 12px on both sides. Position your text layer in the middle of your auto-layout frame, and round the corners to 8px.

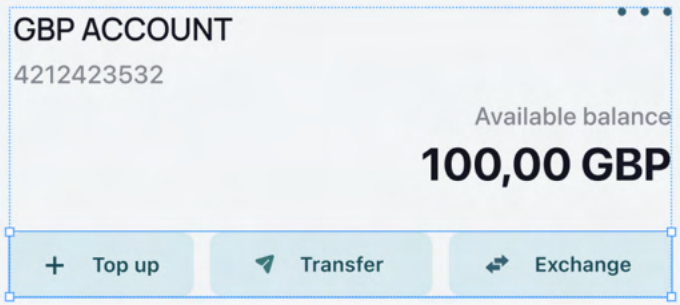


Do the same for all three buttons. Select them all and apply auto-layout to create a new frame. Set the spacing between them to 8px. Make sure all buttons are set to fill container.



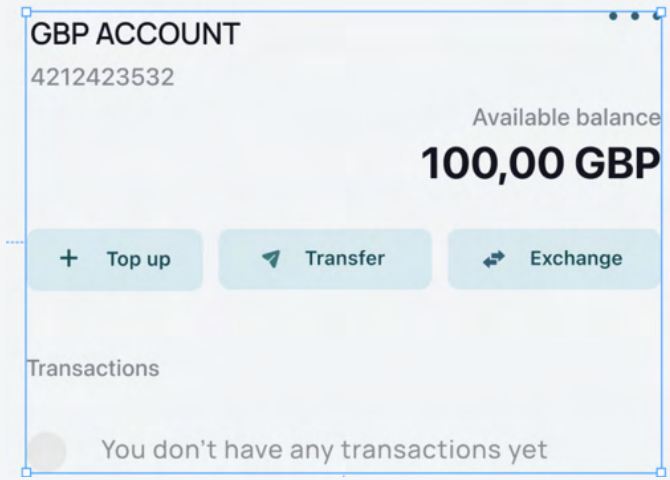
You should end up with something like this.

Now select your text containers from above with the button container and set the position of the auto-layout to the middle. Set the distance between them to 16px.

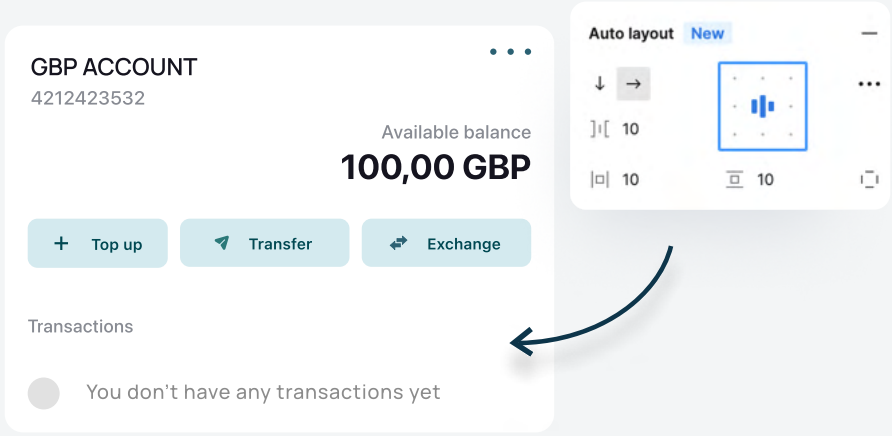


Try to replicate the last part of this container yourself. Look at the reference frame if you need some guidance or copy-paste it and analyze the structure of each selection.

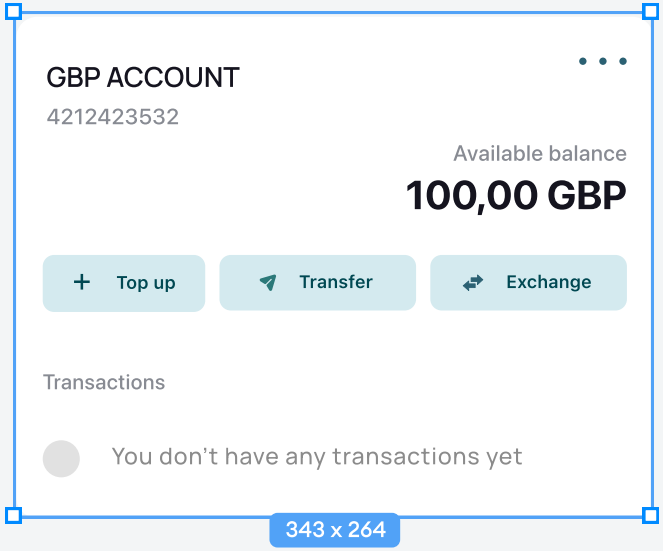
Either way, when you're done, you should end up with a card that looks like this (go to the next page).



We need to apply auto-layout one more time to create our background and add some padding all around our card. With the entire group selected, click “Shift + A.” Change the fill color to white and round the corners to 8 px.



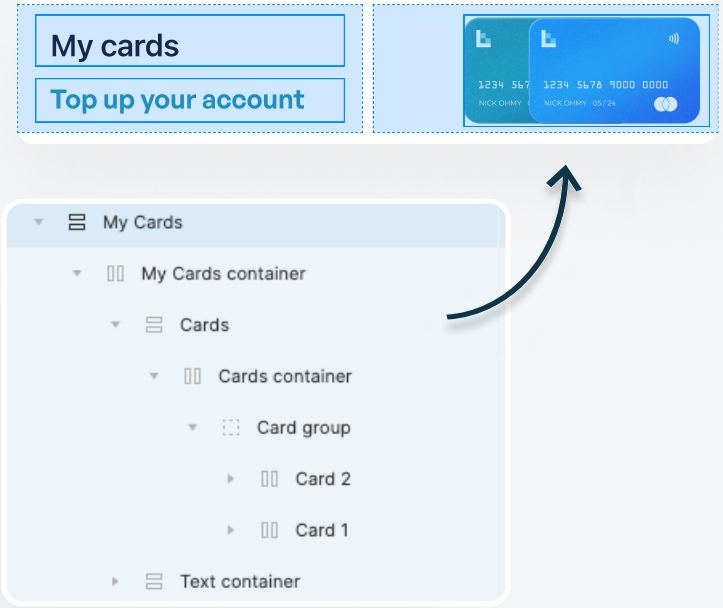
Following our app’s margin, we know that we need to set our container to a width of 343 px (375 frame width - (16 margin *2 sides)).



Make sure every single selection in your auto-layout containers is set to fill-width horizontally and to hug vertically. We’ll set the constraints of “Left and right” and “Top” on our outer-most container since we want it to resize in relation to our top border.

If you’re lost, look at tahe reference frame and check the auto-layout and resizing settings section by section.

My cards card

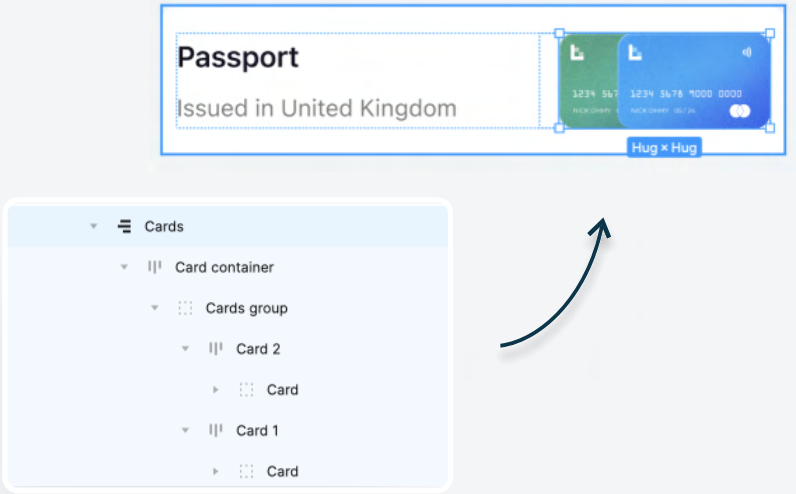


For our “My cards” container, we can easily replicate one of the previous container designs. Let’s use one from the “Choose account type.” We’ll need to remove our chevron first and switch our text and image positions.



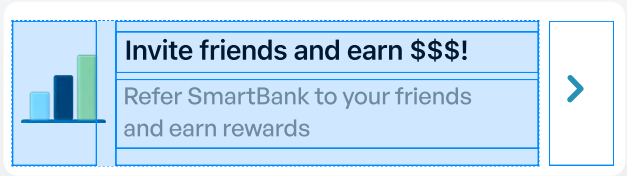
Resize your image frame by selecting it and holding “Cmd/Ctrl” drag one end of the container to fill the entire structure. Change the resizing setting to **fill container** on the **horizontal axis** and **hug** on the **vertical**. Copy the image from our original frame and paste it into our image container on the right.

You can try to replicate the layout of the overlapping cards, but it’s a bit more complicated, so I’d suggest you look into the layers panel and analyze how they are nested.

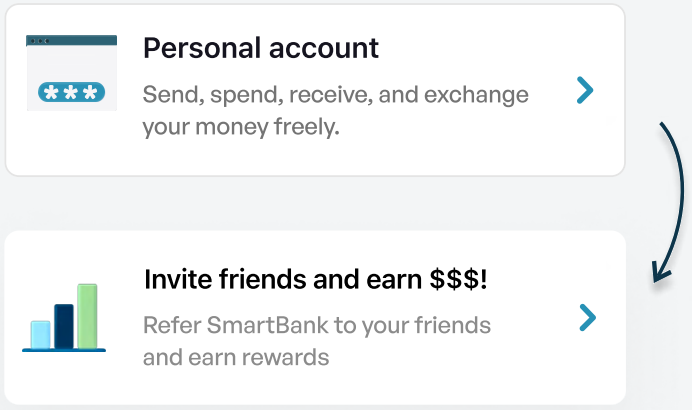


The one thing you’ll have to remember when creating overlapping auto layout design is to group them up first using a regular layer group (Cmd + G) and manually adjusting the spacing. You can also use absolute position on one of the cards to overlap them.

Invite friends card



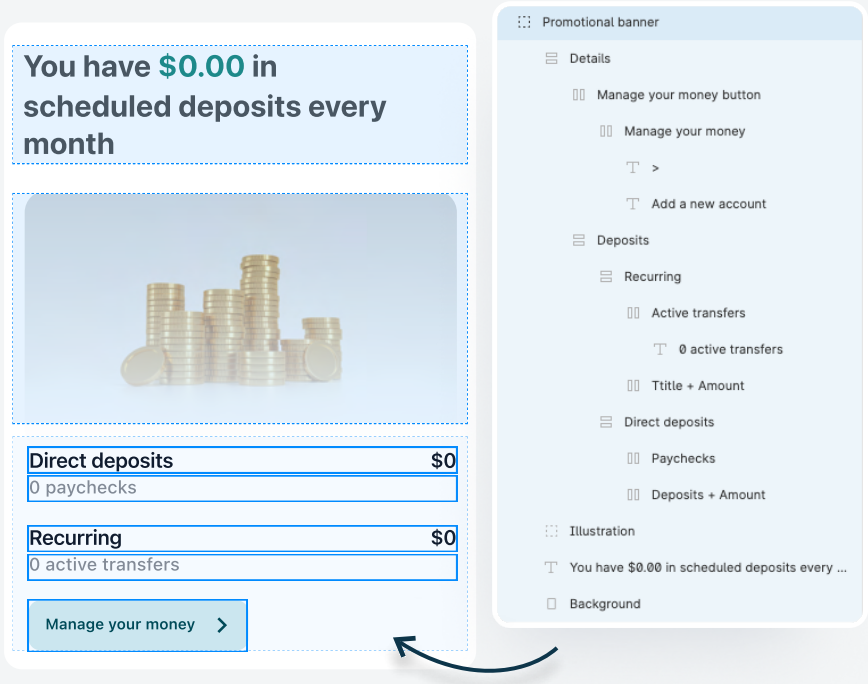
All we need to do is use one of our existing components or copy it directly from our “Choose your account” frame for the next card. Replace the copy and images, adjust the style and we’re ready to move on to our next element.



If you want to have consistency in your design you HAVE to use components. That’s why it’s so important to create components out of every element you might use in the future, even if you won’t use it now, it may come in handy later on. Make it your habit and you’ll already be ahead of 90% of other designers. Besides, your developers, and your design team will love you for it.



Open a balance card

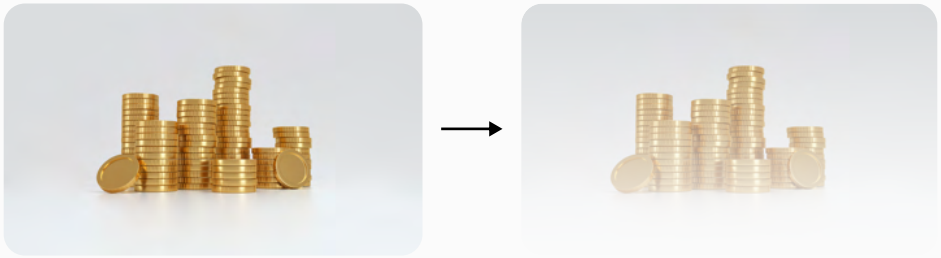


Design our next part the same way we designed the Account balance container. Place your elements around the frame in a similar fashion. It always helps to copy the reference frame either next to your working framework or directly onto it, reducing the opacity and locking the layer in so you don't accidentally move anything around.

Replicate it 1:1 and try to apply the "auto-layout" to each one of the layer groups. If you're not sure how to look at our reference frame, go one by one. I'll leave this one for you. Good luck!

But wait, you might actually need some pointers for the faded-out image. So let's cover that real quick.

Faded Images - mini tutorial



We'll basically need two things to create our faded-out image. An image, obviously, and a rectangle with the exact same dimensions.

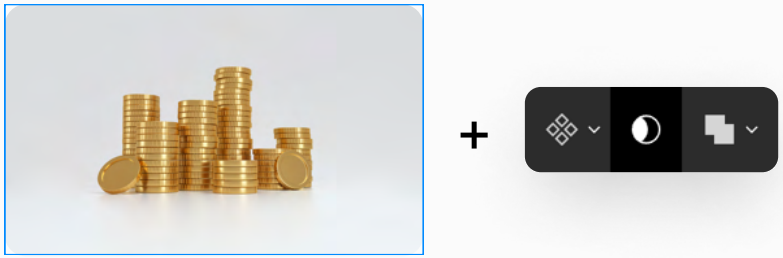
Create our rectangle and apply the exact dimensions. Make it a linear gradient going from 100% opacity to 0% opacity on the bottom.

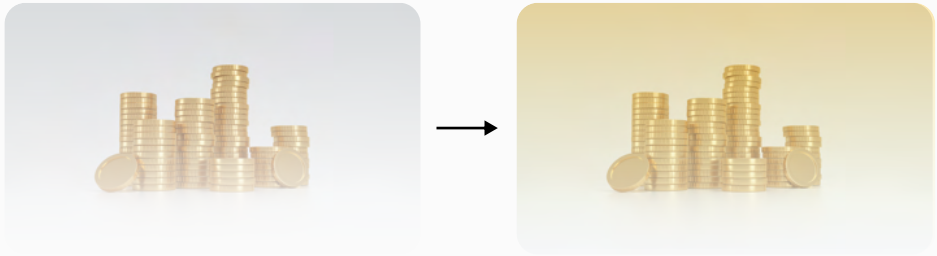


Next up, we'll need to create a mask and mask our image with the gradient we just created. Move our gradient directly behind our image; make sure it's behind the image layer.



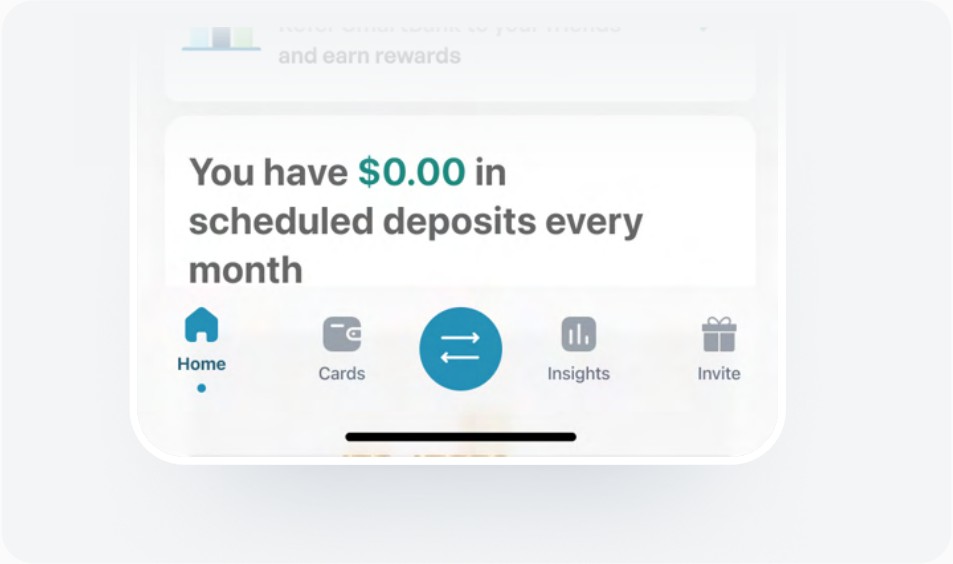
Now select both layers and click "Cmd/Ctrl + Alt/Opt + M," or find the **halfway filled-in circle icon** (mask icon) right above your working area and click on it.





Congrats! We've got our mask now. You can apply similar design patterns using radial gradients, or overlay colorful gradients over your mask to create really interesting compositions.

Tab Bar

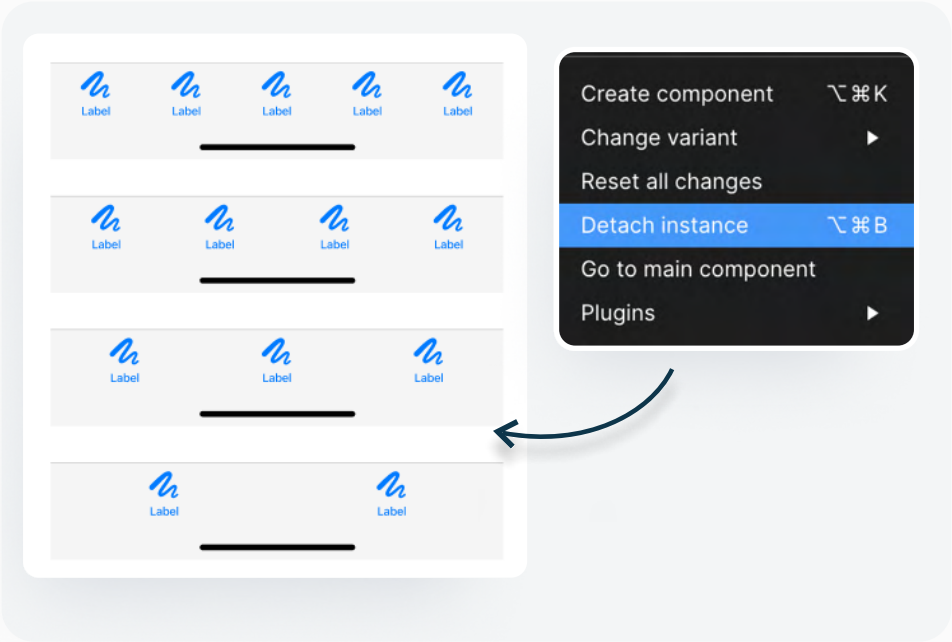


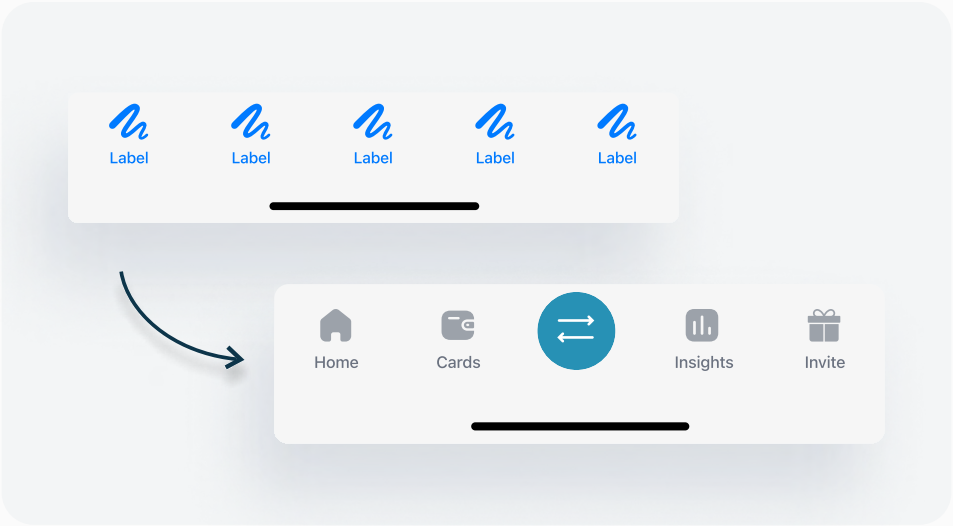
We've finally arrived at the tab bar. An element many people like to overdo and overthink when in reality, it should be simple and, most importantly, readable. Here's how we built it.

When designing my iOS components, I like to refer back to the (un)official iOS UI kit to ensure it follows the same design principles.

We can copy our 5-item tab bar from the iOS UI kit Figma file and start adjusting the design. Before we make any changes remember to always detach the instance and create a new component one more time, so it's now a part of our components library and doesn't link out to a separate file.

Change colors, update text on the labels, and replace our icons. We'll also have to create an active state menu item. We'll use bolder text and a circle indicator to give an additional visual cue for users with visual impairment.





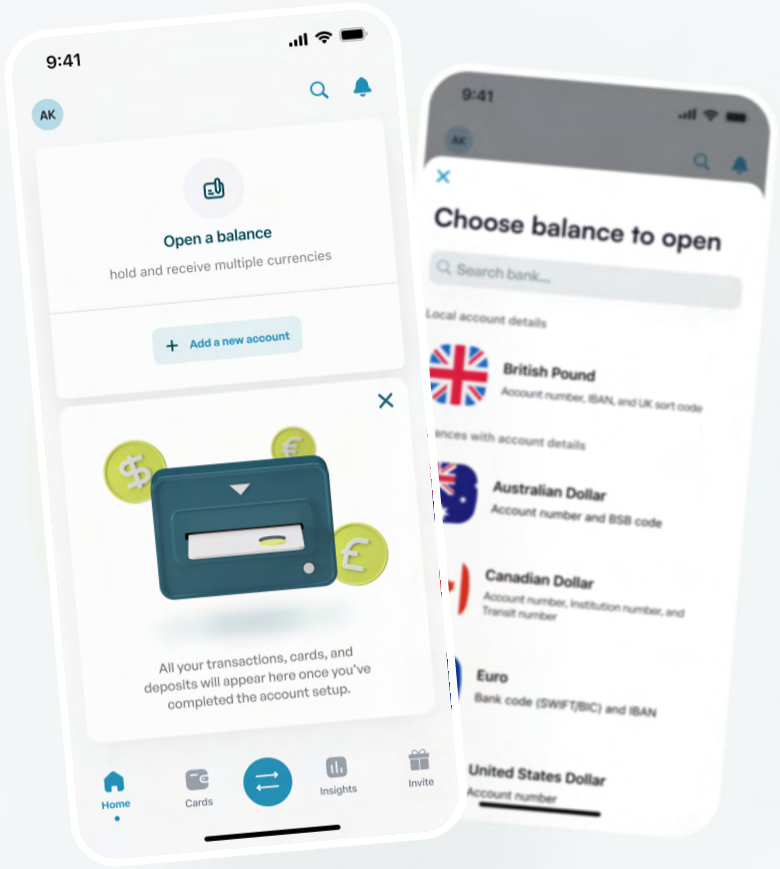
For our middle item we'll need to replace the entire container rather than separate icons and labels so let's do that using our "Shift + Cmd/ ctrl + opt/alt + V" shortcut.

I would also create separate components out of our menu items and design states for both Active and Inactive pages.

Then I would replace the static containers with new components and make a component out of the entire tab bar. With that created I would add variants for each active menu item so we can easily switch back and forth between different states. You can actually play around with the tab bar component in our Components page. But explaining that is a topic for an entire chapter. Stay tuned, I will release a new update soon!

Chapter recap

- When designing your product, first focus on the core functionality for your "first release", and build gradually outwards from there.
- This ebook is like a ver 1.0 of a fintech product and new updates with new supplementary flows can be treated as new app version releases.
- Top navigation bar provides content and actions related to the current screen
- Offer referral and loyalty programs to get new customers through one of the most powerful marketing methods - word of mouth.
- By default, tab bar displays up to 5 tabs with the last one being "..." or "more". If tab bar holds more than 5 controllers consider hiding it in the more list.
- If you want to have consistency in your design you HAVE to use components.
- When designing my iOS components I like to refer back to the (un)official iOS UI kit to make sure it follows the same design principles.



Open a balance

THE INTRO

Open a balance - first login

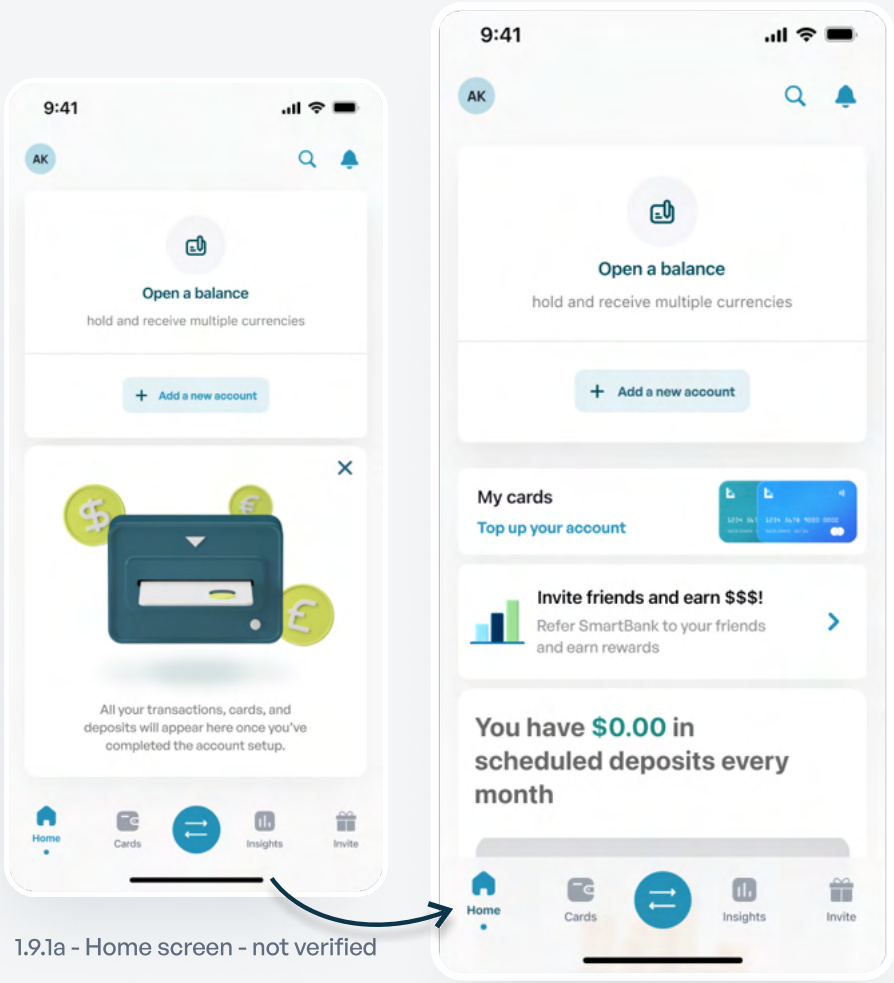
If we skip the entire setup wizard and go back to the app, we'll be greeted by a home screen with much-limited functionality. We don't have an account, so our first order of priority will be to show an appropriate interface card with a CTA to open a new balance.

We also don't have any cards yet, so we'll need to show an action card with a CTA to top up the account.

Our main goal is to keep users engaged and carry out the actions that this app was designed for - receiving, sending, and exchanging money. We need a credit card and an account with a balance to do that successfully.

The rest of the home screen elements can stay as they are. Still, we might also design a demo screen with limited functionality if we have users who have not submitted all their data or are not verified to explore or complete specific flows within our app.

We want to make the whole app experience centered around our users, so our job is to suggest the quickest ways possible to complete specific actions and make it as easy and effortless for them to navigate the app as possible.

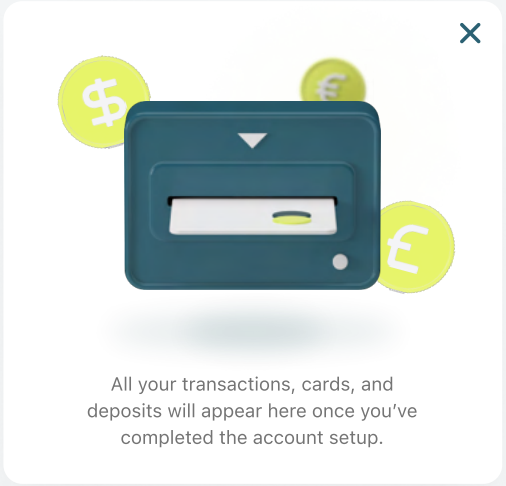


1.9.1a - Home screen - not verified

1.9.1b - Home screen - no accounts

First login - limited functionality

This box would show all the users who haven't yet finished the account setup or haven't been approved by our system. This usually shouldn't happen as the verification process is mainly automatic and takes 15 minutes. Anyway, we need to design it just in case.

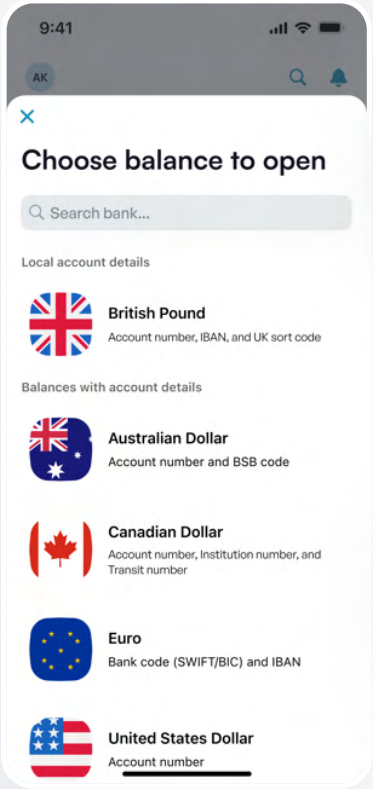


Offering a limited app functionality is a good incentive to show your users what the experience could look like if we were to submit all our details and continue the onboarding setup.

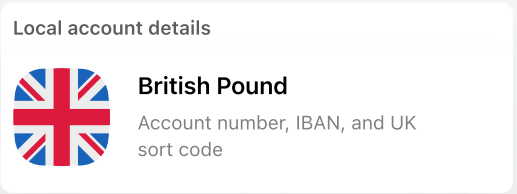


Choose a balance to open

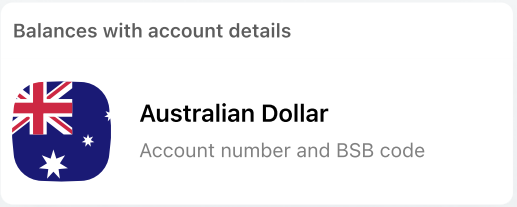
We clicked on the "Add a new account" CTA, and a slide-out card appeared on our screen. Seems pretty simple and intuitive, right? We have a close button to click out and cancel the action, a clear title, a search input, and a list of different currency accounts.



1.9.2 - Choose a balance to open

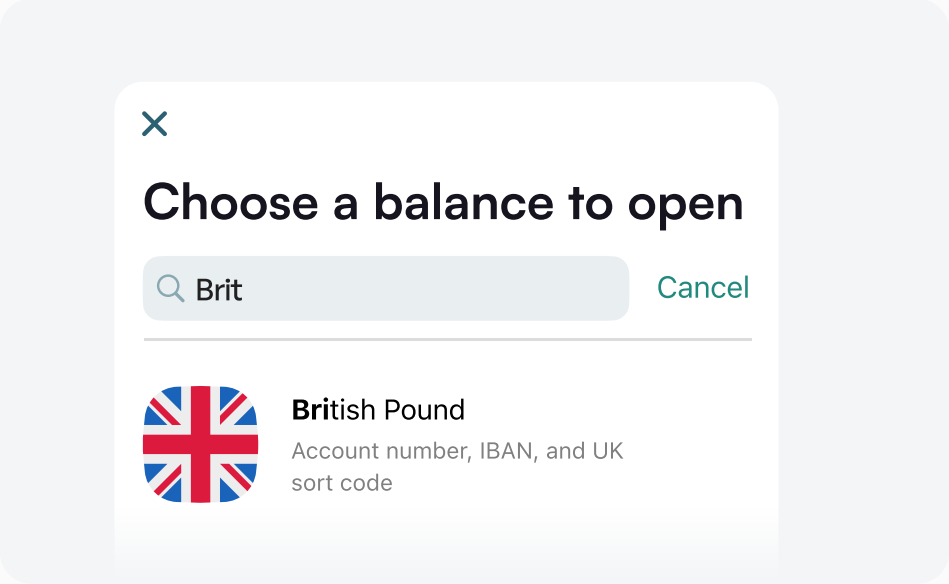


First up, we need to choose a currency. Our system will suggest your most likely option which is your local currency.



And then the balances with the most popular currencies will follow after that.

Provide the user with an option of selecting a different currency. The coins are listed alphabetically with a flag on the left side for easier recognition. Instead of scrolling, the user may directly type in the currency he is looking for.



As we mentioned before, offering a manual search input is helpful for lists of 6+ items. We offer close to 20 different currency accounts

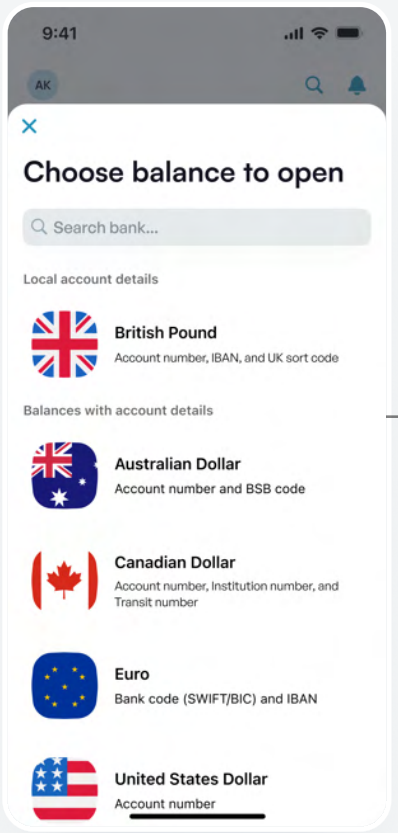
It’s essential to have these search results start appearing as we continue typing keywords.

We’ve already designed multiple search input states, but there are never too many screens when you want to show all the micro-interactions to your dev, so let’s design that.

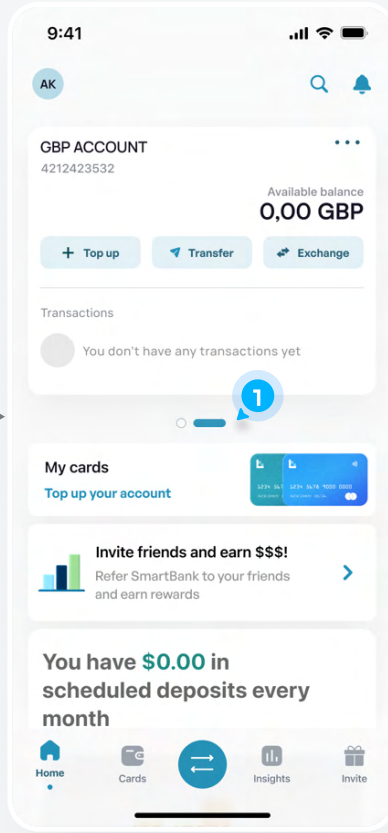
New account added

After choosing a specific currency, a new balance will be automatically created. We’re making it instant to show how easy it is to create new accounts once you’re all set up and verified.

You might have noticed an addition of the **1) slider indication** right below our new account. What is it, and why it’s here, you might ask?



1.9.2 - Choose a balance to open

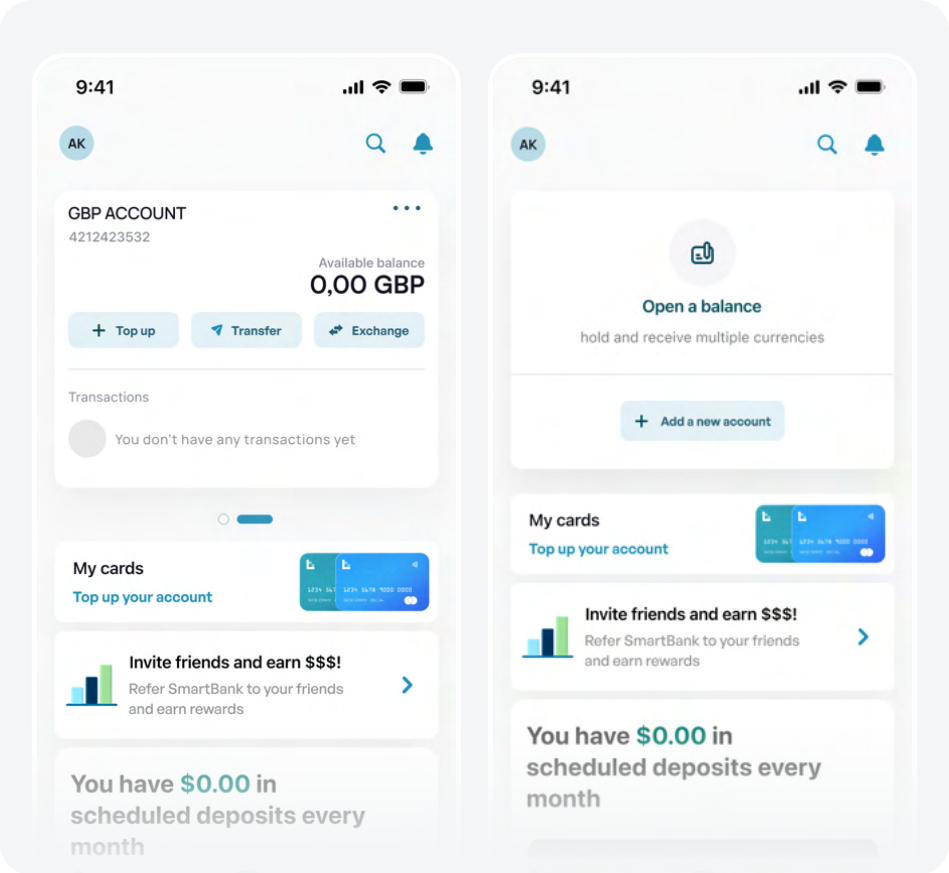


1.9.1c - Home screen - new account

Slider navigation

The function of slider navigation is to indicate something hidden under a swipe motion - which in our example, shows a card with open a new balance action.

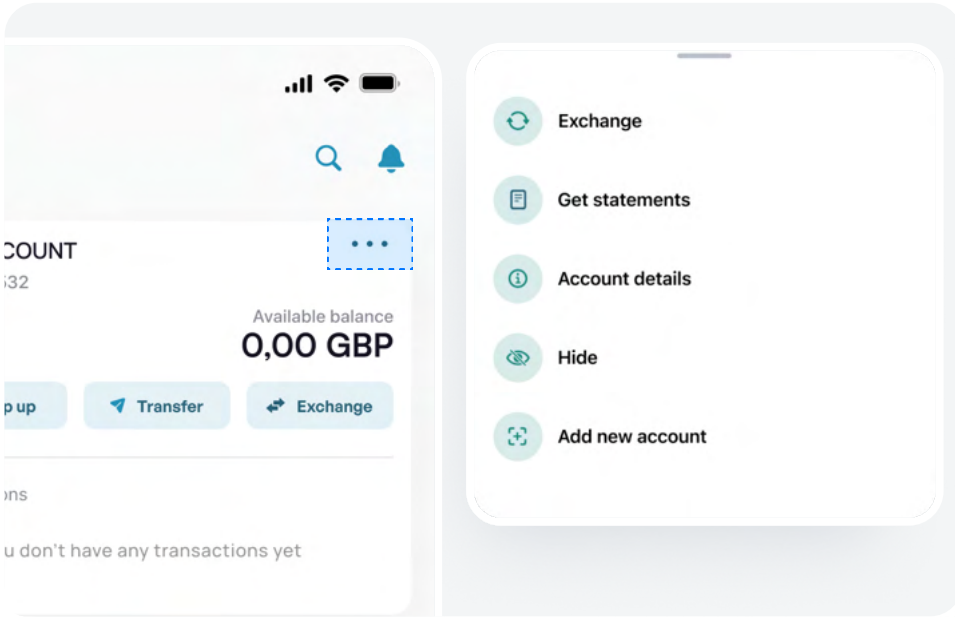
We've also put a meatballs menu opposite our account name to pull up an iOS card modal showing additional options like adding a new account, downloading statements, or getting account details.



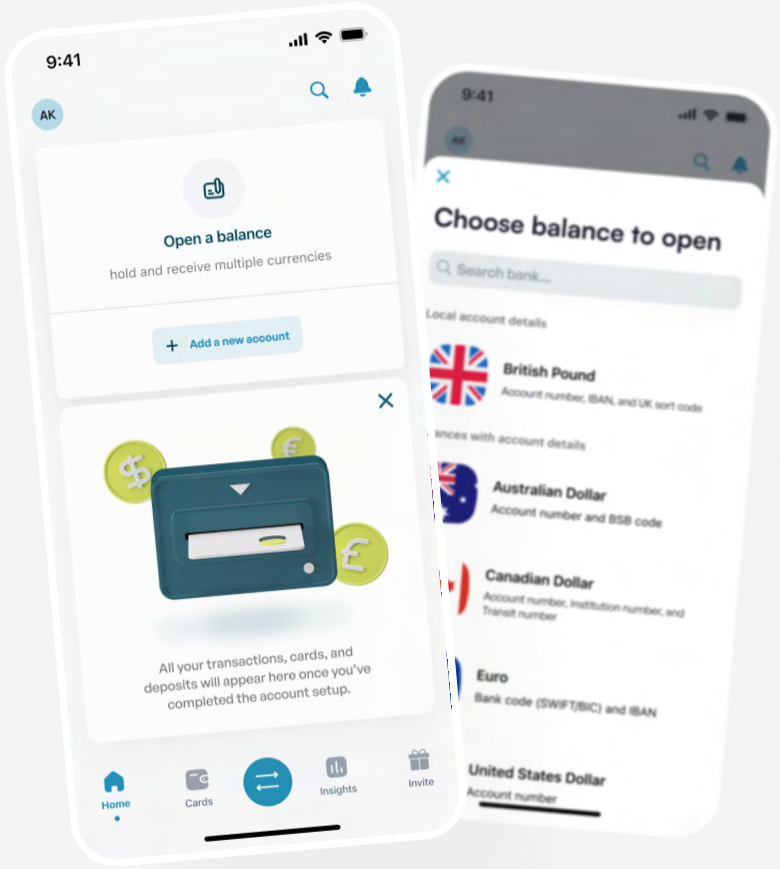
Additional account buttons

If you look closely at the top right of your account container, you'll see three dots - that's usually where designers hide additional items related to a particular component. In our example, these are - exchange money, get statements, account details, an option to hide/ show the account balance, and an option to add a new account.

Apart from adding a new account straight from the metaballs menu (a common name for the three-dots icon), we can also do so by sliding the account card to the right and unfolding a new card.

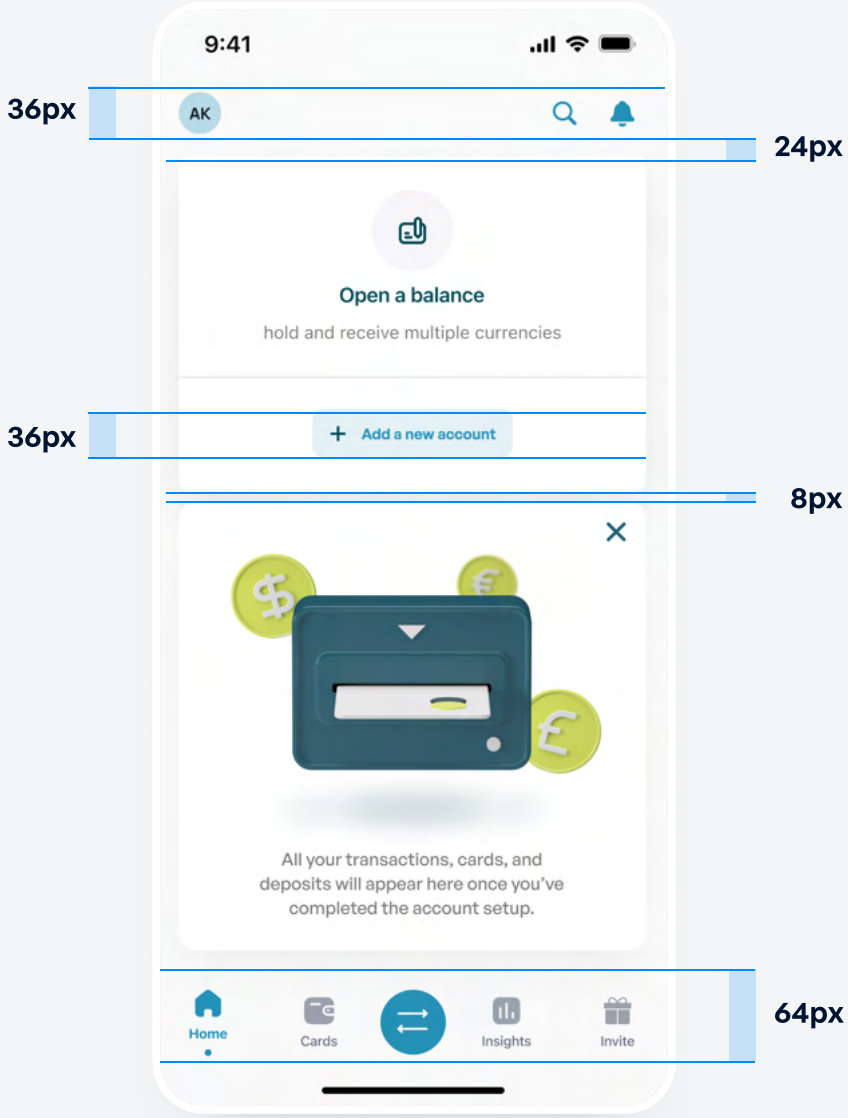


A sheet appears as a card that partially covers the underlying content. The card's top corners are rounded to visually distinguish it from the parent view.



Open a balance

THE STRUCTURE



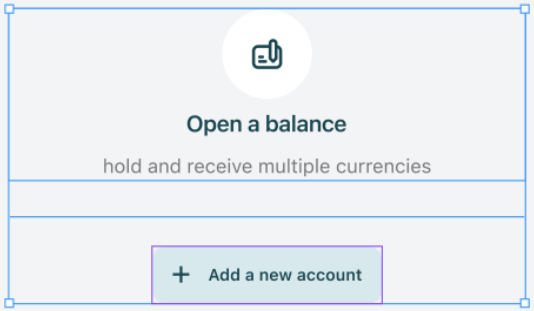
1.9.1a - Home screen - not verified

Open Balance - the structure

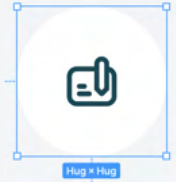
Structurally, this screen is very similar to the previous Open balance screen we've covered. It's a bit easier to design since it doesn't contain that many elements and to recreate it we only need to design 4 elements and group them up.

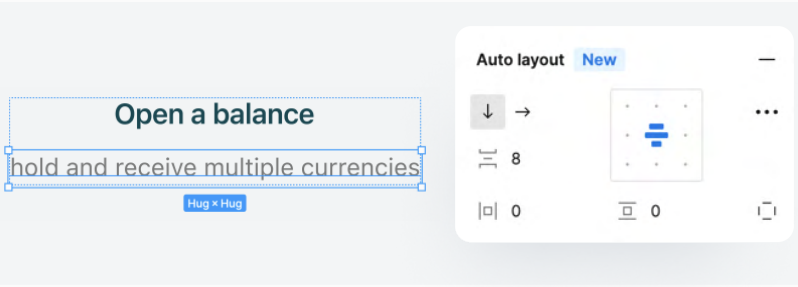
Open a balance card

Let's break down the elements we need to design in order to replicate our layout.

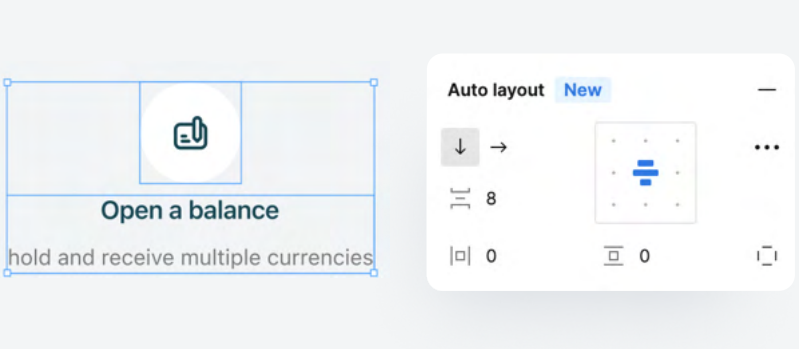


For our container, we'll need to create an icon, add a title and a description, create a divider line, and add an auto layout button (you can copy one from previous screens).

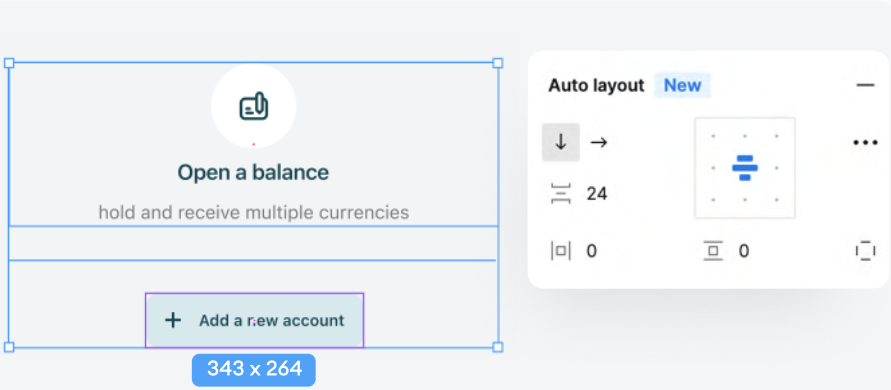




Select both layers and apply auto-layout. Change the spacing between them to 8 px and center align it.

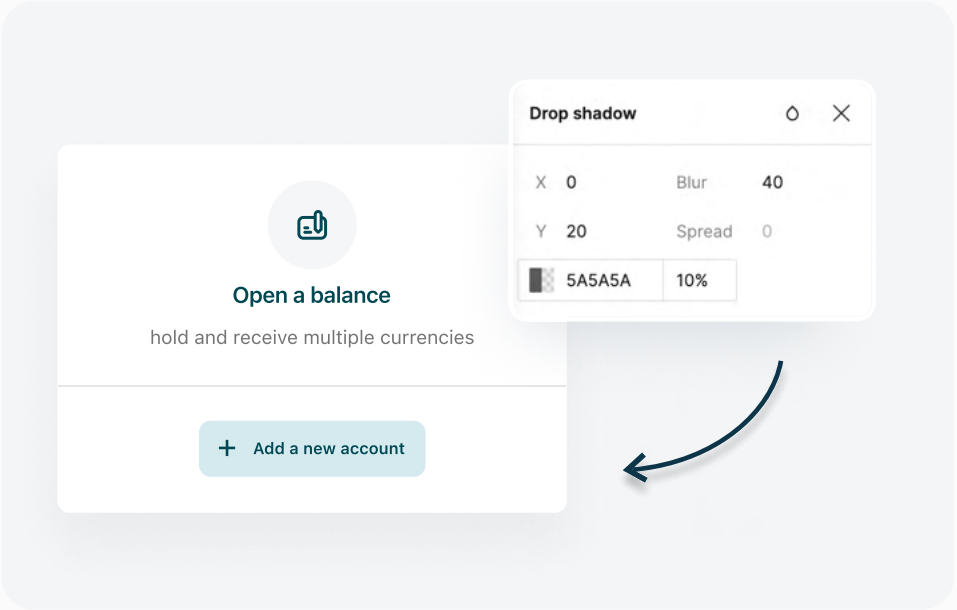


Next, select your icon and auto-layout text group, apply auto-layout, set spacing to 8 px and center-align it.



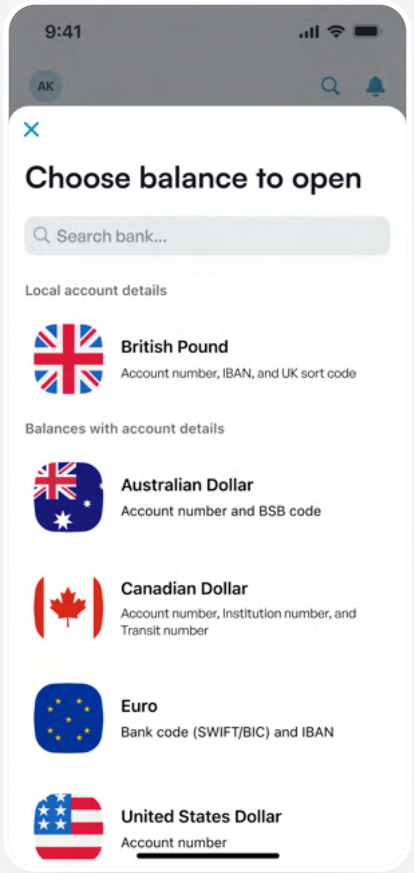
Lastly, select your new container, the divider, and your CTA button and apply auto-layout. Set the distance between these elements to 24 px and center align everything.

Apply auto-layout again, add 10 px all around and fill it with white color. Add a shadow to our box. Use the following settings.



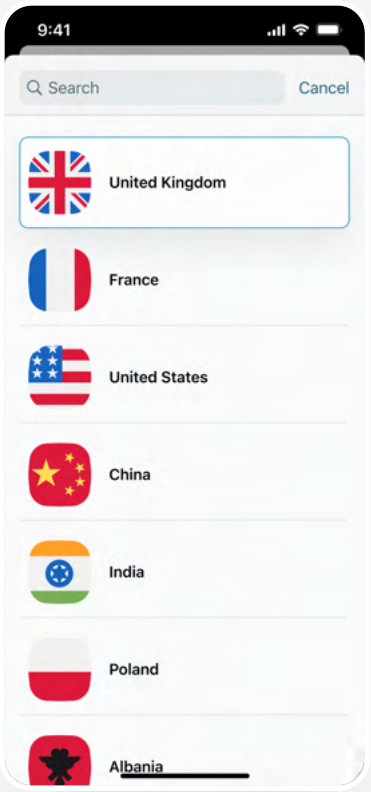
If in doubt, as always, refer back to the original frame you see on your left and analyse every single selection and its settings. It really is a trial and error kind of practice until you start getting it right. And you will get it right in no time.

Choose a balance

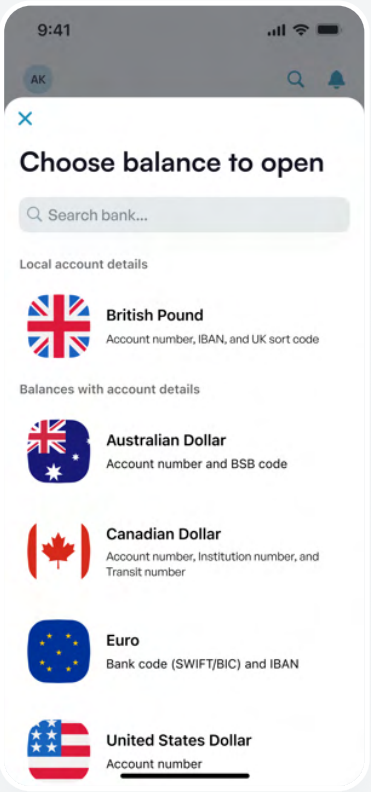


1.9.2 - Choose a balance to open

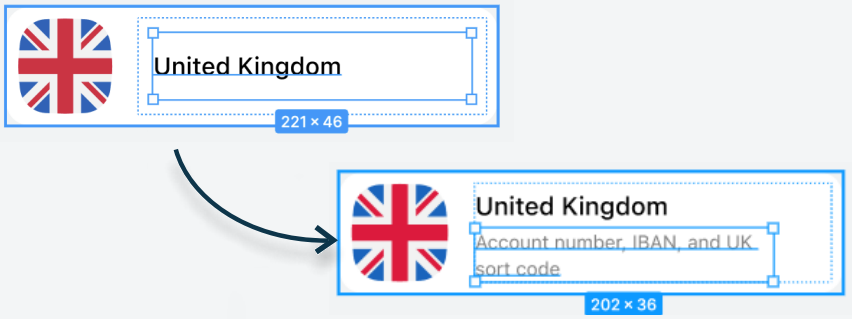
For our next screen let's copy the section directly from our "1.3.4a - Select Country 1-2" frame. We'll only need to add another text layer for the description and add a title above our list.



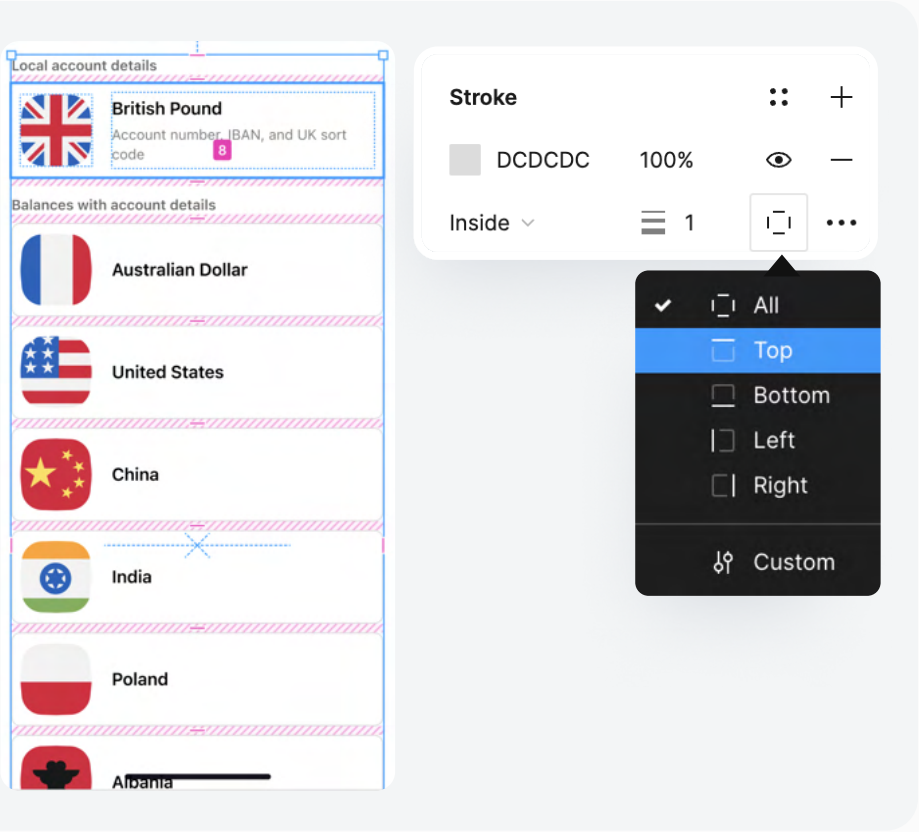
1.3.4a - Select Country 1-2



1.9.2 - Choose a balance to open

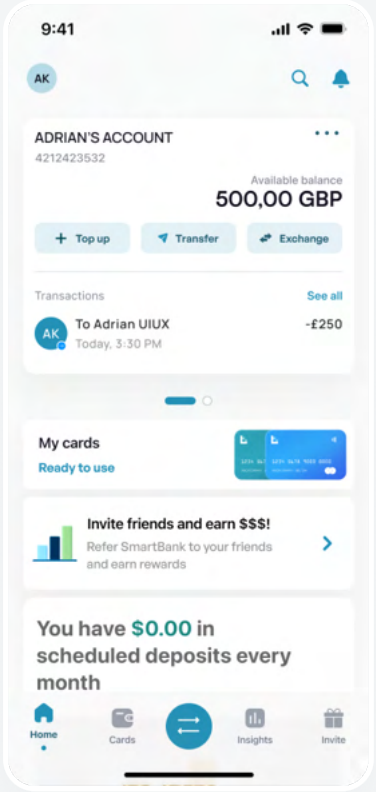


We'll also need to adjust the style a little bit. Select all layers within the auto-layout selection and change the border of all cards to the top one only. Set the width to one and color to "#DCDCDC."

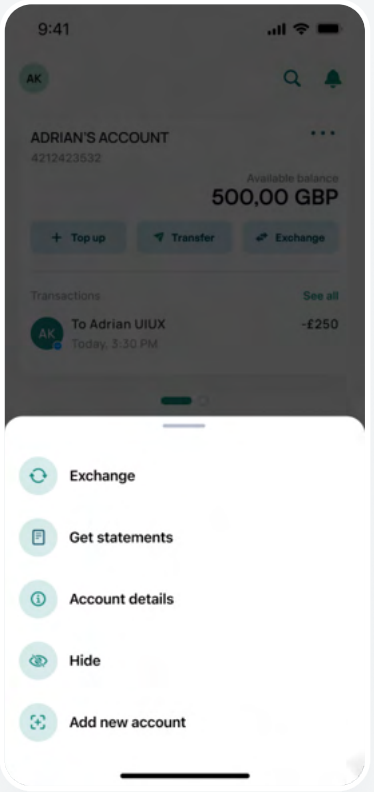


This whole layout is quite complicated though so make sure you check every setting on the reference frame. It's always best to adjust one auto-layout selection and then duplicate it to replace the copy and images instead of readjusting all selections separately.

Meatballs menu popup



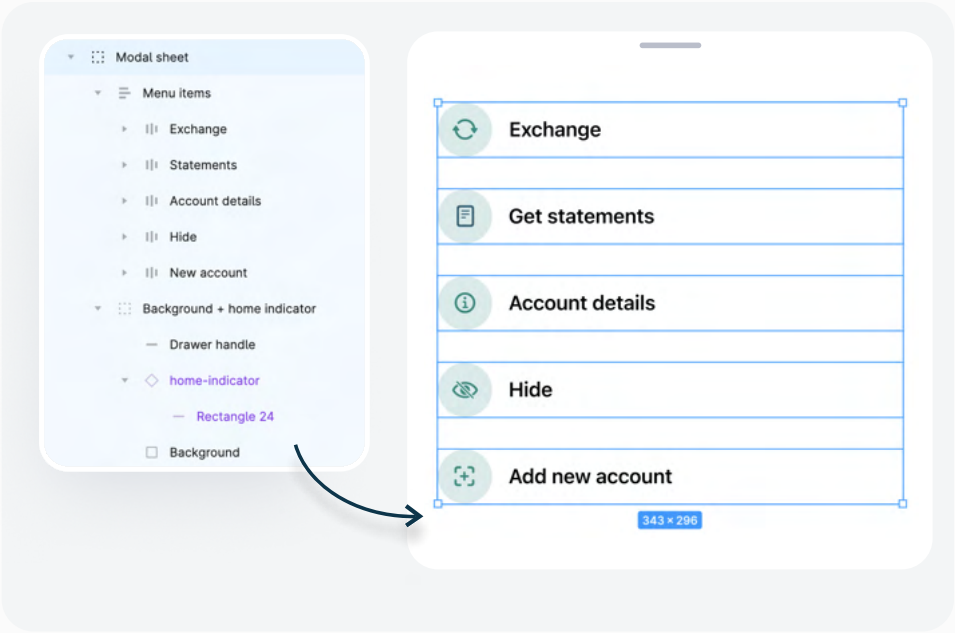
1.9.1e - Home screen + Transaction



1.9.1f - Home screen - extra actions

To create our modal sheet all we need to do is create a rectangle, add a background overlay in between the modal and the home screen, and a list of menu items.

To create our modal sheet background, simply add a rectangle with these dimensions: 375 px wide x 384 px tall. Change fill color to white.



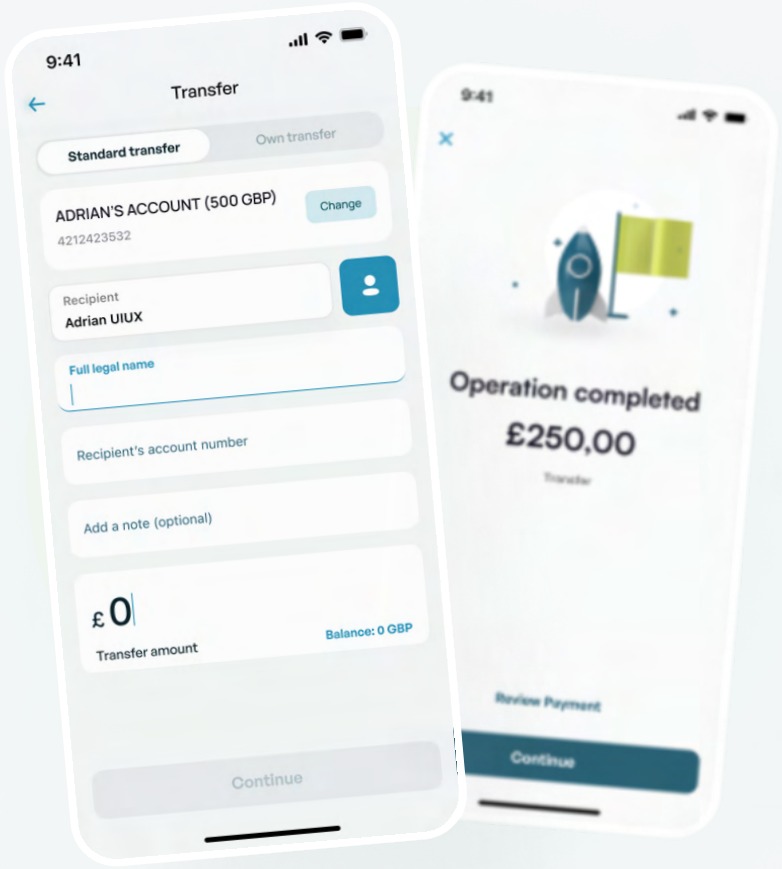
Replicate the menu items by modifying the "Verify your identity frame." You'll need to replace checkmarks with the menu icons that you can find on the components page.

Make sure your Menu items resizing properties are set to "Fill" on a horizontal axis and "Hug" on a vertical axis, and that your "Menu items" container is set to a fixed value of 343 px and the Constraints are set to "Left and Right," and "Bottom."

When we change our frame size, the container will be restricted on both sides to a 16 px margin, and with our child menu items being set to fill container, they will fill out the entire auto layout frame. You'll get this with enough practice, so don't worry!

Chapter recap

- We want to make the whole app experience centered around our users. Our job is to suggest the quickest ways to complete specific actions and make it easy and effortless for them to navigate the app.
- Offering a limited app functionality (before the paywall) is a good incentive to show your users what the experience could look like if we were to submit all our details and continue the onboarding setup.
- It’s essential to have the search results start appearing as we continue typing key string queries.
- The three dots icon is casually called meatballs and used to hide additional items related to a particular component or an element.
- The function of slider navigation is to indicate that there’s something hidden under a swipe motion.
- If in doubt, as always, refer back to the original frame you see on your left and analyze every single selection and its settings.
- It’s always best to adjust one auto-layout selection and duplicate it to replace the copy and images instead of readjusting all elements separately.



Money transfer

THE INTRO

Money transfers

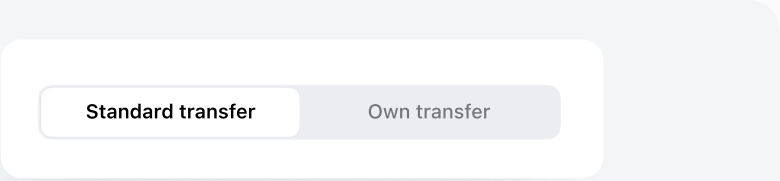
Let's discuss our core functionality now - transferring money. Without further ado, let’s break down our frame first.

1. Title

There's not much to explain here except that we’ve switched from our standard H1 title container to a centered label design. Why? To save space and drive more focus toward the transaction settings.

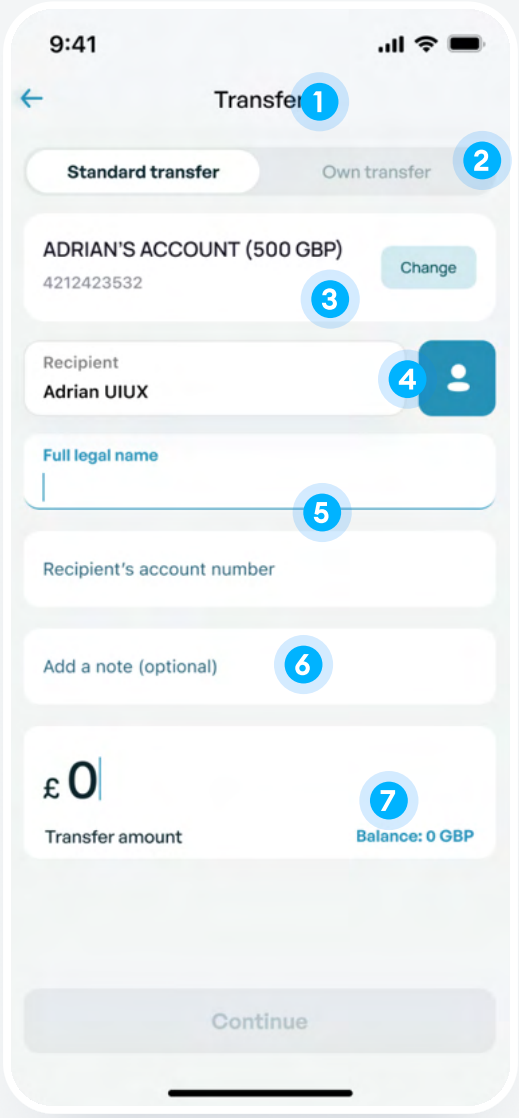
2. Segmented controls

We’ll need a way to switch between transfer types, whether it’s an outgoing transfer or internal between our accounts.

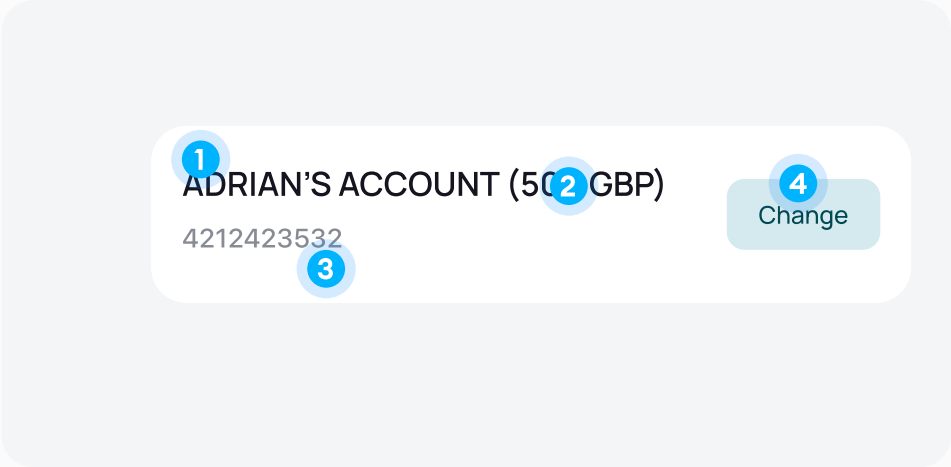


3. Active account card

Let’s dig a bit deeper and break down our active account card. What kind of information do we need to display?



1.10.1.1a - Standard transfer



1. Account name

This would not be visible for single-owned accounts. We want to show this in case people make a mistake or want to switch out to a different account if they decide to do so.

2. Account balance

Let's also show the available balance, we don't want to overdraft our account.

3. Account number

For 100 % transparency and to avoid mistakes we want to show the account number too, but let's be real, you wouldn't probably remember your account number anyway.

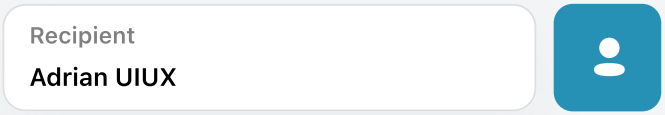
4. Action button

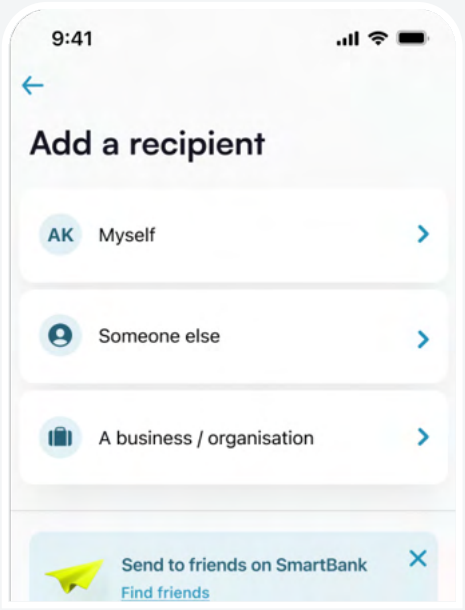
We've added a chevron here to indicate that clicking on this card will start a new flow.

4. Recipient

Next up we have our recipients field that acts as a search input. Whenever we start typing our recipient we'll see a list of recipient with a similar name start appearing in a dropdown list.

We've also added a button showing a complete recipient list whenever clicked.



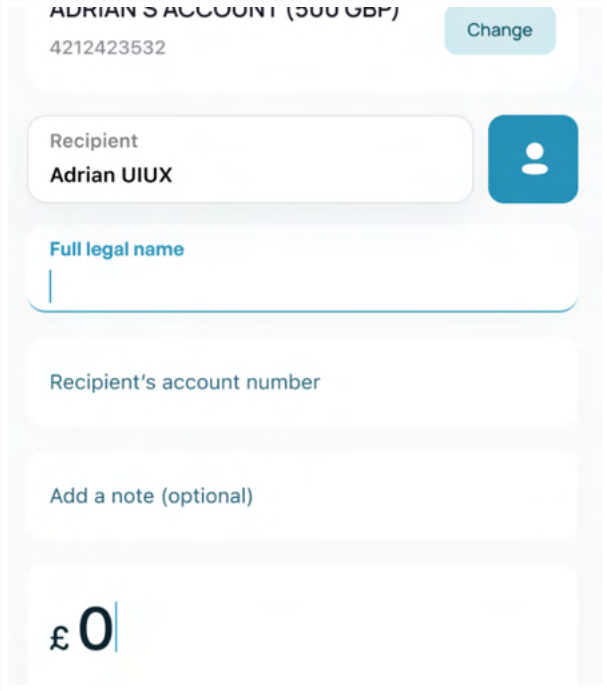


When our list of recipients is empty, clicking on the User Icon would take us to a new screen where we'd be able to add new recipients without breaking the transfer flow. As with all standard transfer we should segment our recipients into our own accounts, other people, and businesses we might send or receive transfers from at any point in the future.

With users already added to the list we would see them in order of most to less frequently interacted with.

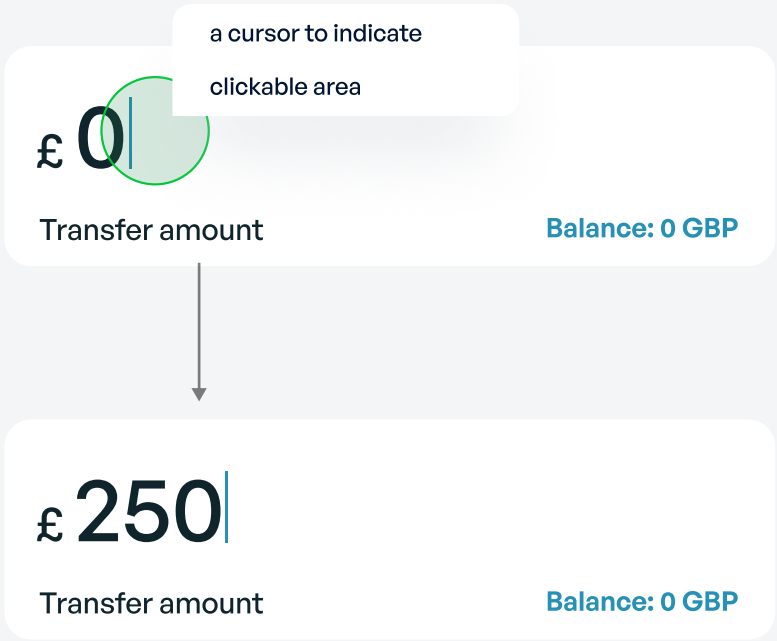
5. Transfer details

The transfer details include the basic details every bank requires during a transfer money process. By deafult you need to provide the recipients legal name and account number, adding note is optional.



After submitting all input information, our system will run a quick background check to see if the provided data is valid. If so, no errors will show and we'll be able to move to the next part which is the transfer amount.

6. Transfer amount



Moving on to our next input -> Transfer amount. We need to add a few things here. An indicator to indicate a clickable area, and a balance left that reacts to the amount we type in.

We could have probably designed this flow in a number of ways but let's say we tested it internally and decided we want to start simple and iterate from here.

7. Transfer date

There are situations where we need to plan out our transfers to be sent out on any specific day in the future. This is why we've added this last part.

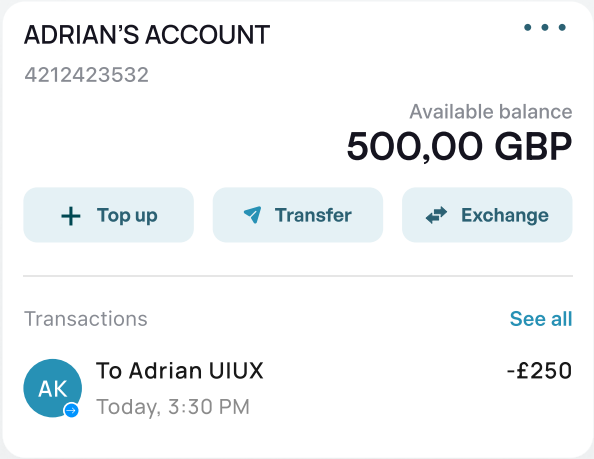
We could have also added another variable like type of tranfer, whether it's a recurring, or one-time, instant, or regular etc. but for this release we'll focus on the most basic layout design.

We want our design evolve along with this ebook to show you how, potentially, a real-life app project could look like. This way we are able to test different solutions and offer new ones in flows where users tend to drop off or are not interacting with at a satisfactory level.

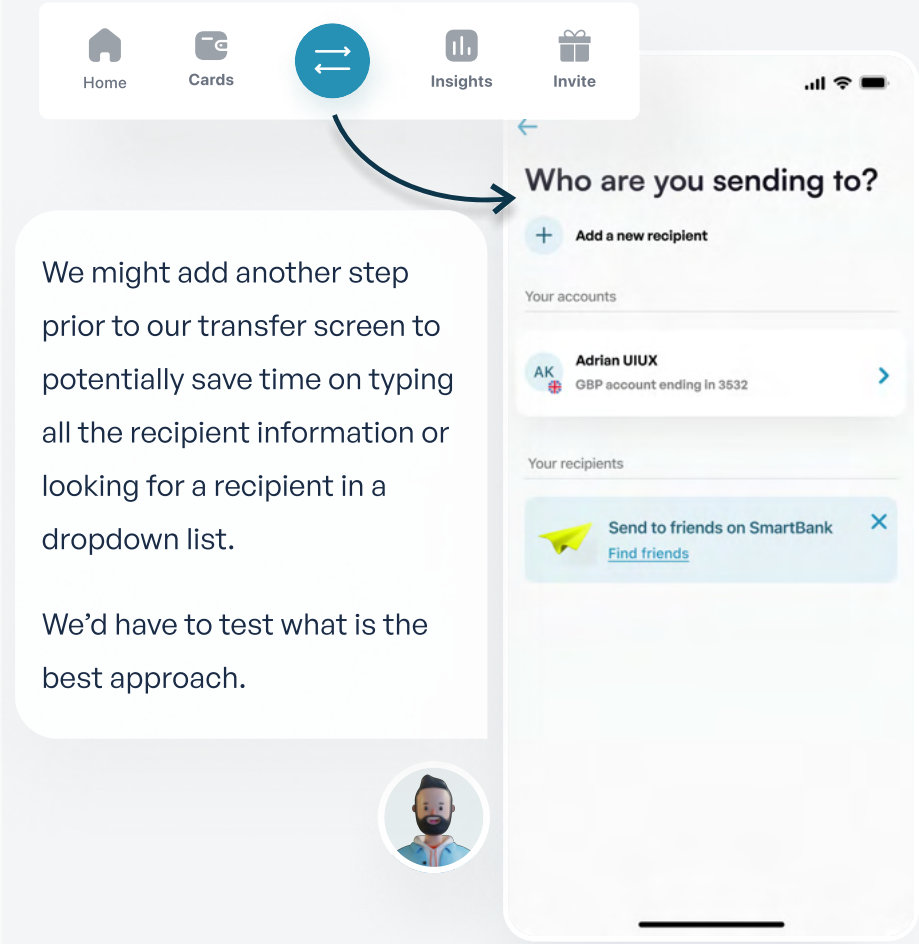
How to access transfers?

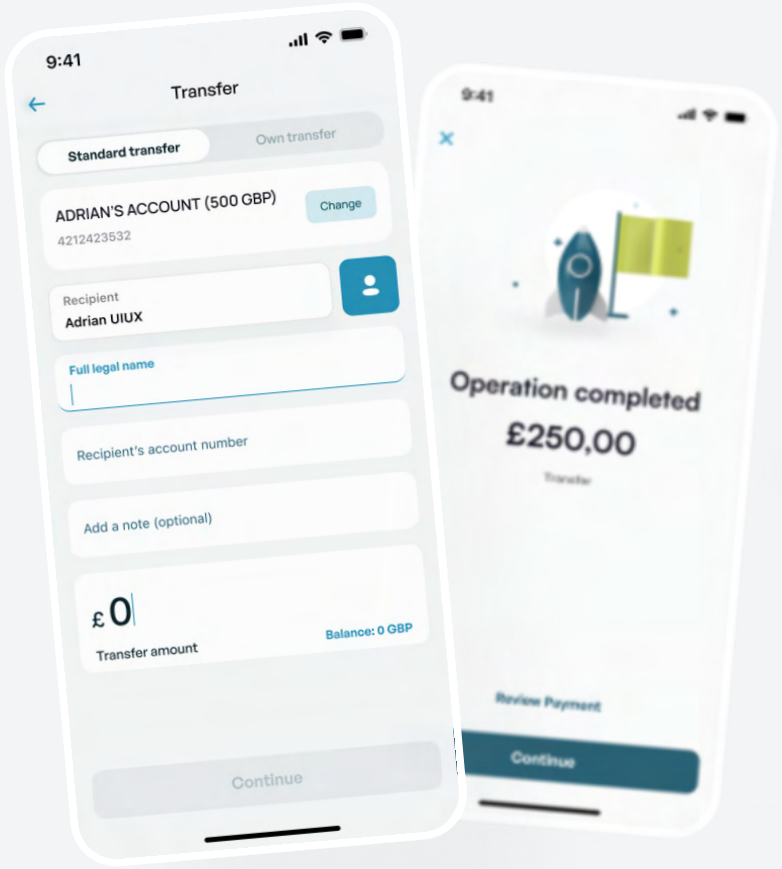
We can access it either by clicking on our action button on our account balance card or by clicking on our tab bar menu item right in the bottom center of the frame.





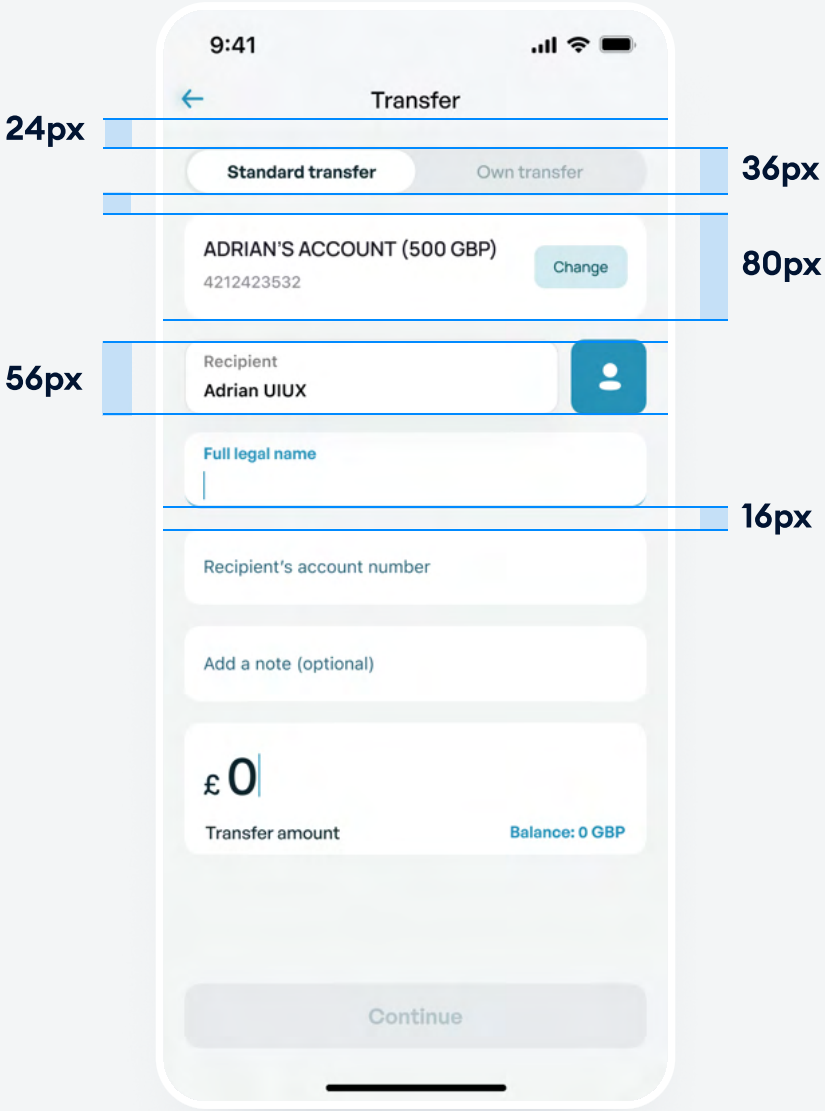
In a real-life example we would of course test a lot more iterations of each flow. We might have ended up with something that's completely different. It's all about testing.





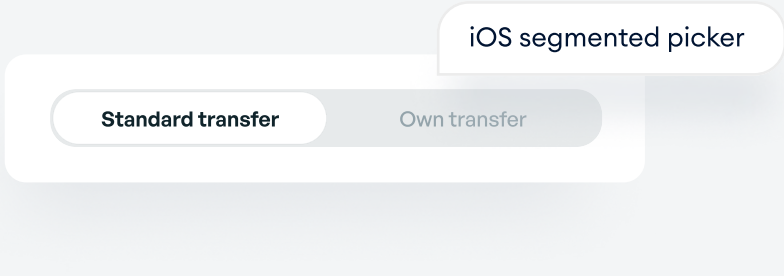
Money transfer

THE STRUCTURE



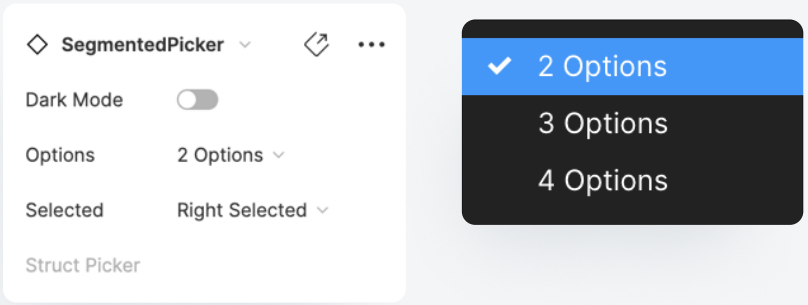
1.10.1.1a - Standard transfer

Transfer money - structure



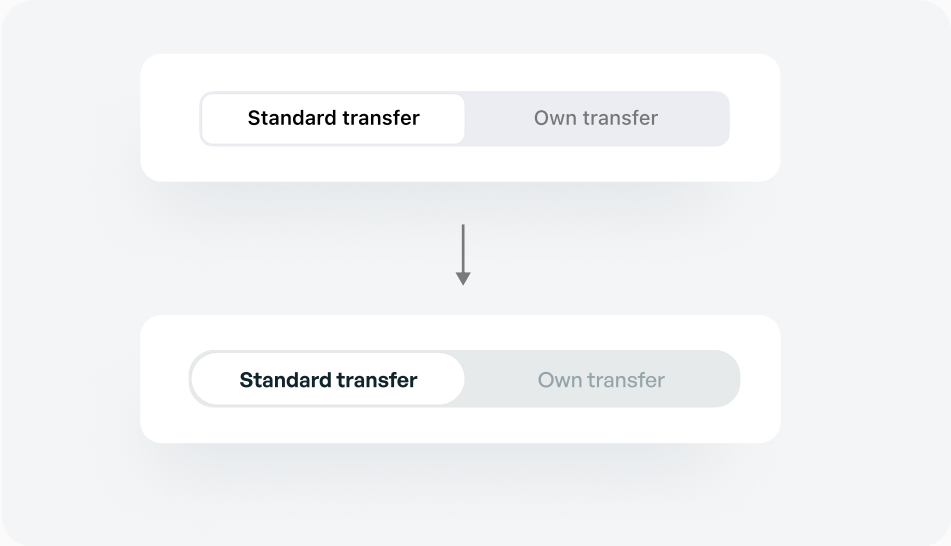
Label container and segmented picker

For the top part, we can copy our label from our “Account details” frame. For the segmented control, use the one from “Order a card” frame. Position it 24 px from the top label.



Reminder: If you ever need to change the settings of your controls, look at the right property panel. This is where all your component’s variable properties are.



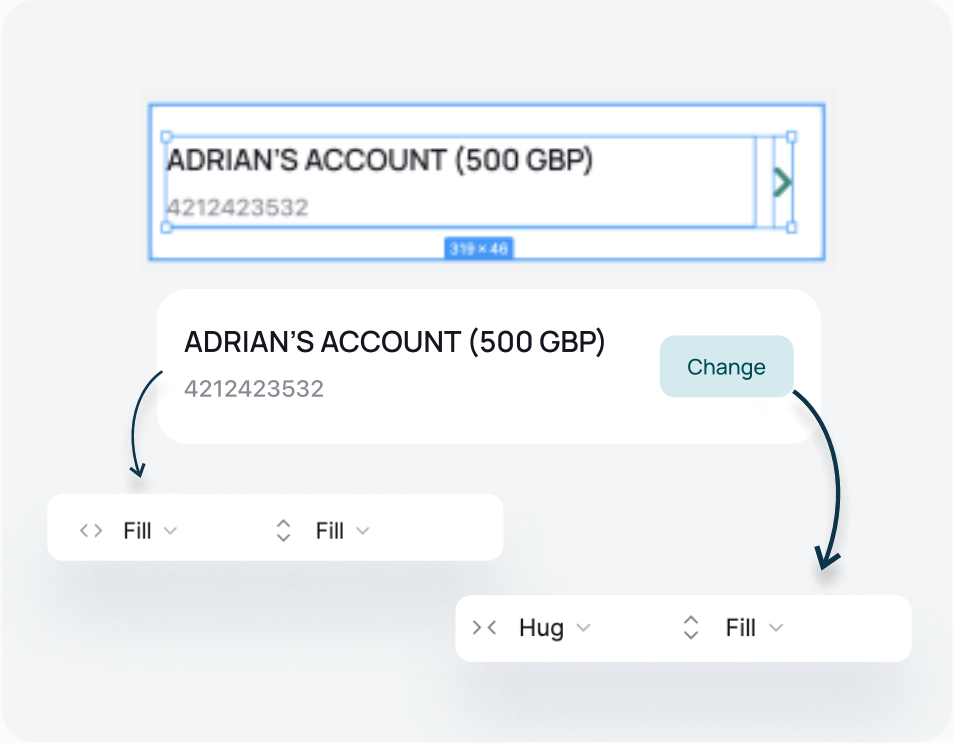


When copying our components remember to apply our accent colors to highlight the active tab state (if they're not applied).

Transfer from

We've already designed many similar card layouts so by now we should have one, at least very similar, in our Components page.

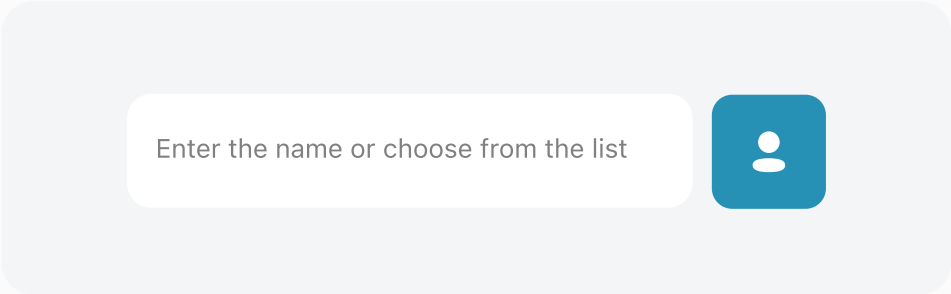
You can of course reuse a container we've created for a different frame like the one on "Choose a balance to open" or "Verify your identity".



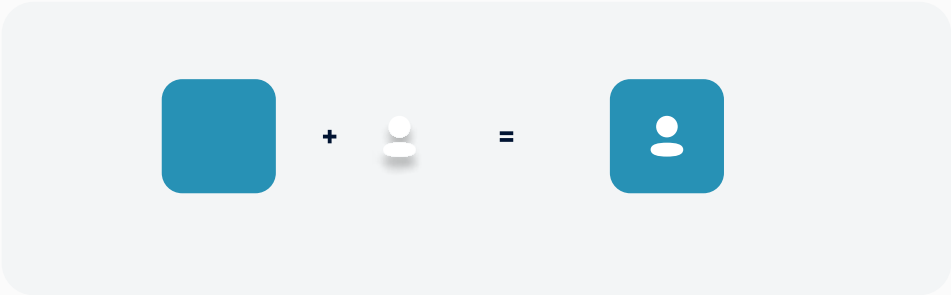
Whatever option you choose, make sure that after you adjust the style of these cards you look at their rezising properties to see if they'll remain responsive when we start adjusting the frame size.

I deliberately skip on repeating the same things over and over again, but you should keep in mind the rezising properties and your constraint settings at all times!

Recipient information



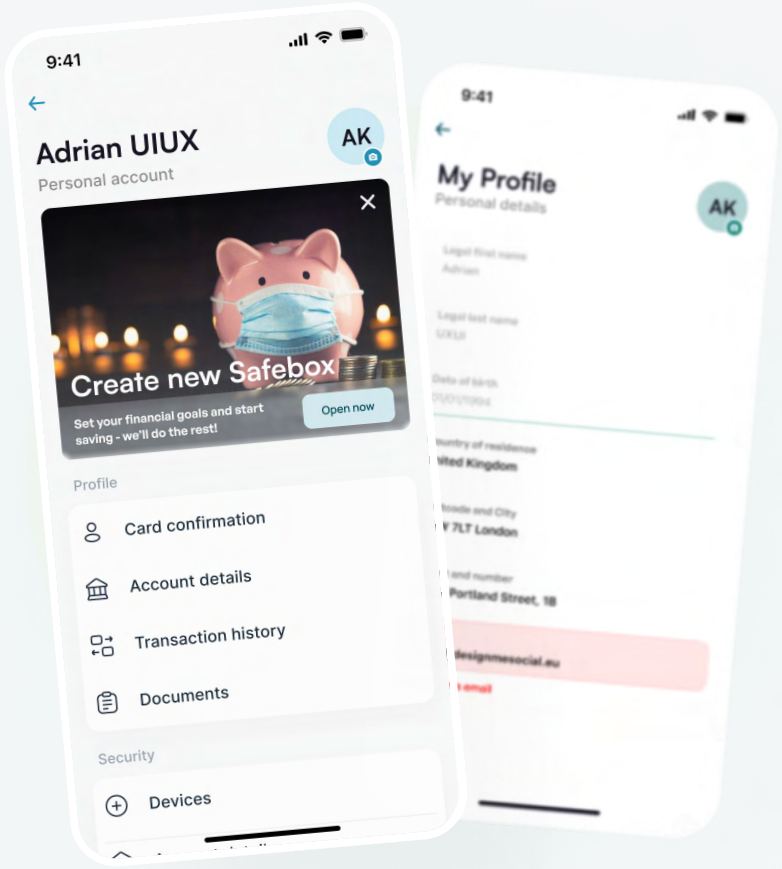
For the recipient input field we can copy the “Not selected” variable from our “Components” page. Resize it to 279 px width. Now we need to create our “Recipients” button.



For this we need a rectangle of 56 x 56 px dimensions and a “User profile” icon. Group them together and center the icon. Place the Recipients button 8 px from the input field, select them both and apply auto-layout. Make sure your input field elements are set to fill on both axis, and your Rectangle is set to **Hug**. Apply contraints “Left and Right” and “Top”.

Chapter recap

- Tabs are used to switch between multiple views inside the same context, not switch to different areas. This is one of the most critical aspects of segmented flows. The purpose of tabs is to stay in one location while rotating views.
- We want our design evolve along with this ebook to show you how, potentially, a real-life app project could look like. This way we are able to test different solutions and offer new ones in flows where users tend to drop off or are not interacting with at a satisfactory level.
- In a real-life example we would of course test a lot more iterations of each flow. It’s all about testing.
- Reminder: If you ever need to change the settings of your controls, look at the right property panel. This is where all your component’s variable properties are.
- I deliberately skip on repeating the same things over and over again, but you should keep in mind the rezising properties and your constraint settings at all times!



Profile settings

THE INTRO

App settings - intro

App settings help users manage items such as personal details, documents, push notifications, languages, biometrics, and login credentials. Every single app has settings, but users and designers often overlook the interface panels that control them.

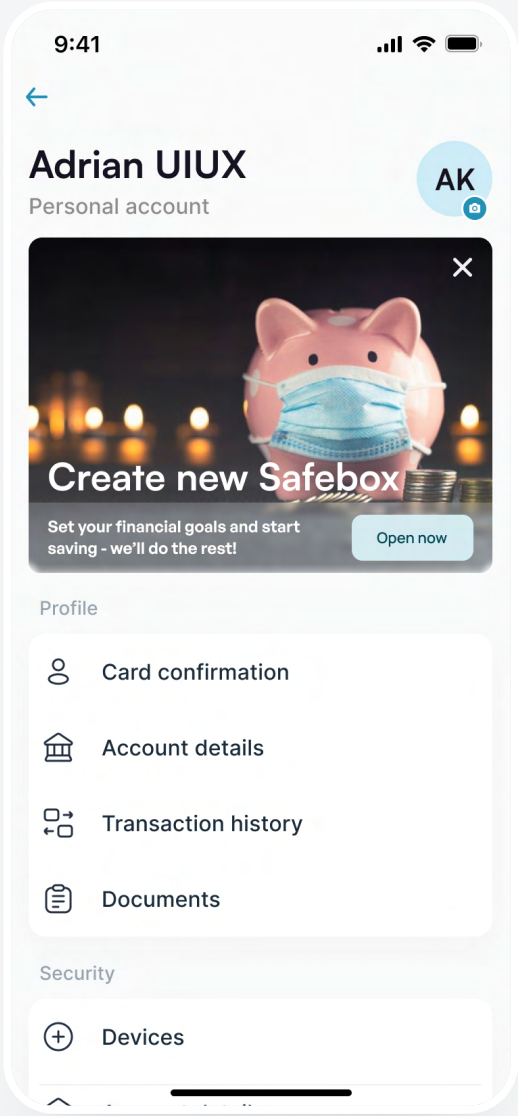
Settings are often neglected until very late in the app design process, but they are still a significant part of it.

Well-designed settings reduce required customer-support costs and boost engagement and app advocacy by allowing users to customize their apps however they want to.

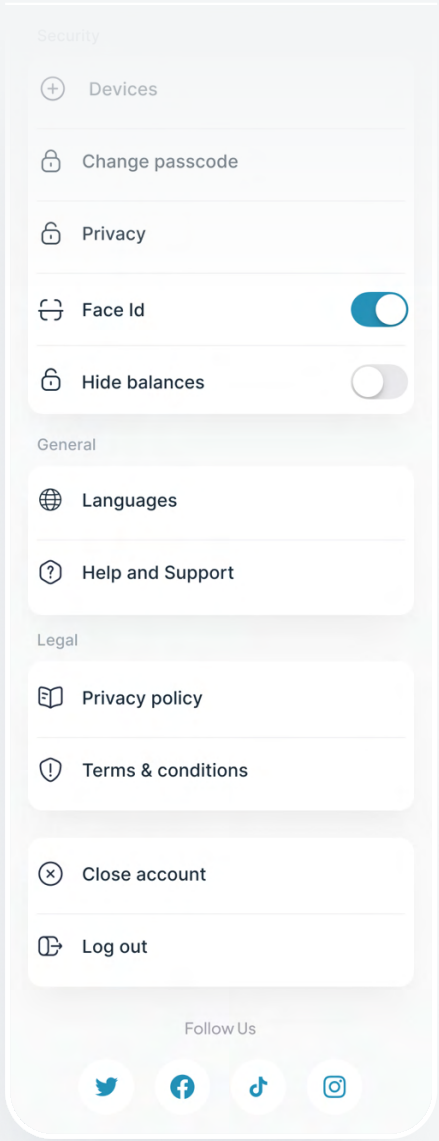
Our goal with the settings page is to make it easy for users to manage their preferences and edit every detail they might need to change.

When apps have ample settings options, good organization of data is crucial. If there's no clear visual hierarchy of information or the information architecture of the settings page is not logically built, our users will feel lost and frustrated.

What design practices should be ensured for better UX?



1.12.1.1a - Profile



1.12.1.1b - Profile

Group Categories

Our app has a ton of settings, and it can be difficult for users to find what they need. The best solution for that is to group settings into categories.

Visual Hierarchy

Settings panels need a clear visual hierarchy. You should always place the most frequently used settings to the forefront; if there are too many options, put them into subpages to reduce cognitive load.

Naming conventions

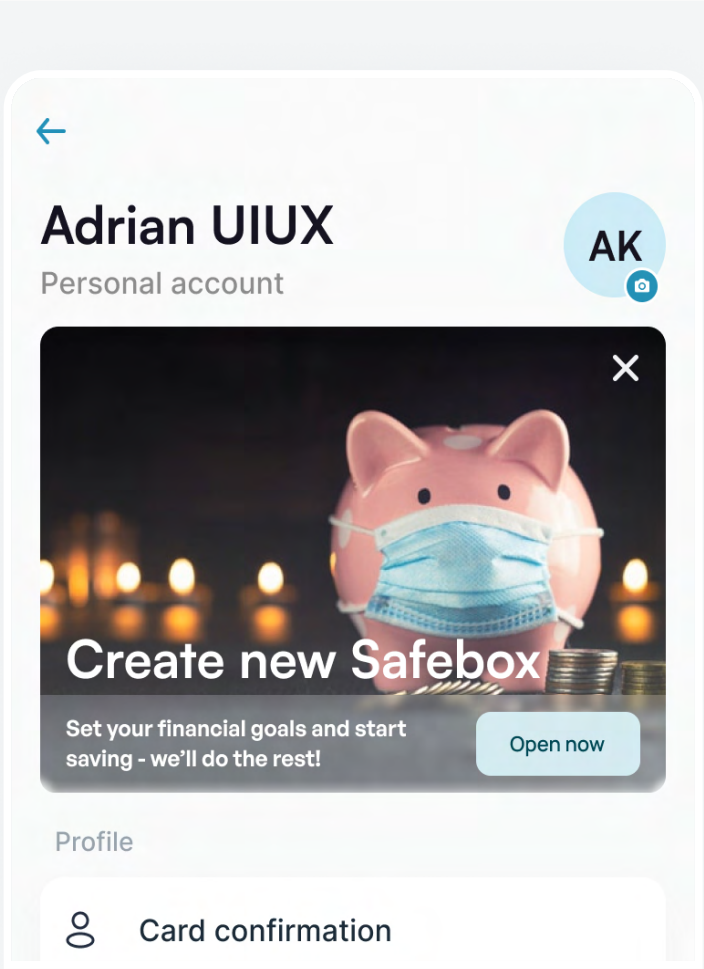
Avoid using unclear label names. Settings are best described in plain language that indicates functionality. Avoid using technical names that product and marketing teams might find understandable but could confuse users.

Define default settings

Default settings help establish optimal experiences from users' first interactions. With our detailed onboarding process, we want to equip users with default settings that support their goals and make our app effortless and fun for future use.

Personalization

We want our users to feel welcomed, safe and secure when using our app. Apart from top-notch security, we want to address them by their names, give them the ability to upload their own image avatars, and show them personalized offers that they might find interesting.



This is what really sets the Fintech companies apart from their banking counterparts. Clients of traditional banks often complain about the lack of understanding from their financial providers. Why does it happen? Because most banks are simply trying to promote their services to as many people as possible. They do not dig deep enough into the needs of each customer group. Fintech apps wanted to change that.

Users want to do business with a bank that would help them solve their problems rather than offering too broad or generalized services. With personalization in banking, we can tailor the experience and provide relevant recommendations to our user's needs and expectations, all based on the information we collect from our app's everyday use.

Important settings

Profile

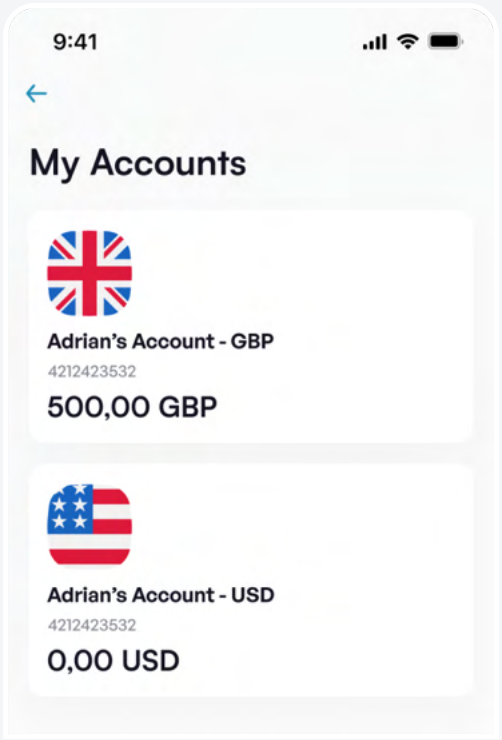
Basic personal details are automatically filled in with the information users submit during the sign-in process - name, business name, date of birth. These inputs are disabled, and we can’t change them since they’ve been already verified and confirmed.

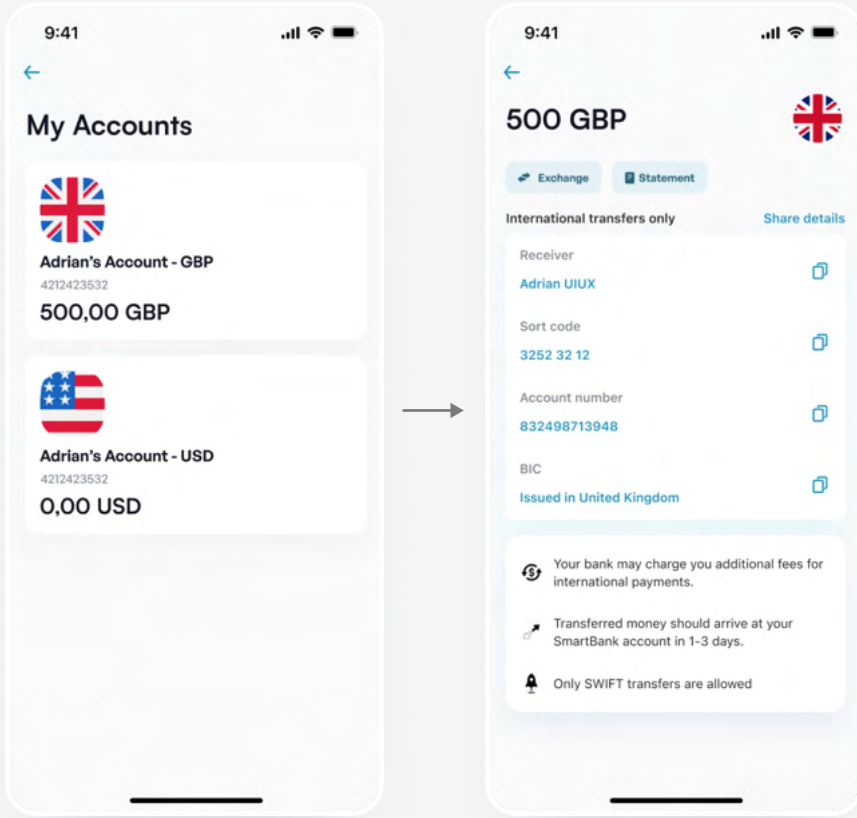
We can, however change our address details and email.

Account details

We can access all our accounts directly from the home screen, but what if we only want to see the account details of all the accounts we own? We can put them under our main profile settings. A card layout seems most fitting for what we need.

When our accounts grow in number, we'd introduce a search field and possibly a filter function to filter the accounts by specific parameters. For now, though, this is all we need.





1.12.1.3a - My accounts

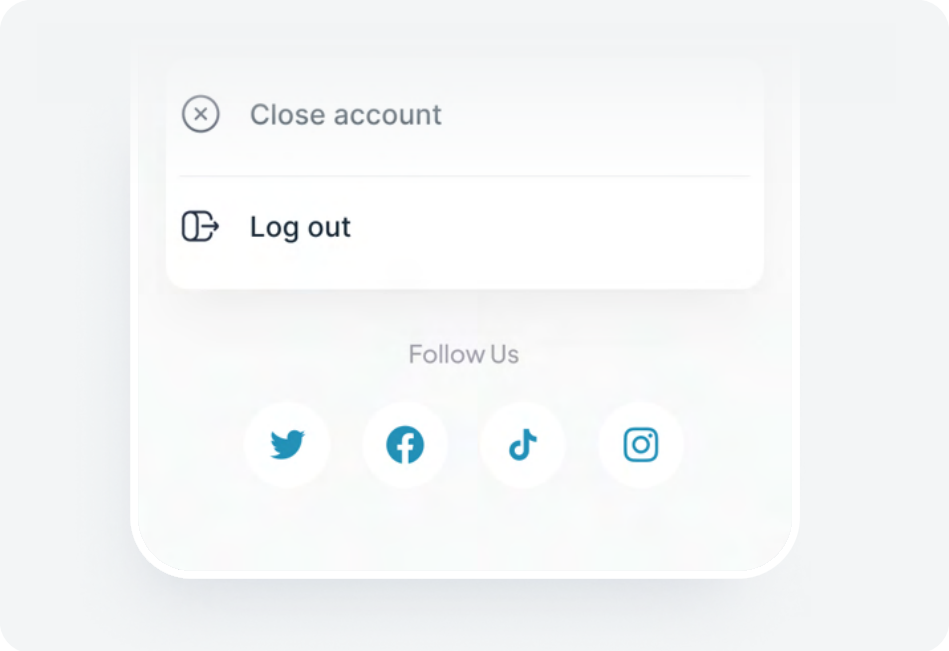
1.12.1.3b - Account details

On our accounts page, we can see the account details and a few CTAs like "Exchange" and "Statement."

With no funds on an account, we'd also add a direct call to action to top up the account directly through open banking, a bank transfer, or credit/debit cards.

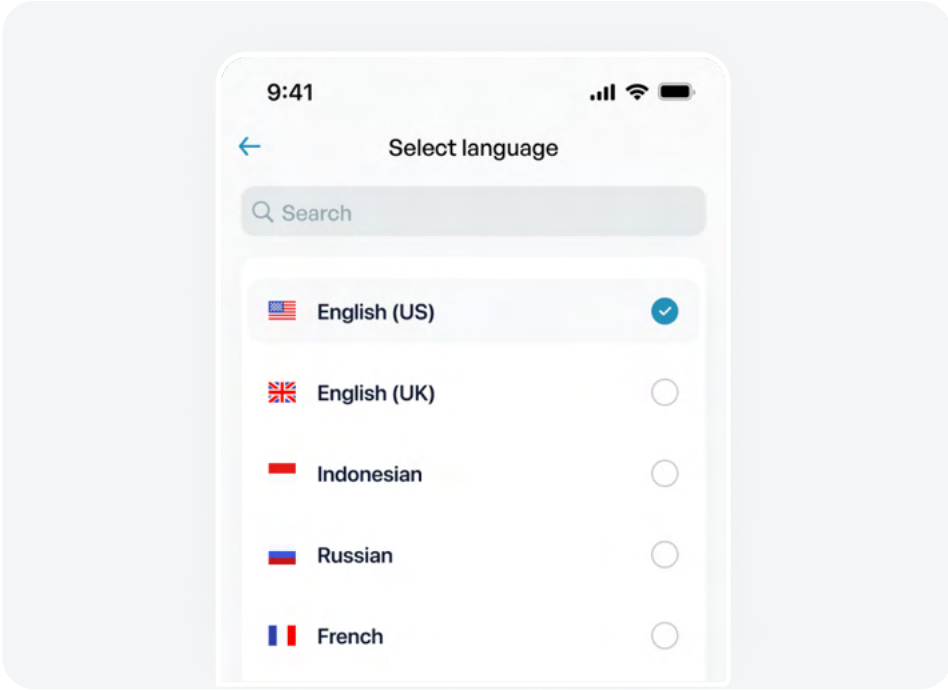
Destructive/secondary actions

We want to place the destructive and secondary actions at the bottom of our list since we don't want our users to close their accounts. Even though we've added our Log out button here, after every app session or when we put our app in the background, we should be automatically logged out for security reasons.



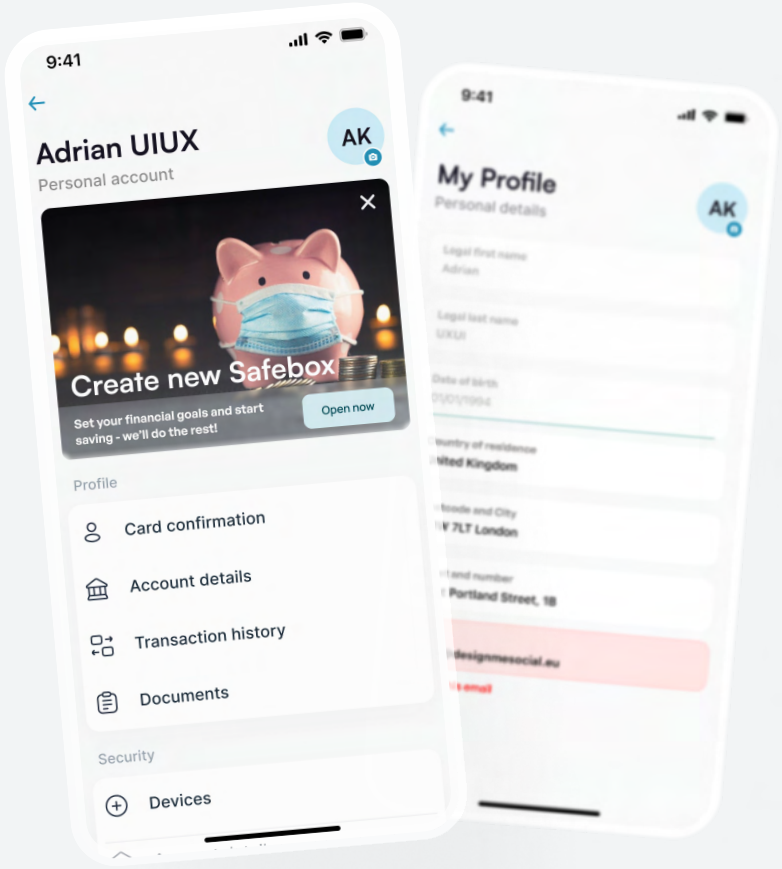
We've also added a small section with our social media accounts for users to stay up-to-date with our company news and the opportunity to learn about discounts and promotions.

Languages



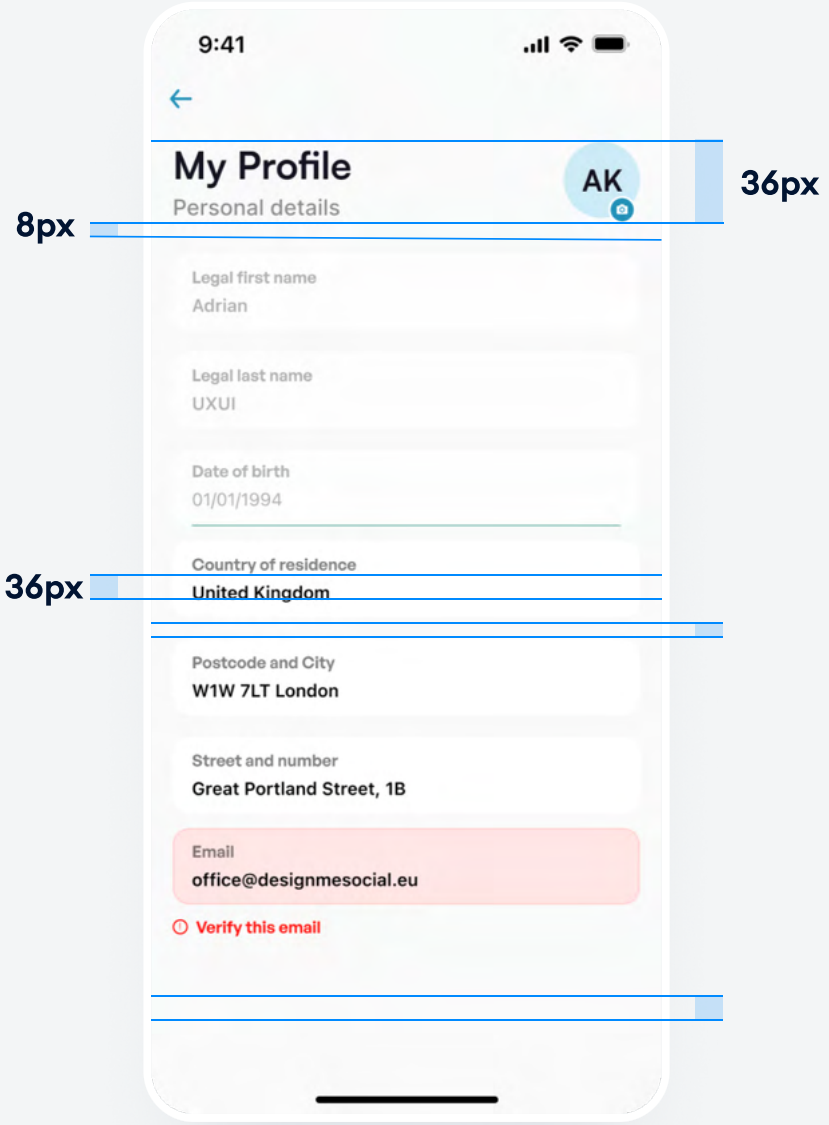
By default, language doesn't have to be assigned to the country of residence; instead, the active language settings that are set on your iPhone. The user must have an option to choose his preference manually.

Changing it is as easy as searching for a specific language, either by scrolling up and down the page or manually searching through the search input. Changing this would prompt the user to re-open the app again to save the settings.



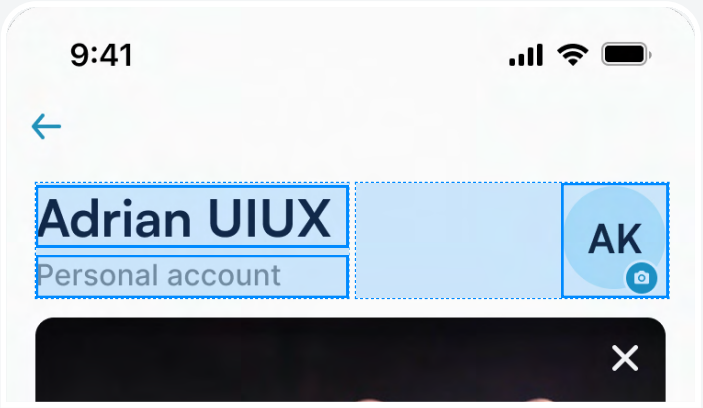
Profile settings

THE STRUCTURE

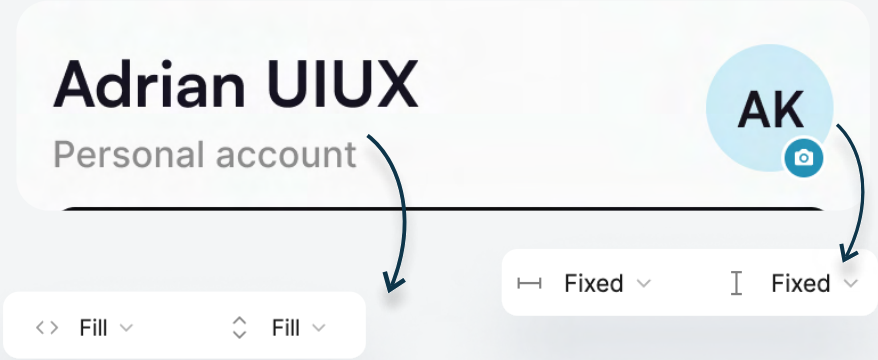


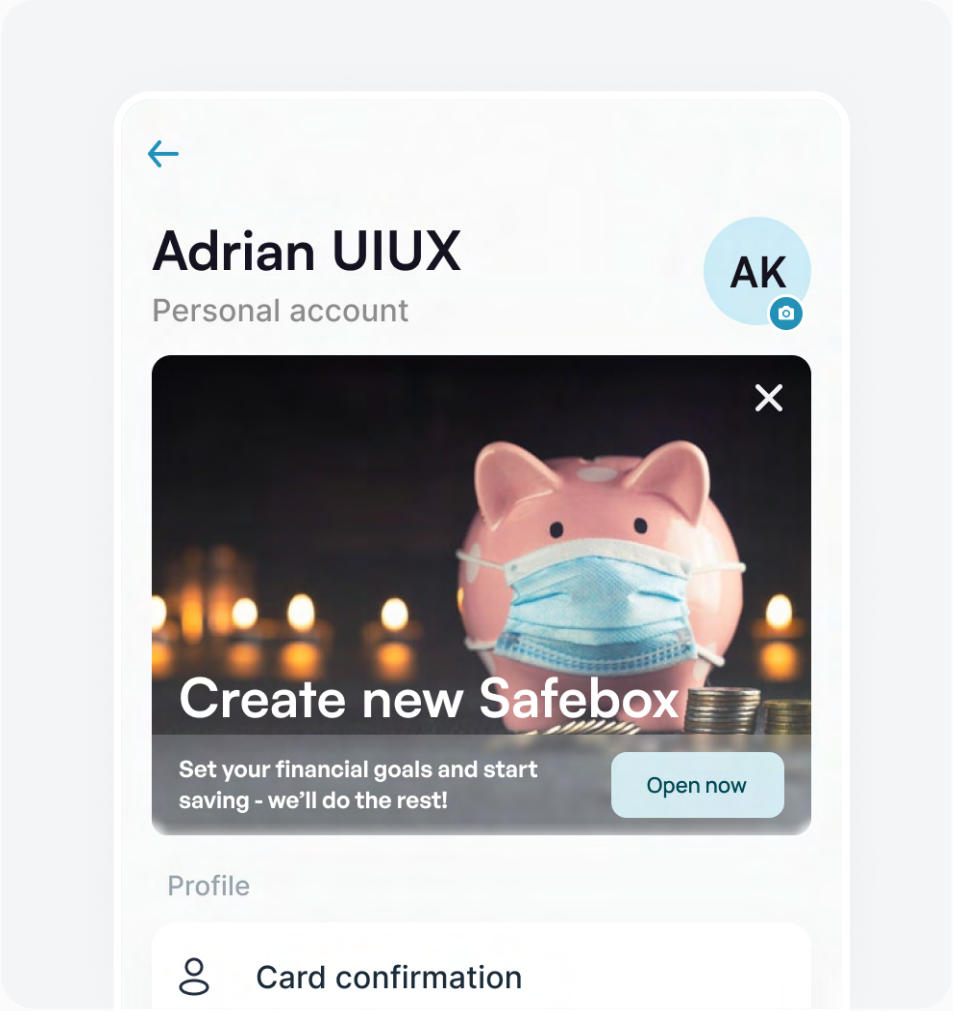
1.12.1.2 - Personal details

Profile settings - structure



You're probably a pro and know exactly how to replicate the above example. Find a similar card container and start replacing the copy and shapes. Remember about the **"Shift + Cmd + Opt +V"** shortcut to paste directly into a frame!

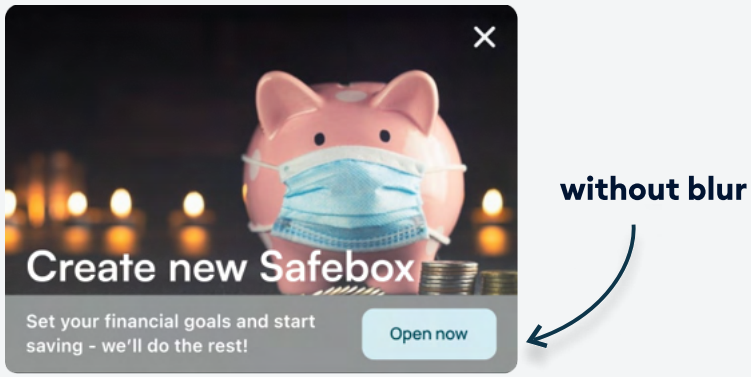


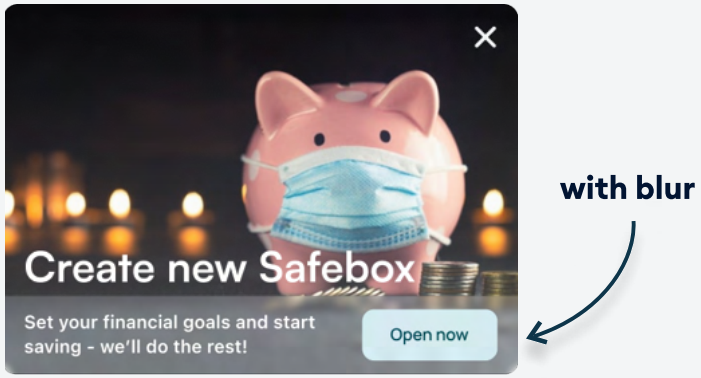


Before we go and replicate our promotional material container, we'll have to know two design patterns - masking objects and applying background blur. We've already covered both concepts before, but there's nothing wrong with refreshing our memory a little bit.

iOS Blur

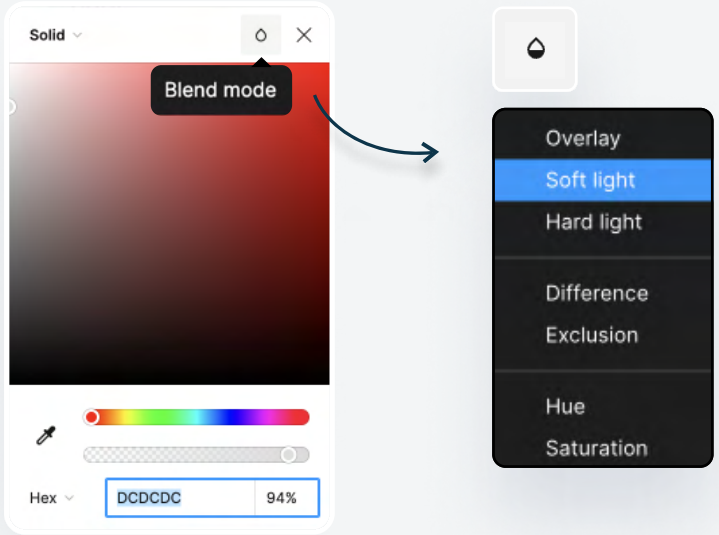
Using blur gives us designers a perfect opportunity to make overlaid text legible — we can simply blur a part of the underlying image to make the text more readable and add some additional depth to our visuals. Let's look one more time at how to replicate this effect.





You could simply choose a color, and reduce its opacity to a number between 30-90%, all depending on your background's focus, but there's a little trick to make it even better looking.

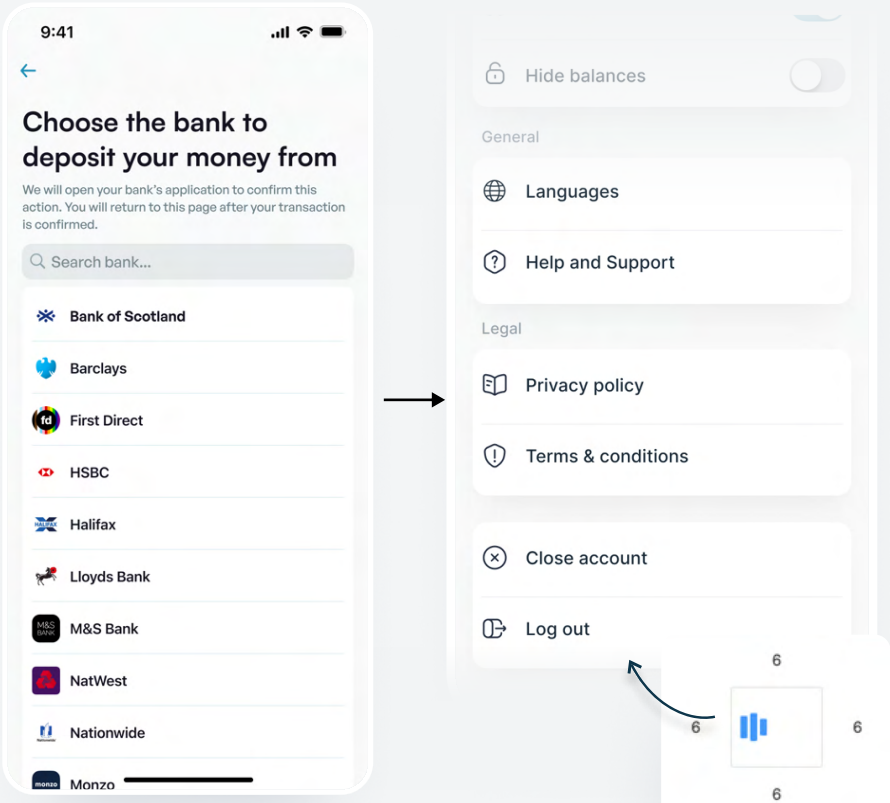
Here's what you need to do.



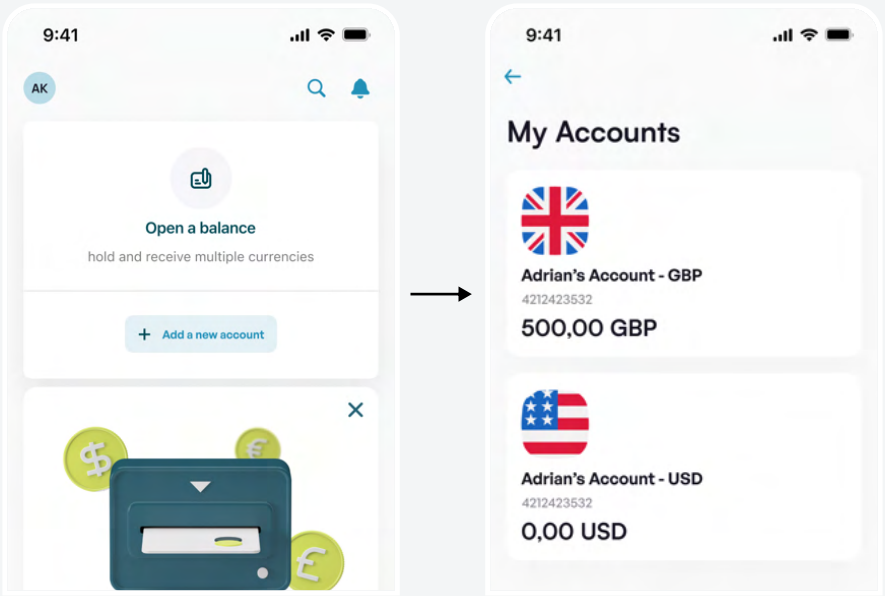
Apart from reducing the opacity to 94% and applying blur value of 6, you should always change the blend mode of your background layer to "Soft light." This will increase the saturation of the objects behind your blurred layer, similar to what iOS is doing in their native apps.

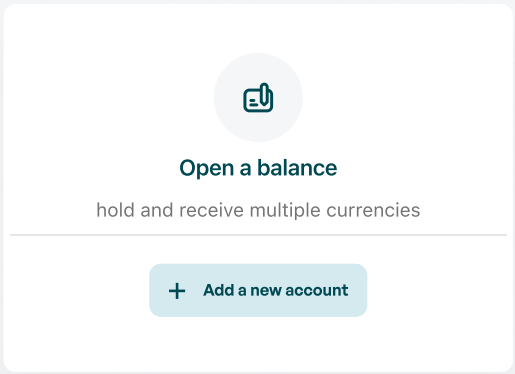
Settings

To replicate the settings page, we can use the menu items from our "Open banking" frame. We'll have to delete the extra menu items and replace the logos with icons and all labels above our groups. Duplicate it for all our settings. You got this.



1.8.1 - Open banking

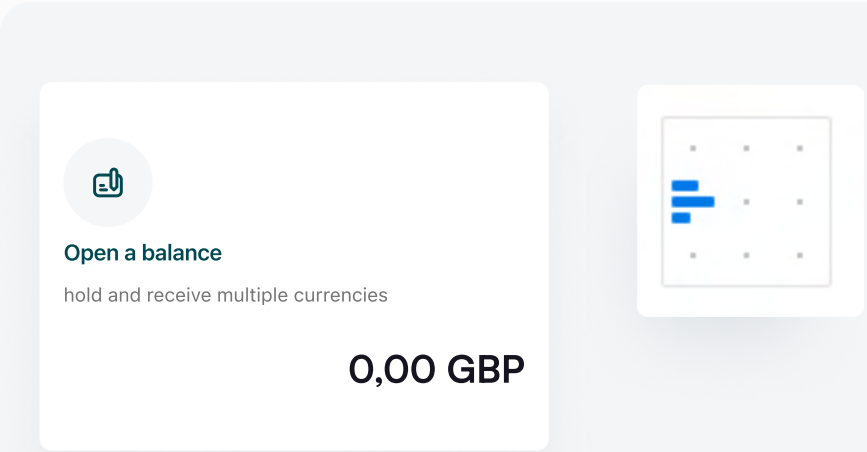




Account details

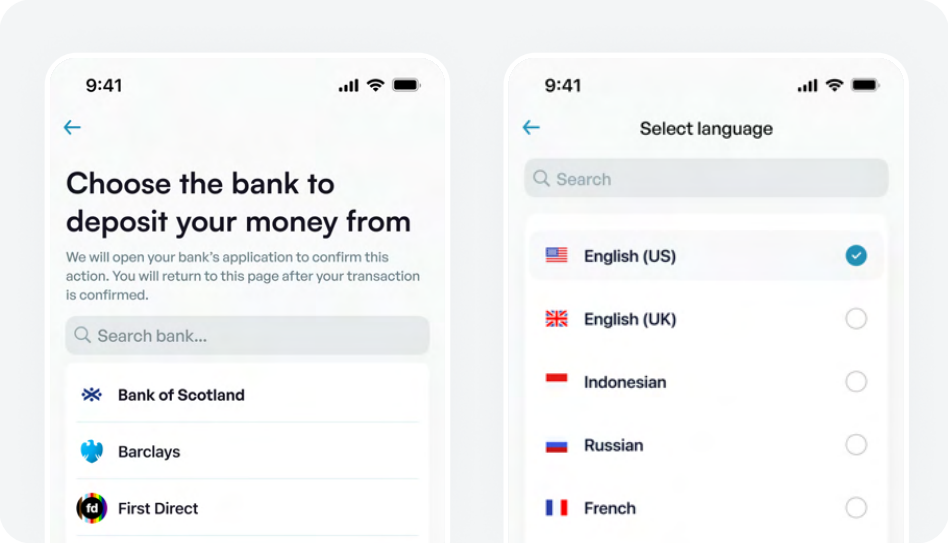
We can reuse the “Open a balance” component from our Home screen to create our account card. We’ll have to change the text alignment from the center to the left. You should also align your container and your child elements to the left in the auto layout properties.

Next, remove the divider. Copy the cash amount from the “Home screen - active user” by selecting it and clicking “Cmd/Ctrl + C.” Select your “Add a new account button” and click “Shift + Opt/Alt + Cmd/Ctrl + V” to replace the selection.

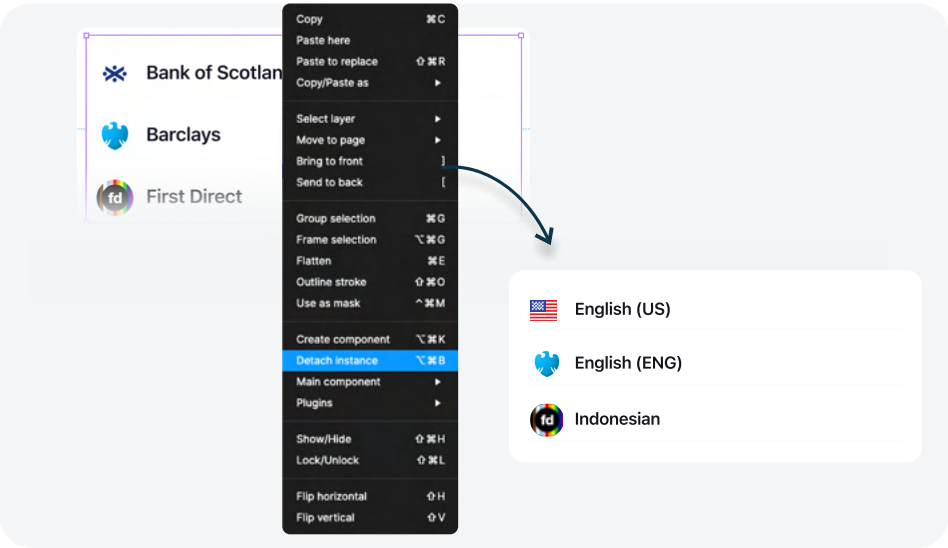


Lastly, align your amount text to the left and update your balance icon with the currency flag. Update your text styles to match the original frame on your left. You should be good to go now. Check everything with the original and if ready, let’s move on to the next one!

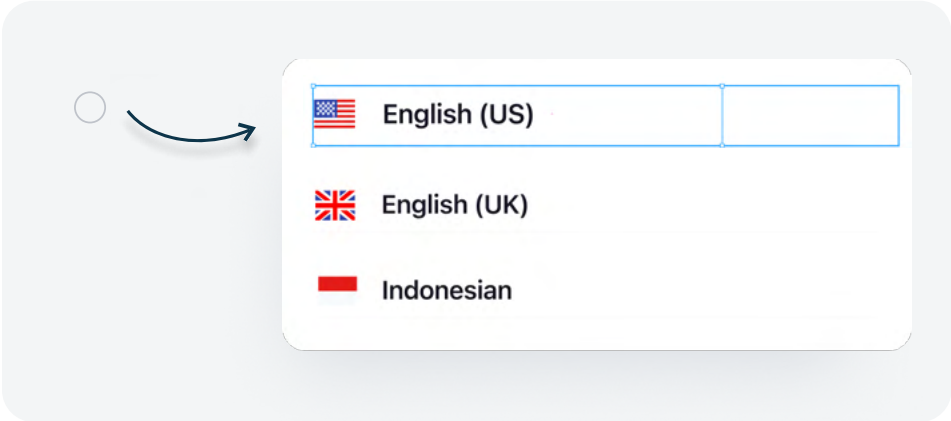
Languages



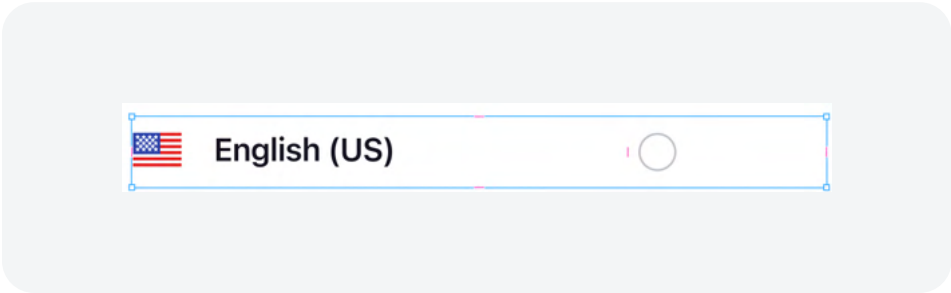
To create our languages list, we have to modify our existing components from the “Open banking flow.” Copy-paste them on your frame, and detach the instance. Update the copy and replace logos with flags.



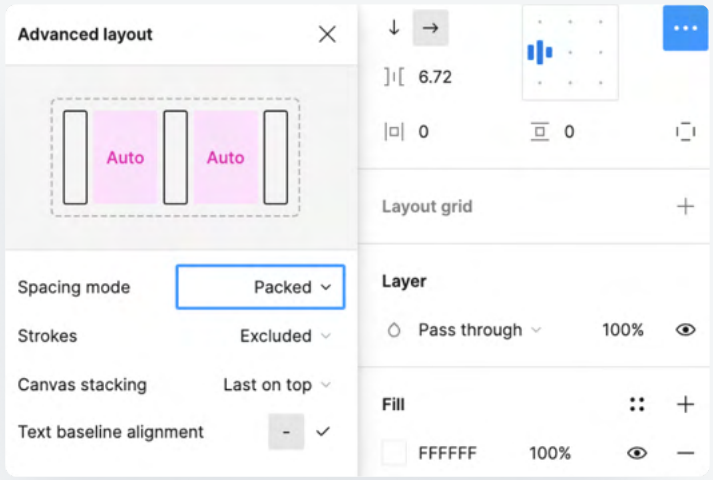
Create a circle shape, remove fill and add a 1 px border. Pick the color from the original frame. Create one checkbox icon, and make it green to indicate the selected state. Copy it by clicking “Cmd/Ctrl + C.”



With your language container selected, make sure it’s the outer container that nests the flag and the text; paste the circle by clicking “Cmd/Ctrl + V.”



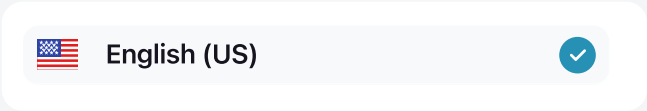
We’ll have to adjust the spacing mode now so that our flag and country group sits on the left end of the list and the radio button sticks to the right end. Navigate to your properties panel on the right.



Click on the three dots near your auto layout properties and change the spacing mode from “Packed” to “Space between.”

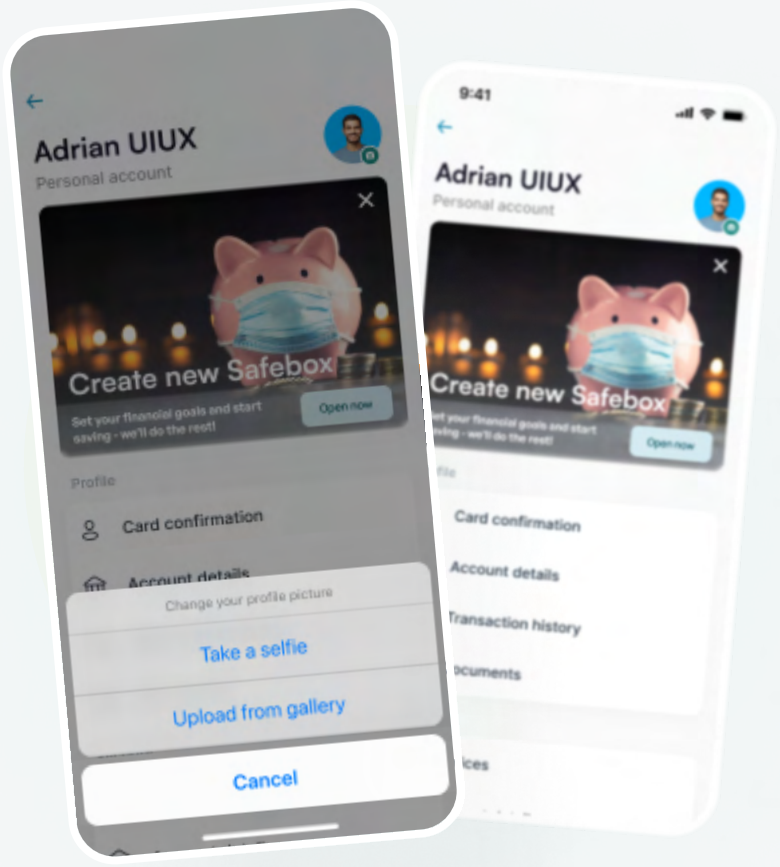
Your radio button should now jump to the right end of the container.

The last thing we need to do is select our entire item list container and apply fill, round the corners to 8 px and replace our radio button with a green checkmark to create our active state. That’s it!



Chapter recap

- Well-designed settings reduce required customer-support costs and boost engagement and app advocacy by allowing users to customize their apps however they want to.
- When apps have multiple settings options, good organization of data is crucial.
- You should group settings into categories if you have many similar settings options.
- Settings panels need a clear visual hierarchy. You should always place the most frequently used settings to the forefront; if there are too many options, put them into subpages to reduce cognitive load.
- Avoid using unclear label names. Settings are best described in plain language that indicates functionality. Avoid using technical names that product and marketing teams might find understandable but could confuse users.
- Default settings help establish optimal experiences from users' first interactions.
- With our detailed onboarding process, we want to equip users with default settings that support their goals and make our app effortless and fun for future use.



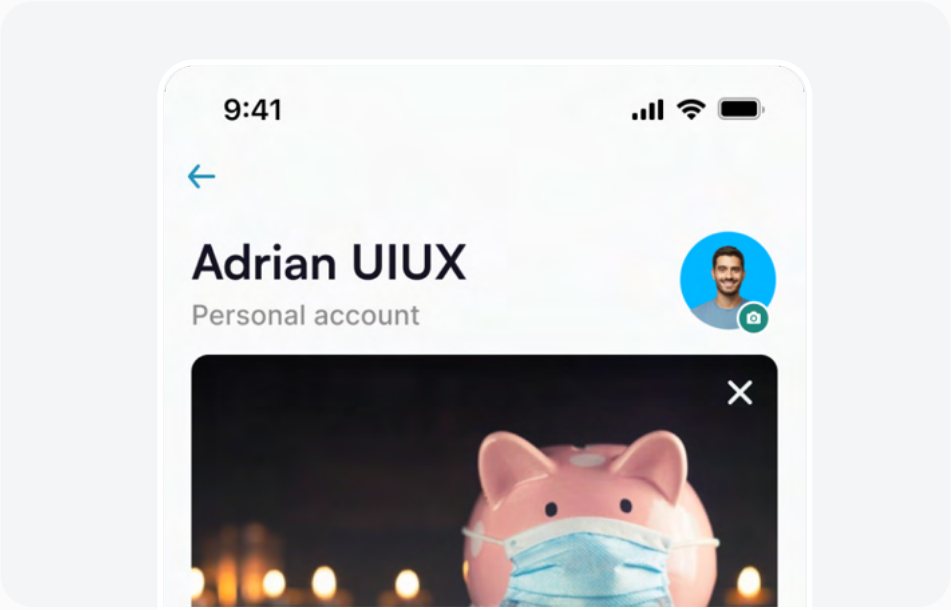
Profile avatar

THE INTRO

Profile avatar

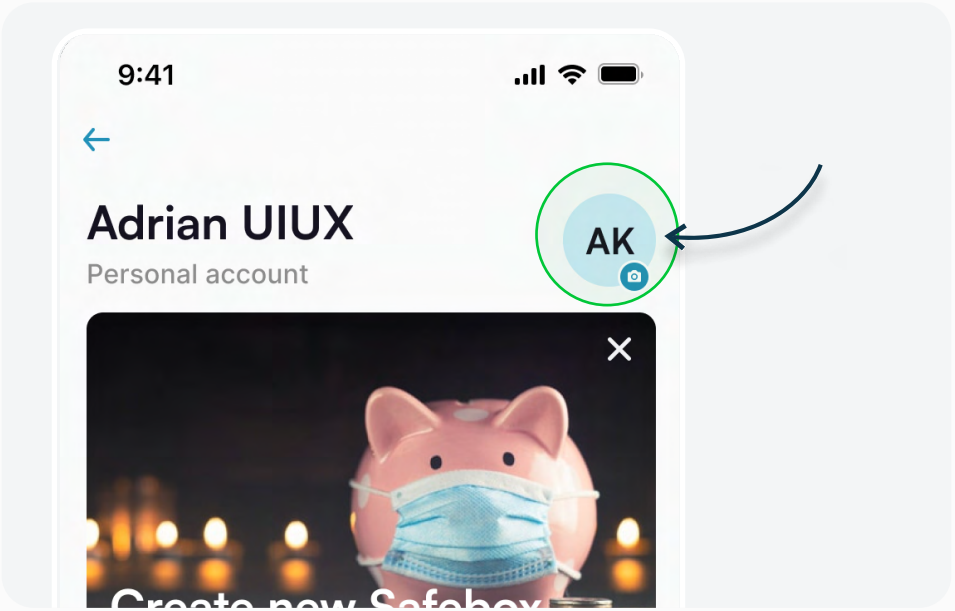
We wanted to introduce more personalization by adding profile avatars to user names. These will show up on your home screen, on your profile page, as well as display on your friends' profiles. and other users you'll be sending to or receiving money from. Granted, they have an account with SmartBank.

This way we'll for one, fill the entire app with images of familiar faces, and two, we'll make it feel much closer to the social apps our users are familiar with which might create closer bond with our brand by association.



Uploading an avatar

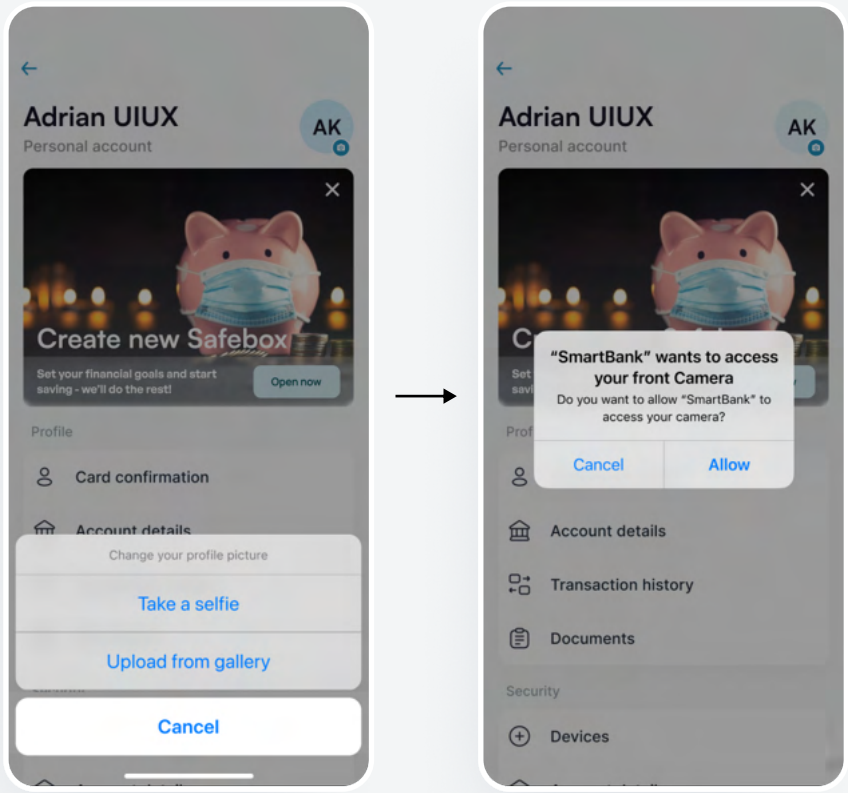
Users can personalize their avatars with a chosen image or photo by clicking the avatar in the right corner.



This will pull up an iOS bottom sheet with a list of options related to the action taken which in our example are:

- Take a selfie
- Upload from gallery
- Cancel

This is also complemented with a descriptive text above the sheet. It's all native to iOS and there's not much we can modify here except the copy. Let's see how it looks like on the next page.



1.13.1a- Upload an image 1-3

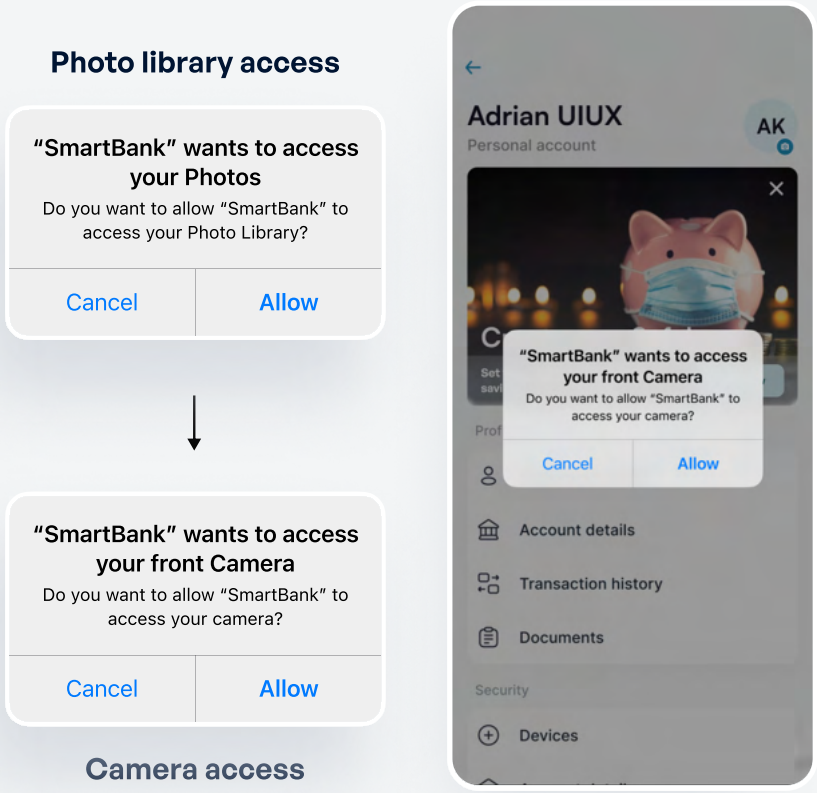
1.13.1b - Upload an image 2-3

iOS Bottom Sheet

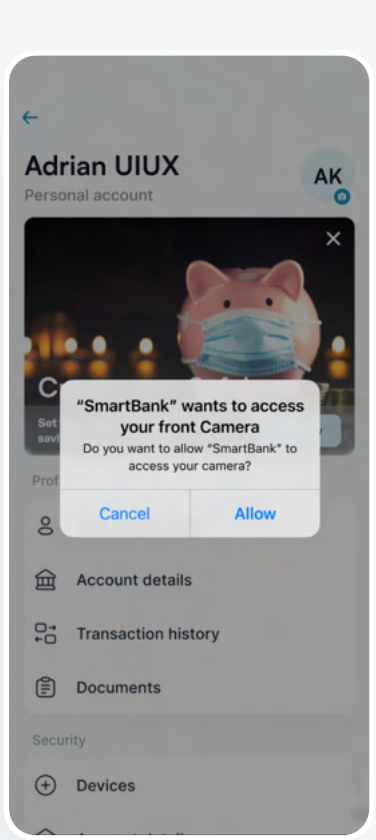
A sheet allows people to do a specific task connected to the parent view without leaving their present context. A sheet is a card that partially covers the content underneath it. To visually separate it from the parent view, the card's upper corners are rounded. The corner radius of a sheet can be customized.

iOS Alert message

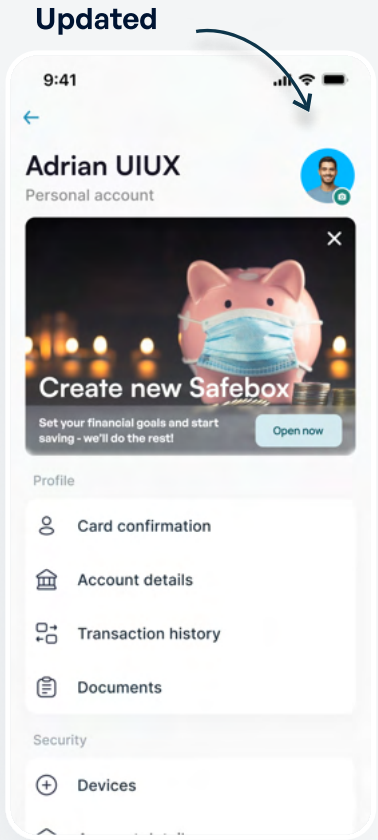
This would only show up if we reverted back the access to our camera while uploading the ID documents in the verification process, or if you've allowed the app to access the camera settings only once. Either way, we'll have to design this state.



1.13.1b - Upload an image 2-3



1.13.1b - Upload an image 2-3



1.13.1c - Upload an image 3-3

Right after the access is given, we'll take a selfie and update the image, we're ready to go!

Part 2 of the Pratical Design Manual is coming soon!

You've reached the end of Part 1 of the Design Manual.

We've covered 15 chapters so far, and 9 more are coming in the next few weeks. I wanted to take this time and thank you for sticking till the end of the Practical part. I hope you got to understand more about the different flows of a Fintech app and, most notably, how to work with Figma and replicate designs that are functional and aligned with iOS guidelines.

This e-book will receive many updates, and along with each update, I will release a changelog document just so you can quickly look up which chapters have been updated and why. We'll keep improving the content with new relevant information and updated design flows.

Once again, thank you for reading through the end of the Practical Workbook. In the following chapters, you'll find information about the design hand-off and a few ending notes with additional information about resources and freebies, so make sure you read through that before you finish this e-book for good.

See you on the next page!

Design hand-off

22. The design hand-off

Design handoff between design and development is essential but not given enough attention in the whole design process.

It's important to know which elements the development team needs from the design team. There are some handoff tools on the market to deal with these tasks, and I'll show you briefly how I do it.

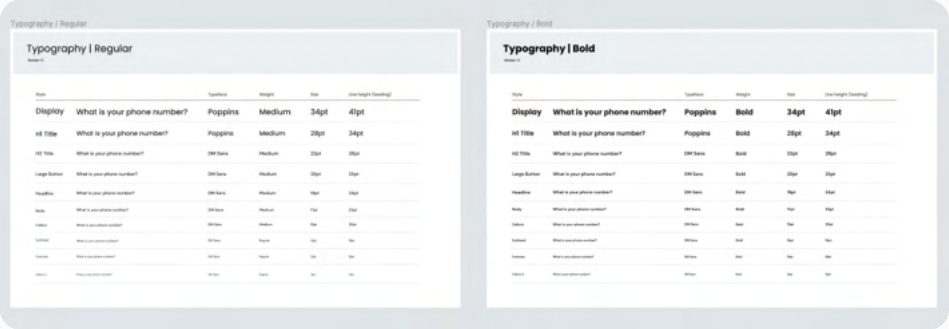
To achieve the right end product, it's essential to set up your style sheets the best way, export elements in the correct form and size, set up component libraries and specifications, and understand responsive grids and layouts, pixels, points, and type scales to make the right decisions early on in the design process.

In this chapter, you're going to learn how to share your Figma or Sketch designs with the development team professionally.

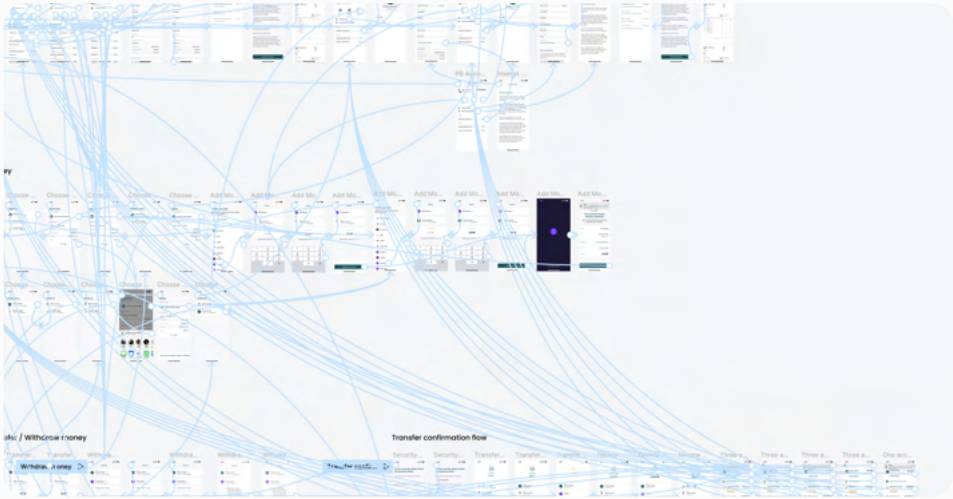
Creating a design system and a style guide for app use will give the development team a clear understanding of what needs to be done and how it needs to be done without too many back-and-forth conversations and misunderstandings.

Before handing the designs off, we want to get an overview of what a development team actually needs from us. Putting together a list of things that we need to prepare to get the best possible results is a good idea to start.

Typography guide - to explain the fonts we are using for each specific purpose. We'll need to create two type scales, one for regular text and one for bold. You never know when you'll need to use either one, so just to be safe, I create both.



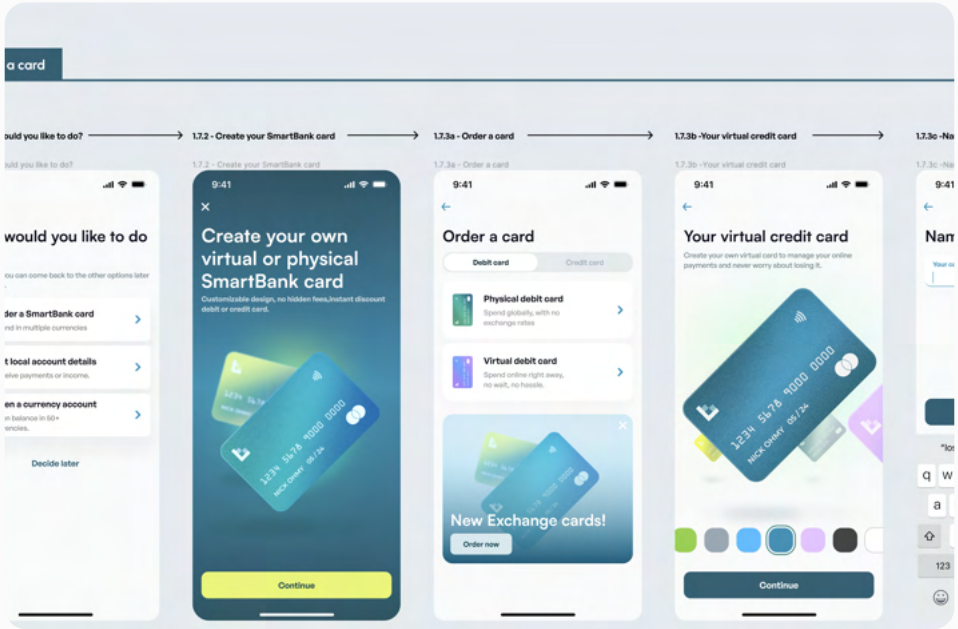
Quick prototype or a flowchart to show how each screen relates to another and what is the user journey for each of the app segments. I also group the app screens into respective flows and label them accordingly to reduce potential misunderstandings.



Frame naming conventions - I follow a pretty simple rule where I label my screens like like in this manner:

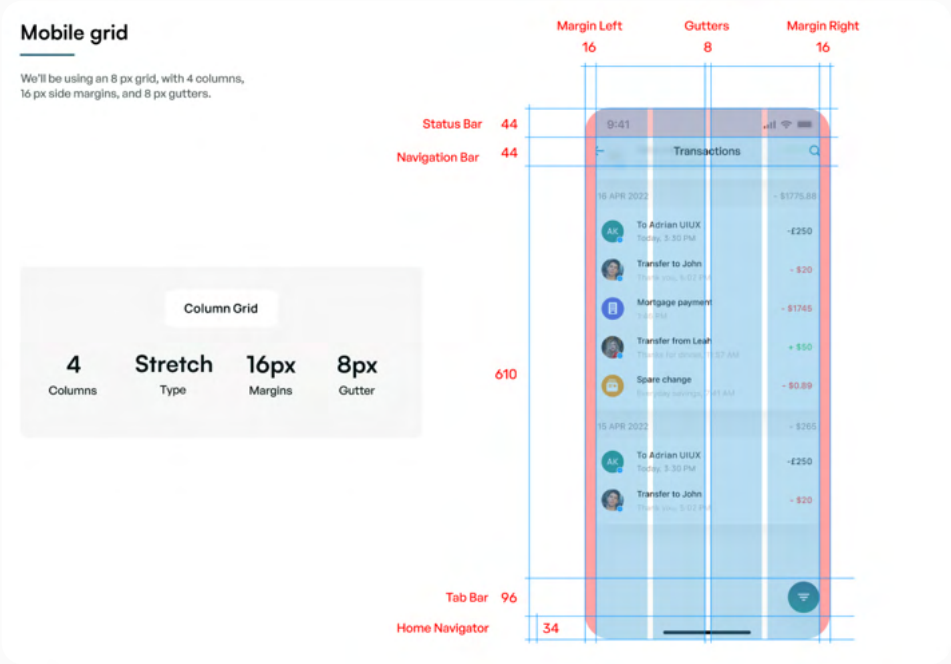
1.1.1 - Onboarding Step 1/3 - what are the numbers, you ask?

The first number is the iteration number - in our case, that's the first one; the second number is the number of the flow, which is our first flow. The third number is the step of the flow. Sometimes the third number can include letters like "a," "b," "c," etc. that's when you have to split the flow to show two states like empty, focused, finished, disabled, etc.



I also line up my screens if they follow a streamlined journey (or use Autoflow plugin) to visualize the entire flow a bit easier.

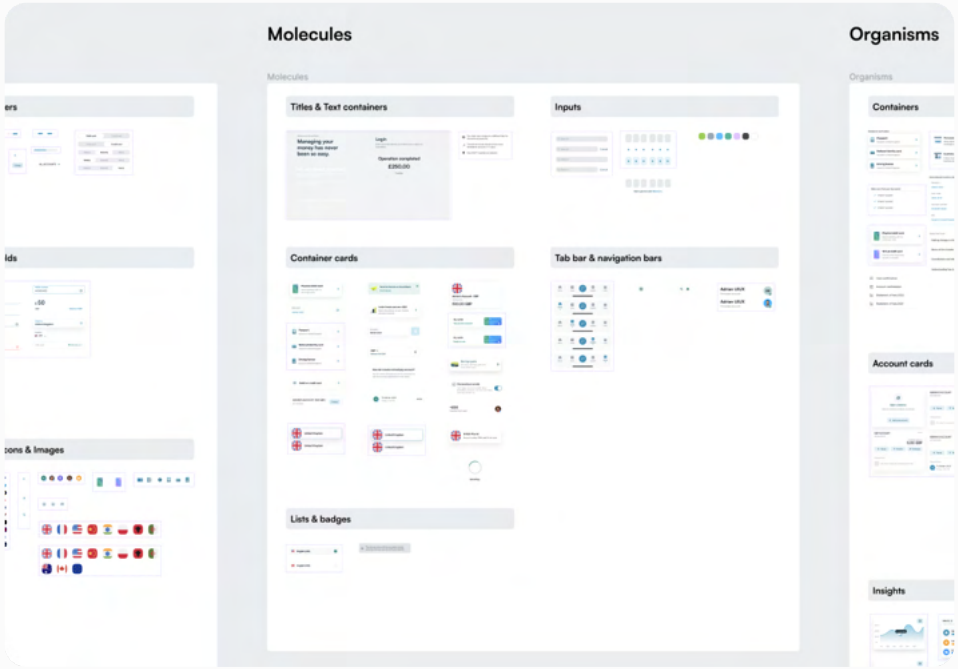
Grids and safe areas - Another thing I create is a grid system guide. I also highlight the safe areas on the frame.



For your grid documentation, you should include an example screen outlining the safe space, the margins, and the gutters, and explain the spacing you used to build the elements and place them on the frame.

If you're creating a website-based project, this documentation would also include different device breakpoints - monitors, laptops, tablets, and mobile devices. In our case, we'll show two designs on each end of the spectrum - the iPhone 13 Mini as the smallest device and the 13 Pro Max as our biggest device.

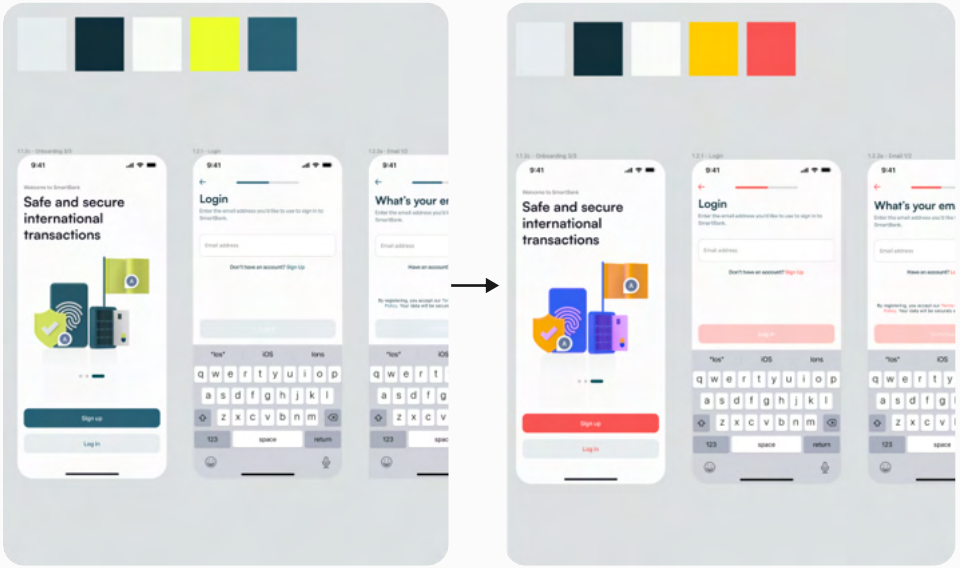
Components library - I create a component out of every single element in the app and build a components library so that developers can lookup single elements and see how they've been designed. It's also super helpful when you need to change the shape of your buttons, the border-radius of your interface elements, or things like text containers and your auto layout selections.



Your components need to be appropriately grouped and labeled so that developers can quickly locate all the files and assets. Annotations might be required to give more context or describe functionality in places where the capabilities of a design tool are limited.

We should also walk developers through the documentation ourselves.

Style sheets - Your styling should be documented in a style guide or a design system. That includes the things we've already mentioned, like colors, typography, imagery, iconography, and grid layout. Still, it should also include things like shadows, blurs, and illustrations used across your entire app. It's essential to replace all your file colors and font styles with dynamic global styles instead of using regular Text properties with static values. This way, you'll be able to easily make global changes and see them being reflected across the entire file.



We're not going to build a full-blown design system for this project because it's way more complicated and needs a separate release to explain it. To learn more about the design systems, check out my Instagram account @uiadrian. I'll be posting content around the design system we created for our in-house project as an agency.

I will include all the hand-off documentation and file templates I use in the folder named "Design hand-off" - it will be released soon!

It may seem like a daunting task to document every element, and usually, this topic creates a lot of confusion. Some developers will require you to work with auto-layout and make your designs pixel-perfect. That's why it's important to know how the content should be structured and laid out into containers. We've covered all the auto-layout structures in the Practical Workbook, so if you haven't read them yet, go back to a few chapters and study them.

How to ensure your designs are pixel-perfect? Make sure you design on a "Soft Grid," and all your elements and spacing between them are built using increments of 4. You've already seen that iOS components were either 32-36-40-44-48 pt tall and usually at least 343 pt wide. Why 343 pt and not 344? It's because when spacing our designs vertically, it's more important to leave margins of round values like 14-16-18-20 pt, and considering the smallest iPhone frame we've designed was 375 pt wide, we needed to add both side margins (16*2) and exclude them from our total width - (375-32) = 343.

This shouldn't concern you, though. If you've used auto layout and constraints to build your designs, the width of all the elements would automatically adjust based on the active screen resolution we set on our frame and fit the frame, leaving enough space for our margins.

What's next?

23. What's next for you?

The goal of this ebook was to teach you the basics of iOS design and show you real practical examples of best design practices so that you, in the end, can become a better designer. But knowing the basics and replicating one app is not enough to set you apart from the rest. It's only the beginning of your new journey.

What we want you to achieve is real success in your design career, and real success doesn't really come without practice and a solid plan. A plan with real actionable steps. So, what are the next steps for you?

Now it's time for you to put this knowledge into practice.

If you've really finished reading through every chapter and followed our design manual to a T, you already have a solid foundation to grow as a designer. The only thing you need now is to keep honing in on your skills and sharing your work with the world. With our design framework you should be able to design at least twice as fast and improve your workflow tremendously, and with enough practice you'll become a literal beast in design. I remember having to spend an entire day on design tasks that nowadays take me 2-3 hours to complete. And the best part is, with your rooster of clients getting bigger with each project you get to increase your prices along with that. You can really make a ton of money as a freelance designer if you do it right. I for one, was able to earn as high as $20k months just by designing cool stuff.

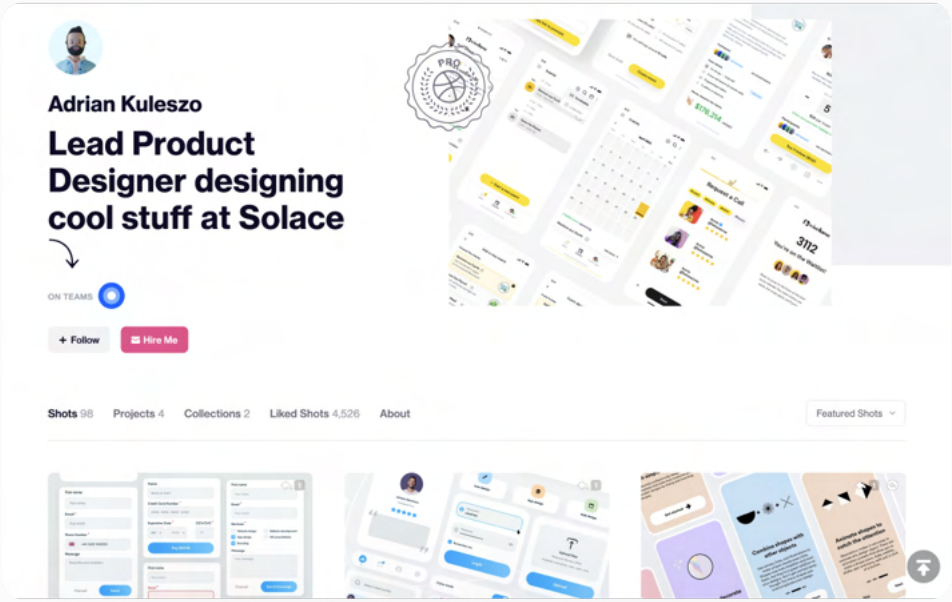
It really feels like a dream job when you think about it. You get to work remotely, literally anywhere in the world, do what you really enjoy doing, get better and more confident with each client and new project, progress as an individual, both financially, as well as mentally. With the extra time you have you could take on another project and make more money or work 2-4 hours a day and enjoy life. It's all possible!

There's a world full of opportunity waiting for you my designer friend. You just need to get noticed. But to get there as fast as possible, you need a solid plan. So, let me tell you what I would do, if I just read this e-book and started from scratch.

Design, design a lot

This is where I would start. I'd give myself a week or two, or even 30 days of just simply designing interfaces. Challenge yourself to design daily and really commit to it. Spend as much as you can invest into this but really try to block at least 30-60 minutes of your day to design.

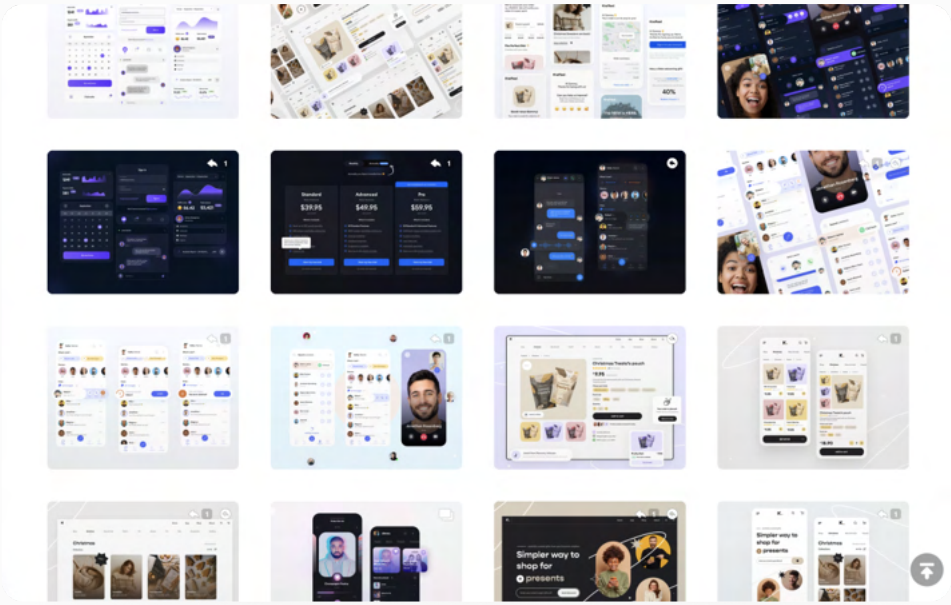
Create a dribbble account (if you don't have one) and post your work out there. Keep it up for as long as you can, this momentum will get you going and you'll see soon enough how quickly it piles up. Don't push yourself too hard though, you wouldn't want to burn out before you blow up! Try to focus on delivering quality work that's actually usable and functional but don't be afraid to get a little bit more creative and design stuff just for fun. It's all for good measure.



While designing focus on the most popular flows. Flows that you'd find in almost every app - a login screen, profile, home screen, search, onboarding screens, insights, analytics, you get the gist. To find more flows to recreate you can always refer back to this e-book and its practical guide. We wanted to give you as much content to design as possible. If you however run out of ideas, go to mobbin.com, buy a subscription plan (it's not expensive) and filter your search results by categories. Replicate the most popular apps, add your own twist to them, change colors, fonts, borrow some interface elements from a couple of different apps, mix and match them. It's a lot of fun to create concepts like that.

Try lots of different styles. Play around with colorful designs, create something more serious, try 3d elements, or flat illustrations.

While practicing you'll also begin to notice your own style coming through most of your designs. Work on it. This will be the thing that differentiates you from the rest. Gain confidence and start building your portfolio while you're practicing.



If you look closer you'll start noticing my personal style of designing

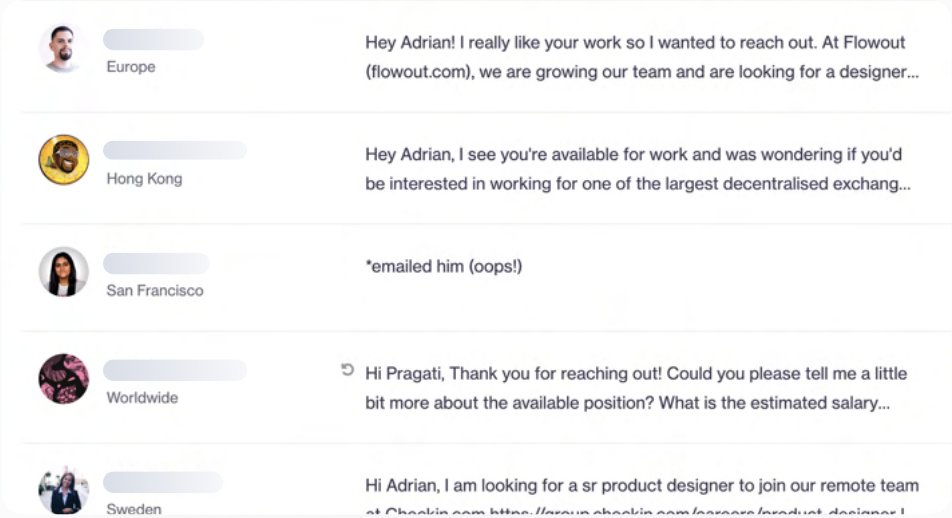
Start building your portfolio

If you really commit to practicing design for the next few weeks, you should have created anywhere between 10-30 different concepts. With that, you should have enough content to create your first portfolio. But nice looking designs are not what sells you to a client or a hiring agent.

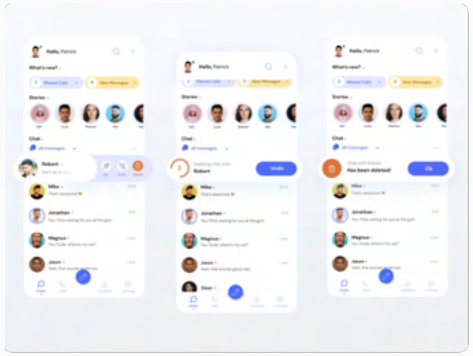
How to get noticed?

You need to post your work where your clients and other designers hang out - that is Dribbble, Instagram, Behance, and Linkedin. These four platforms should be more than enough for you to find good work, and get noticed by other agencies. You don't have to be active on all four but it really helps to get noticed. Besides, it's really quick and easy to repurpose the same content for all the above platforms.

Even though I'm repeating all the time that Dribbble is not an ideal place for serious research, it's also one of the best platforms to get yourself an influx of quality leads. If you can get through the noise and land on that explore page, there's no way you won't get swarmed with job offers. I haven't posted to Dribbble in a few months and I still get one job inquiry every other week. That's also how I got noticed for my full-time job at Solace.



With Dribbble, you just have to be consistent and post at least 3-4 times per week to get the algorithm favor you. One easy way to keep posting new stuff is to build a concept app and create a number of shots out of it. If you design an app consisting of 20 screens, that's like 20 different ideas for a Dribbble shot. Change fonts, create a dark UI, take components out of the app and present them all separately, show off your style guide. The ideas are endless. If you're out of inspiration, check out my Dribbble account - I'll be posting shots from this UI kit for the next few weeks just to show you how much content you're able to produce from one concept app.



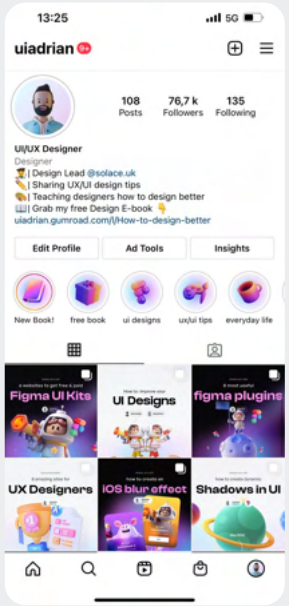






What about Instagram?

Instagram is great for exposure and building a community around your personal brand, but is it good for getting clients? Well, not really. Of course, you get a lot of different inquries if you land on that explore page but for me, they've never been of high-quality. Although, I wasn't actively looking for a job when I started posting on Instagram so I never really got to know the job offers I've been receiving. I'd recommend you post to Instagram regardless of that - if you're consistent with your posting schedule, you'll start growing on your own, and modifying that Dribbble shot into an Instagram frame will take you an extra 1-3 minutes. I actually started from posting designs a looong time ago :D

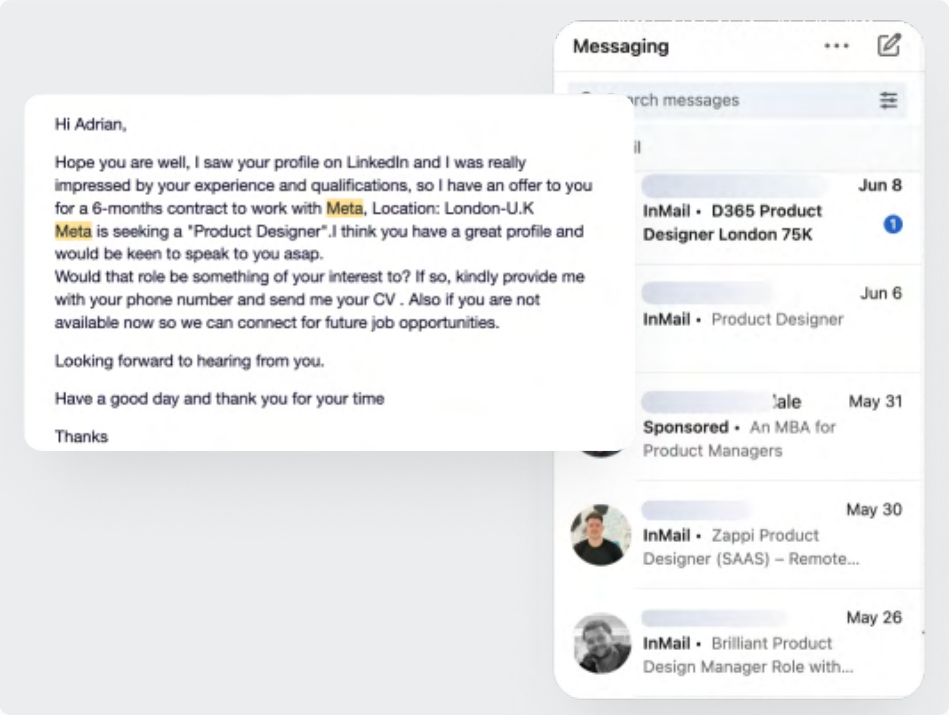




Is Linkedin really that good?

This might come as a surprise but Linkedin, for me, has been the best platform to get industry-related job offers and tons of freelance projects. That's the platform that brought me most of my retainer clients and made it possible to make steady $10-20k months.

And when I changed that job title from an owner of a design agency to a Product Design Lead I've been swamped with job offers, receiving one inquiry per day, even from companies like Meta and Twitter. It really is an amazing platform to generate leads and get noticed, so make sure you're active on Linkedin too.



Behance

I kind of have a love and hate relationship with Behance. It never really brought me that much exposure, except leads from local agencies looking to outsource additional work. I've also never been too active on Behance anyway, maybe because I was lucky enough with client work that I never really had time to create elaborate case studies. If you can spend a good amount of time and really polish your presentation and make it into a really professional, in-depth breakdown of a problem-solution concept work, people will notice it, and Behance will push your content further. I can't really say much about Behance but I guess it won't hurt to post there too, especially when you're just starting out and testing different waters.

I've got a little bonus for you!

I've included a separate file with a number of presentation mock-up templates you can use to display your future work both on Dribbble and Instagram. You simply need to drop your images onto a frame and they will be automatically transferred to the mockup. You can then change your background color to match your designs or unlink the templates and use them freely, as you want to!

Hope this little bonus will help you get started a bit easier. Try it out and tag me on Instagram (@uiadrian) when you're done! Would love to see your concept works live!

Final word

24. This is it! 🙌

Whoa, you made it to the end, that's awesome! I hope you enjoyed this e-book, the UI kit, and our design framework. I really wish this e-book provided you with good value and I do hope that it will help you progress as a designer. It was a blast writing it for you!

As you've noticed already, this is a long and extensive book covering both theory and practice. Don't make yourself feel like you need to know everything after your first read. Come back to it anytime you feel lost in the design process or need to get some information about iOS design, check the type sizes, or look at the Onboarding designs for example. Treat this as your design manual.

I'm already writing more chapters for you so please, make sure you check your email every now and then and look for my face in your inbox - I'll be sending out updates everytime I release something new.

I would also really appreciate if you could leave a short review with your rating about this e-book. It won't take you long but will help us improve and release better product in the future. Thank you so much! 🙏🏻

Share your opinion here

Clicking on the button will take you to complete a Google Form

Don’t forget about the rating on Gumroad!

If you enjoyed reading this ebook, I would really appreciate if you could leave a 5-star rating on my Gumroad page. We’ve spent over 600 hours, probably even much more on writing and designing it all and it would mean the world to us if you could leave a positive review 🙏

To do it, you can go to gumroad.com/library, log in if necessary and then find my e-book in there, click on it and it’ll take you to a page where you should find the ratings section. You can also click on the link below to go to the library directly without having to type it.

Go to Gumroad and leave a review

Clicking on the button will take you a Gumroad library

If you’d like to see a particular chapter written for the next release, please send me an email at hello@uiadrian.com or write me a DM on my Instagram @uiadrian. I’m waiting for your feedback friend.

More is coming soon!

This is just the begining of what we’ve prepared for you. I’m starting to work on another 9 chapters as well as a new educational piece, but this time in a different format - a video course.

It will go over everything mentioned in the practical part of this ebook to take you through the whole process of designing an iOS app from start to finish. I know that some of you might prefer a more visual presentation and the video course will complement this e-book, so I hope it’ll be able to dispel any doubts and questions that may have arised while you’ve been reading it.

If you have any suggestions what you’d like to see in a video course, what topics you’d like for me to cover, please send me a short message on my IG @uiadrian. I’ll take every suggestion under consideration to help improve and provide you with the most valuable content and real practical knowledge.

Thank you again for supporting my content. Even though I don’t know your name (yet!) I really appreciate your support. Don’t be a stranger, say hi on my Instagram if you’re reading this. I’ll be waiting for your messages!

For now, take care, and I wish you happy designing! 👋

Best,
Adrian

