

# Dark Bean

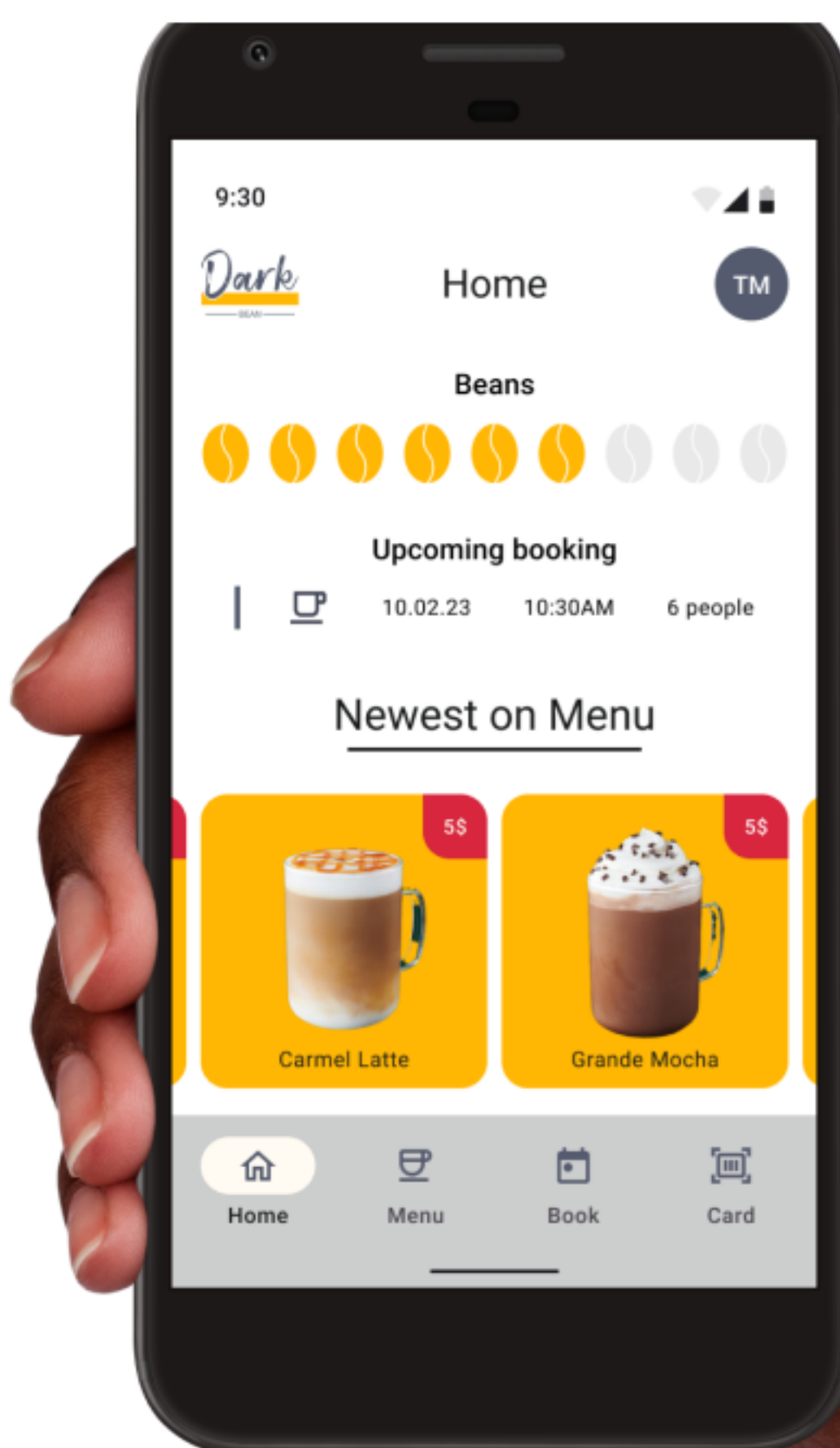
## The best booking app for a cafe

The idea for this project was to create the best app for the coffee shop, which will allow the users to book the table in their favorite place. The additional features of the app are the actual menu and loyalty program with a visible points counter and a virtual cafe visitor card.

## The goal

The main goal was to make a project that would be simple and quick to program, and would also solve the problem of the inconvenient way of booking seats in a cafe over the phone.

This project was made as a part of Google UX UI Certificate Course



Set the goals

Understand users

Create persona

user flow & wireframing

Mockups

PROBLEM FRAMING

SOLUTION

## User research

My user research was based on an interview with potential users.

When preparing the questions for the interview, I was focused primarily on remote employees looking for a convenient place to work. However, it turned out that people involved in organizing small group meetings are also often interested in booking tables in cafes. As a result, I had to expand the research to include questions about booking seats for more guests and problems related to the preparation of such a meeting.

## Pain points

1

### Long booking time

Reserving a seat takes a long time and involves calling the café.

2

### Rebooking

Changing the booking date is cumbersome and takes a long time.

3

### Table properties

It is not certain whether the reserved place will have everything that user needs to work.

## Personas



### Todd Steward

28 years old  
B. s.  
Single  
Programmer



### Anna Smith

38 years old  
M. a.  
Married  
Readers club president

### Goals

- I wish that booking a place to work did not require me to spend too much time.
- I would like to be able to choose a place to work without fear that it will be too loud or without power plugs

### Frustrations

- It bothers me when I am not sure what my workplace will look like and whether it will have all the necessary elements.
- I spend way too much time looking for a place to work

### Goals

- I wish the booking process was faster.
- I wish I could concentrate more on books than organizing meetings.

### Frustrations

- Sometimes it is impossible to call the cafe to reserve a seat
- Every rebooking takes a lot of time.



## User flow & Wireframes

Drawing conclusions from the conducted interviews and created personas, I define the user flow of the application.

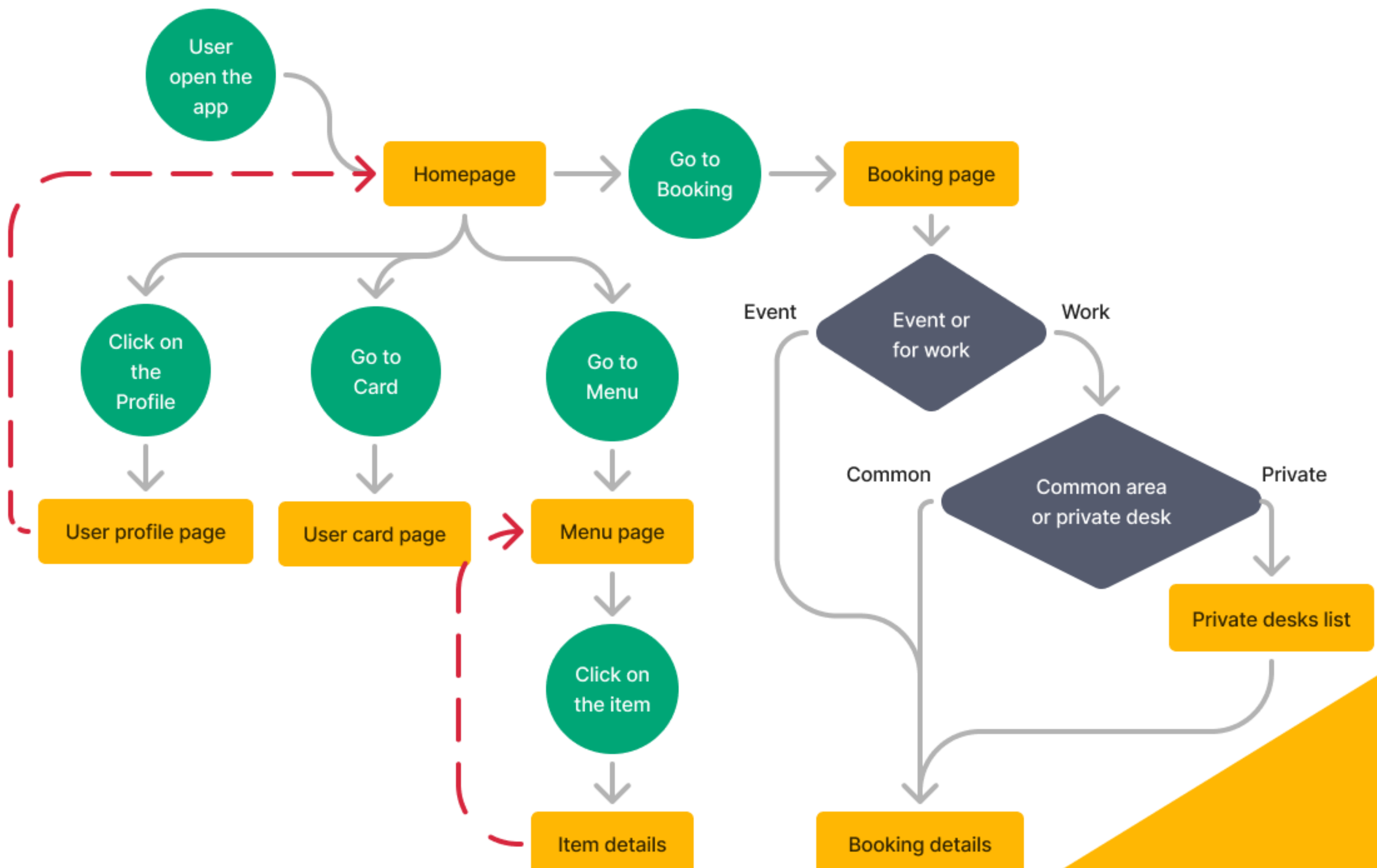
Creating user flow allowed me to determine the number of components needed. Thanks to that I was able to create wireframes of the entire application. During this process, I could notice and fix imperfections and the first corner cases of the app.

## User flow

The basic functionality of the application is the ability to book tables, which is why I write this flow first. Already at the first stage of booking, I decided to separate ordinary guests whose goal is to reserve a table for a meeting from remote employees. at this point, I assume that answering one additional question should not be a problem for a regular user. The remote employee has another choice to make, whether he chooses the free option, i.e. working in a common work area, or renting a private desk with additional amenities. The payment for

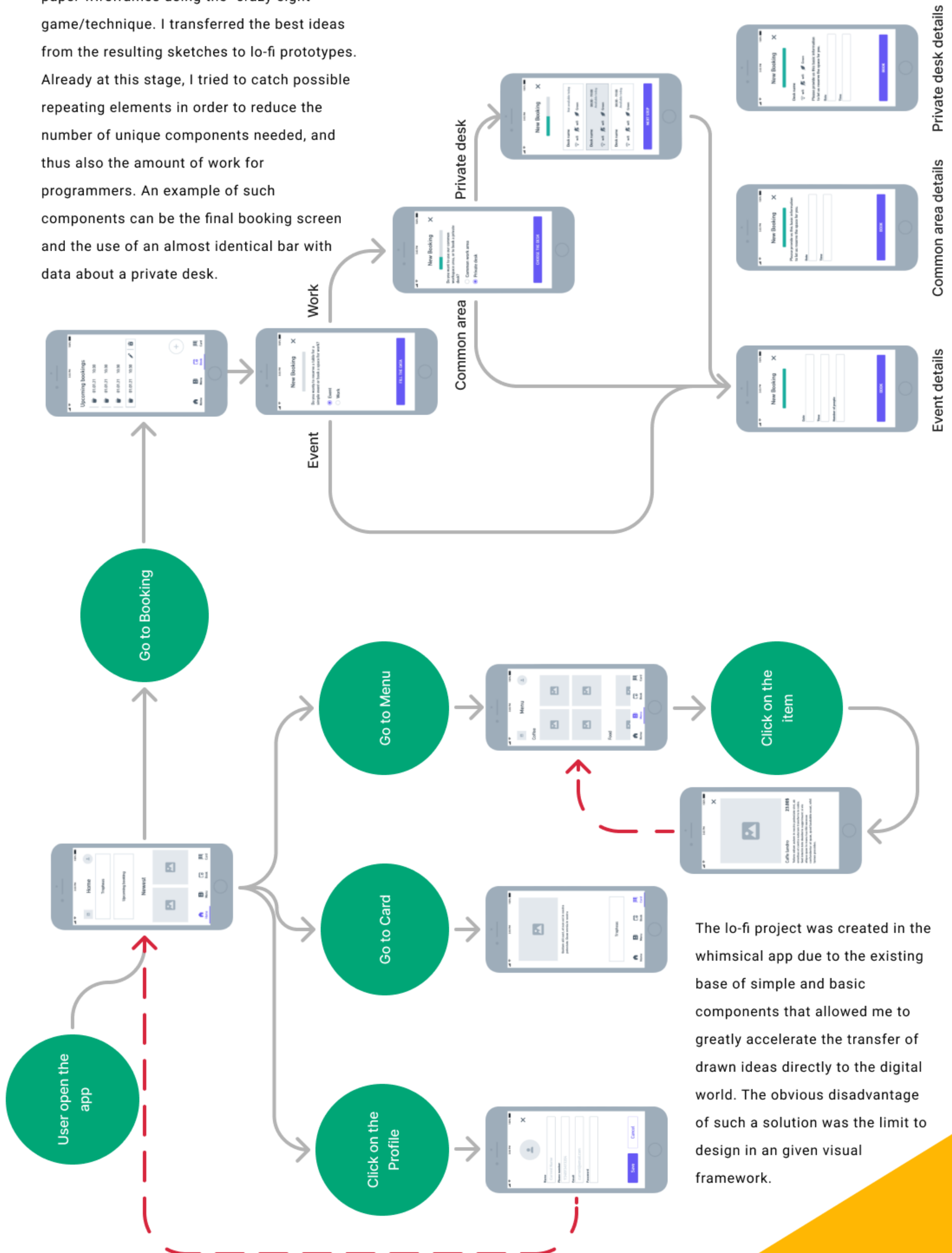
the reservation is already made on site at the premises, so I did not have to include this stage in the plan.

In addition to the reservation in the lower bar, the user will have at his disposal the current restaurant menu, access to the loyalty card with information about the currently held points, and a home screen with a shortcut of the most important information, which will also be the basic screen that the user will see after starting the application.



## Wireframes

I started designing components by creating paper wireframes using the "crazy eight" game/technique. I transferred the best ideas from the resulting sketches to lo-fi prototypes. Already at this stage, I tried to catch possible repeating elements in order to reduce the number of unique components needed, and thus also the amount of work for programmers. An example of such components can be the final booking screen and the use of an almost identical bar with data about a private desk.



The lo-fi project was created in the whimsical app due to the existing base of simple and basic components that allowed me to greatly accelerate the transfer of drawn ideas directly to the digital world. The obvious disadvantage of such a solution was the limit to design in an given visual framework.



## Hi-fi designs

My basic assumption was to minimize the work of programmers and keep it simple.

To achieve this, I decided to adopt Material Design 3 as my main framework. As a result, I transferred all ideas prepared in wireframes to the base components available in this library. To avoid the "bootstrap effect" (default-looking application), I diversified the appearance of the components with lively colors.

## Design system

### Why Material?



I decided on Material Design mainly because the main target of users will have the android system on their mobile phones. For this reason, in order to simplify the work of programmers, I decided not to hinder their work by implementing a whole new custom design system. It was also important for me that I don't know much applications that use the latest material design proposals. Because I like discovering new things, and the new appearance seemed very interesting to me, so I planned to explore and use it in my nearest project. Thanks to this, I could bite into this design system myself and see its solutions in action.

### Colors



Through my programming experience, I instinctively started with finding two basic colors, and a palette of matching auxiliary colors to use in information about successes, errors and important information. However, it turned out that the latest material guidelines do not recommend the use of colors in the snackbars. In this case, I decided not to remove these colors, but to use them to give the appearance of the app more dynamics.

### Typography

Display Large - Roboto 57/64 . 0  
Display Medium - Roboto 45/52 . 0  
Display Small - Roboto 36/44 . 0

Headline Large - Roboto 32/40 . 0  
Headline Medium - Roboto 28/36 . 0  
Headline Small - Roboto 24/32 . 0

Android + Web  
Title Large - Roboto Medium 22/28 . 0  
Title Medium - Roboto Medium 18/24 . +8.15  
Title Small - Roboto Medium 14/20 . +5.5

Android + Web  
Label Large - Roboto Medium 14/20 . +6.1  
Label Medium - Roboto Medium 12/16 . +5.9  
Label Small - Roboto Medium 11/14 . +5.5

Android + Web  
Body Large - Roboto 16/24 . +8.15  
Body Medium - Roboto 14/20 . +5.35  
Body Small - Roboto 12/16 . +5.0

Android + Web  
Caption

In the project I use typography proposed by Material Design. In order to be faithful to the basic resolution of simplicity, I decided to use the Roboto font, which is the base for android systems. Of course, in the case of transferring the application to iOS, I would decide to use San Francisco. The readability of the entire interface is based on the size of the fonts - the most important elements of the page use the largest font. In this project, I did not take into account the accessibility of adapting the screen to larger than regular fonts.

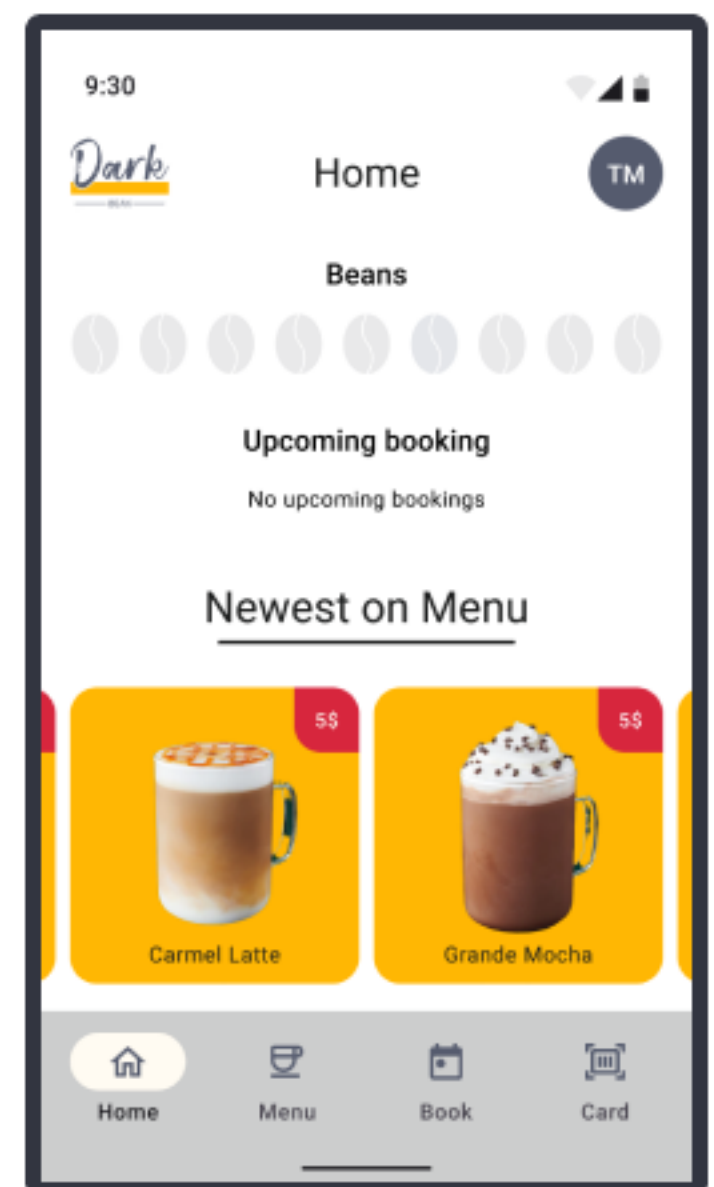
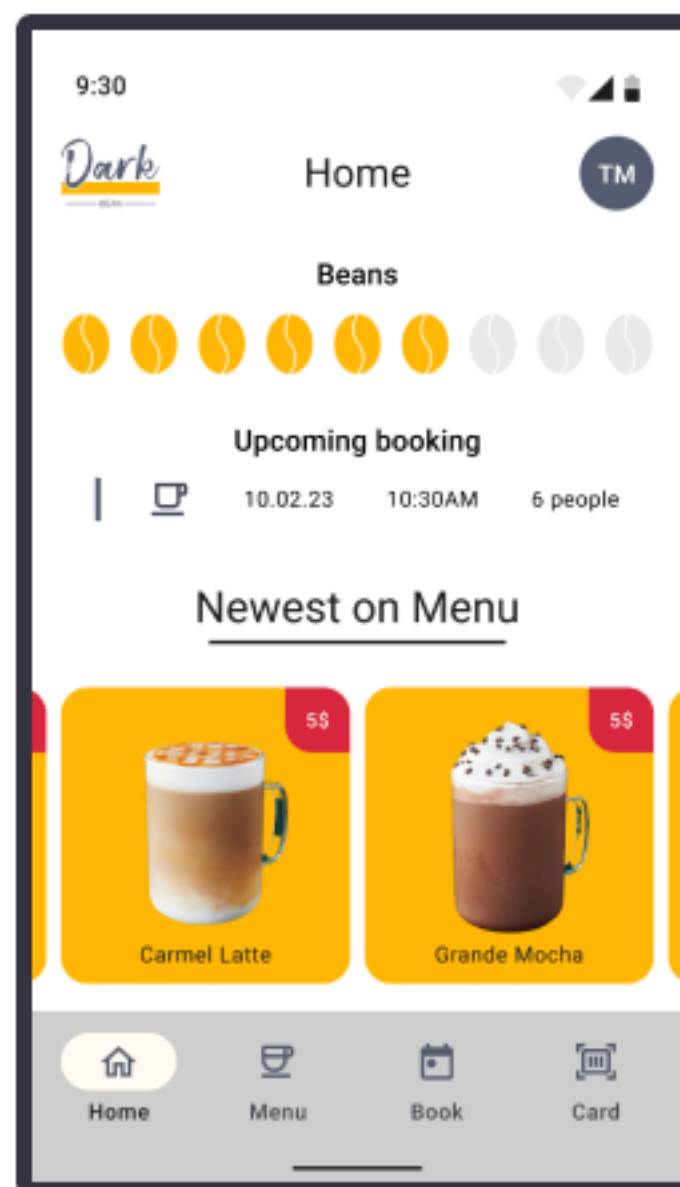
## Designs

User  
open the  
app



Homepage

This screen will show all the most important information in one place. The use of large gaps between elements allowed for the creation of a clear and clean layout. I also decided to highlight the list of new products here, because I thought that it would be overlooked by users who would reflexively direct their eyes to the most important information for them, such as the number of collected points or bookings. As for the functionality, only the nearest reservation will be visible in the upcoming reservations.



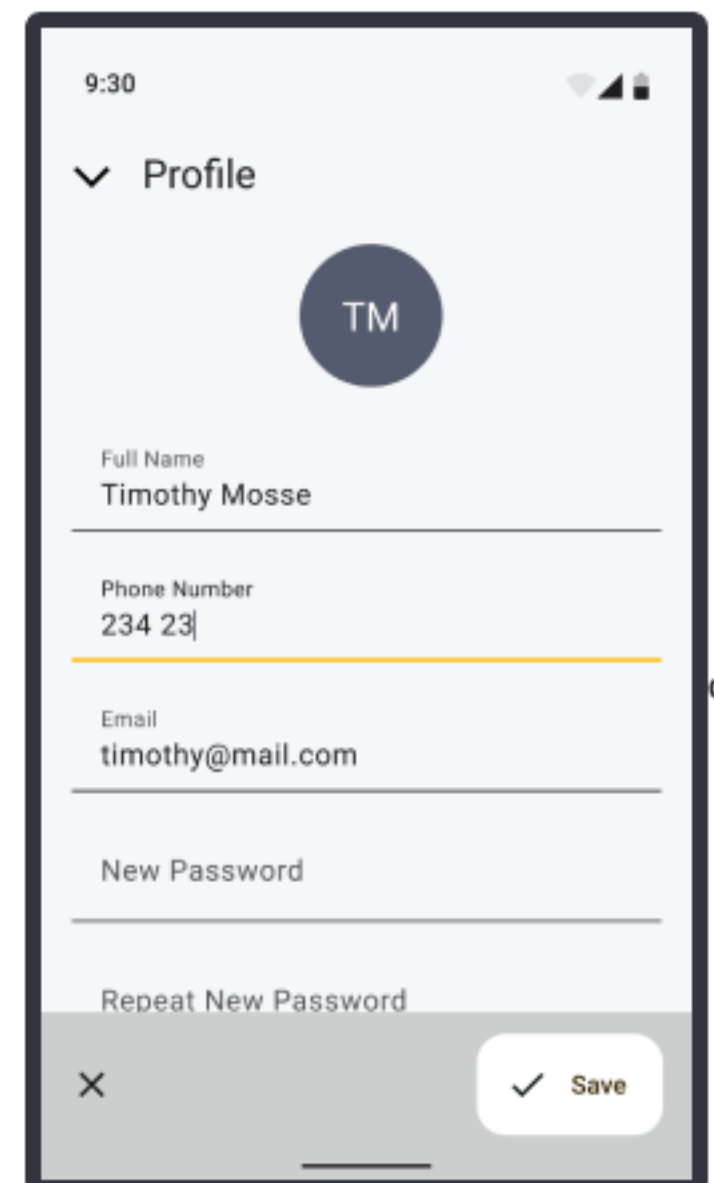
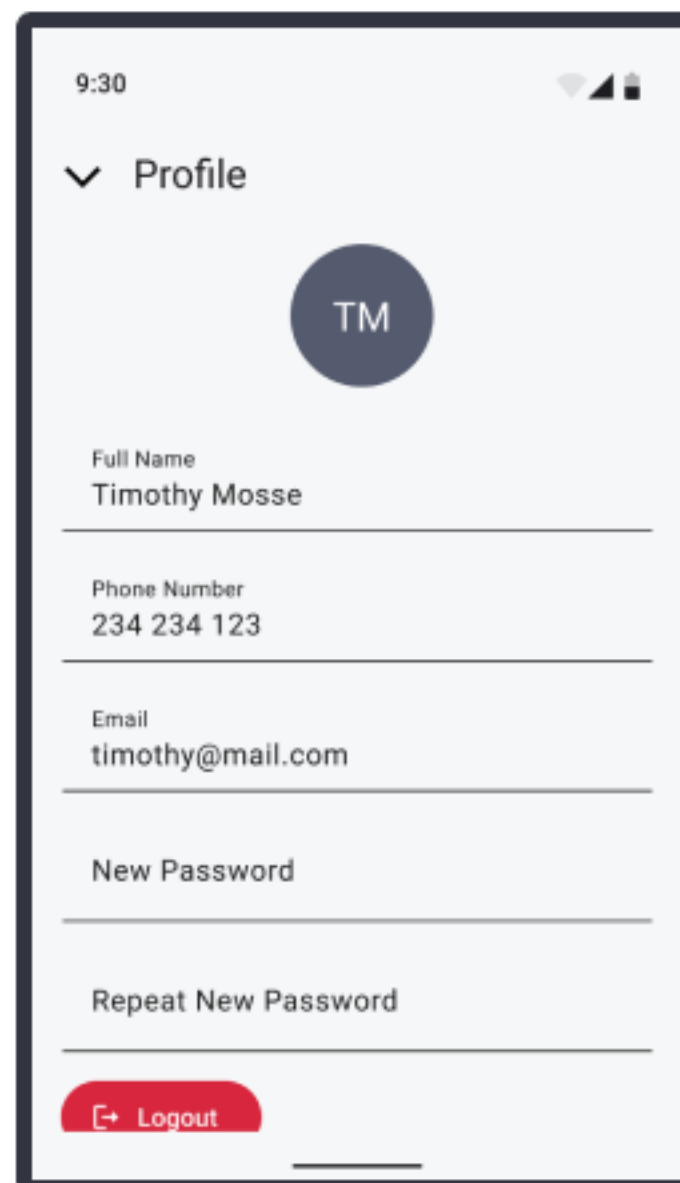
Default home screen

Click on  
the  
Profile



User profile page

This is the profile screen the user will get by clicking the icon on the top right of the screen. I've used a full-screen bottom sheet here to indicate only the functional and practical nature of this screen. I placed the tools necessary to edit the profile in the bottom app bar proposed by Material design. Here I also placed a button to log out of the application.



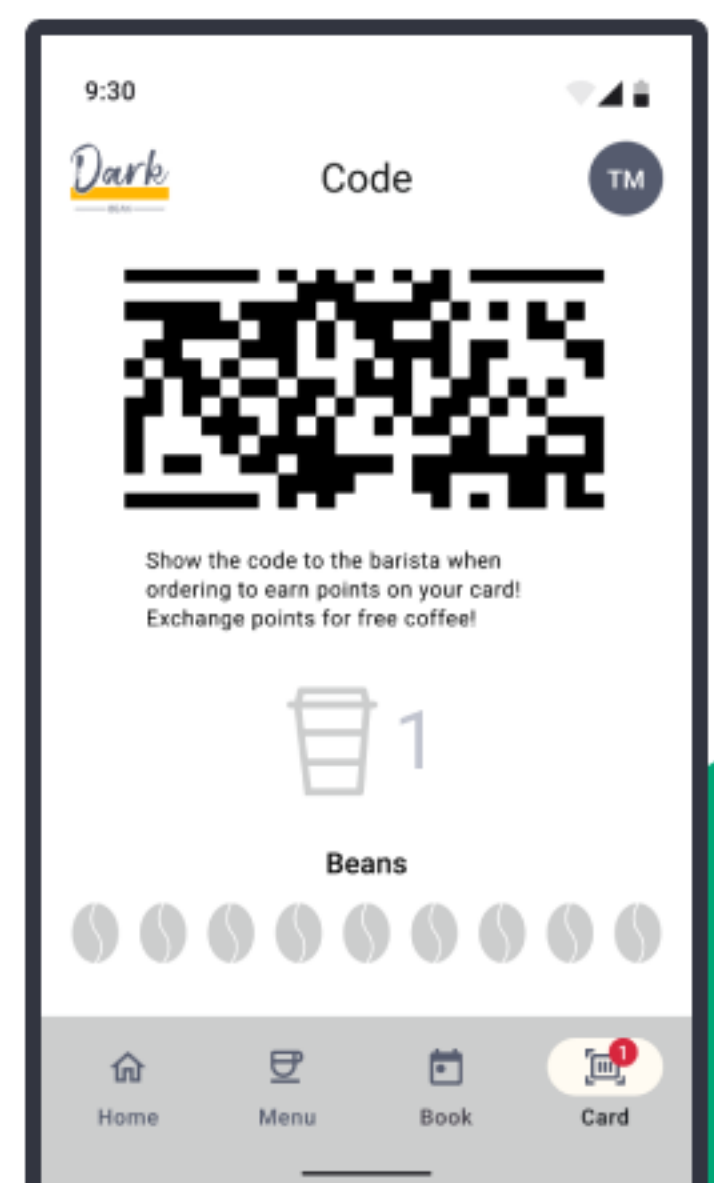
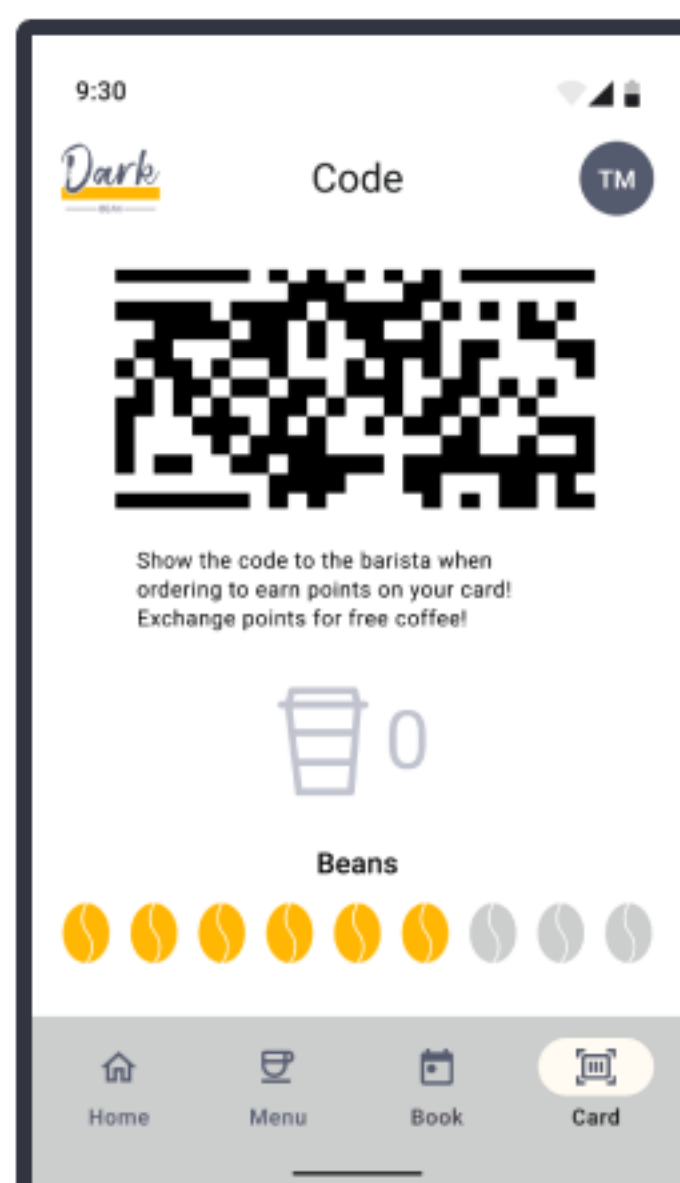
Profile in editing mode

Go to  
Card



User card page

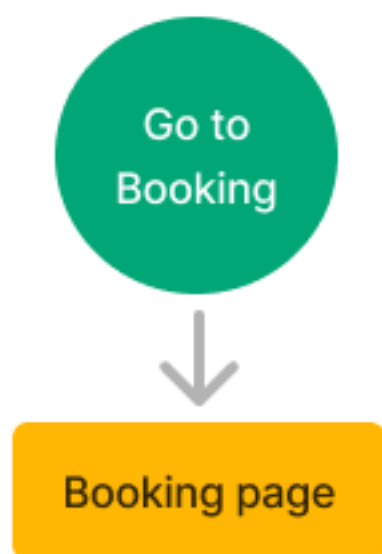
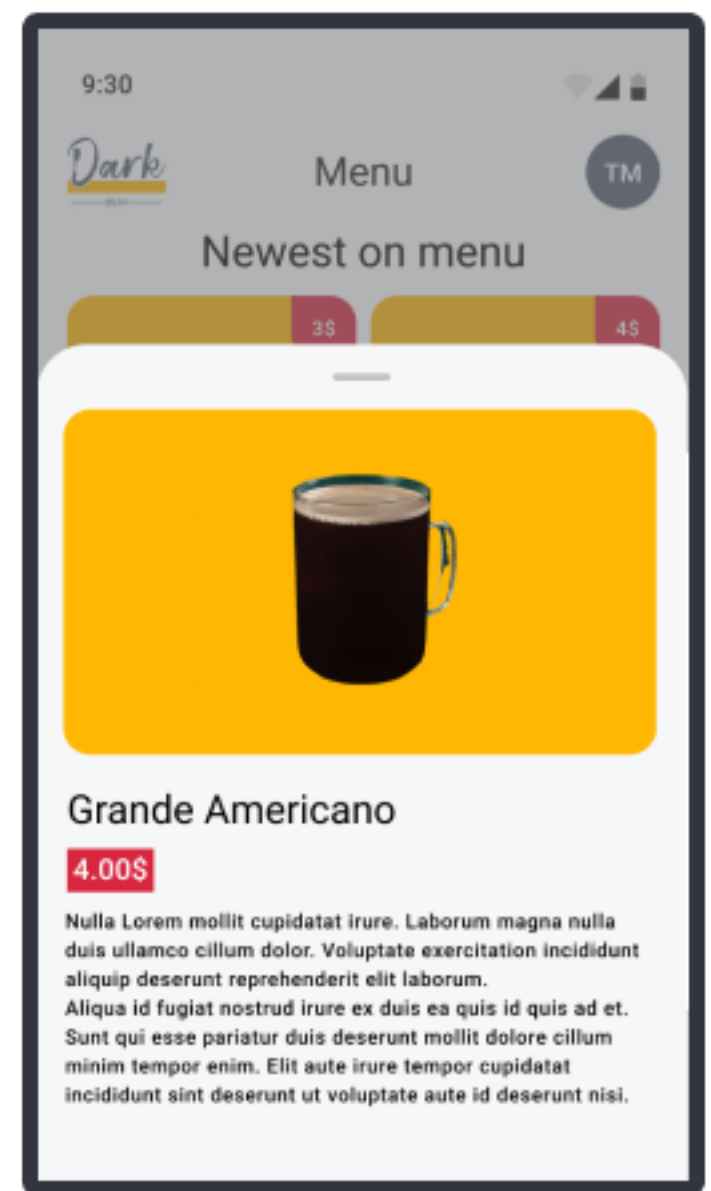
The card screen will probably be the most opened screen of the application, which is why its icon is located closest to the thumb. The main issue with this screen was the functionality related to collecting points. My idea to solve this problem is a coffee counter. To get the reward, the user must request a free coffee from the barista. Employee will scan the customer's qr code and marks the amount of received rewards.



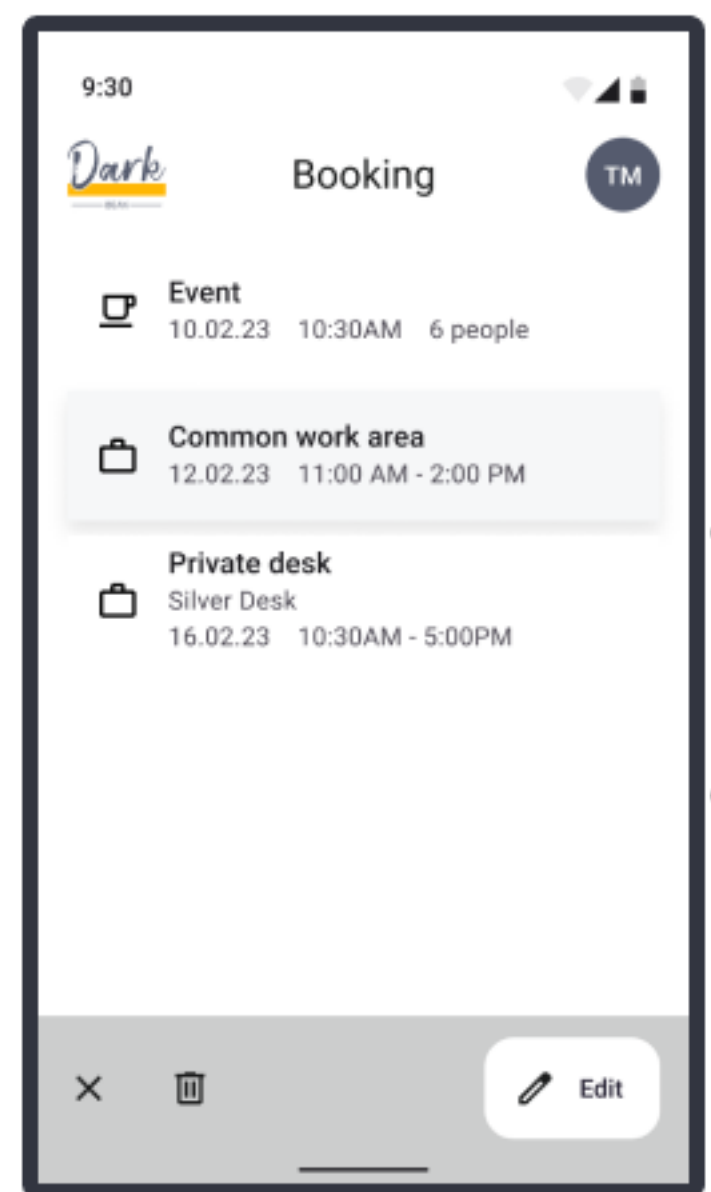
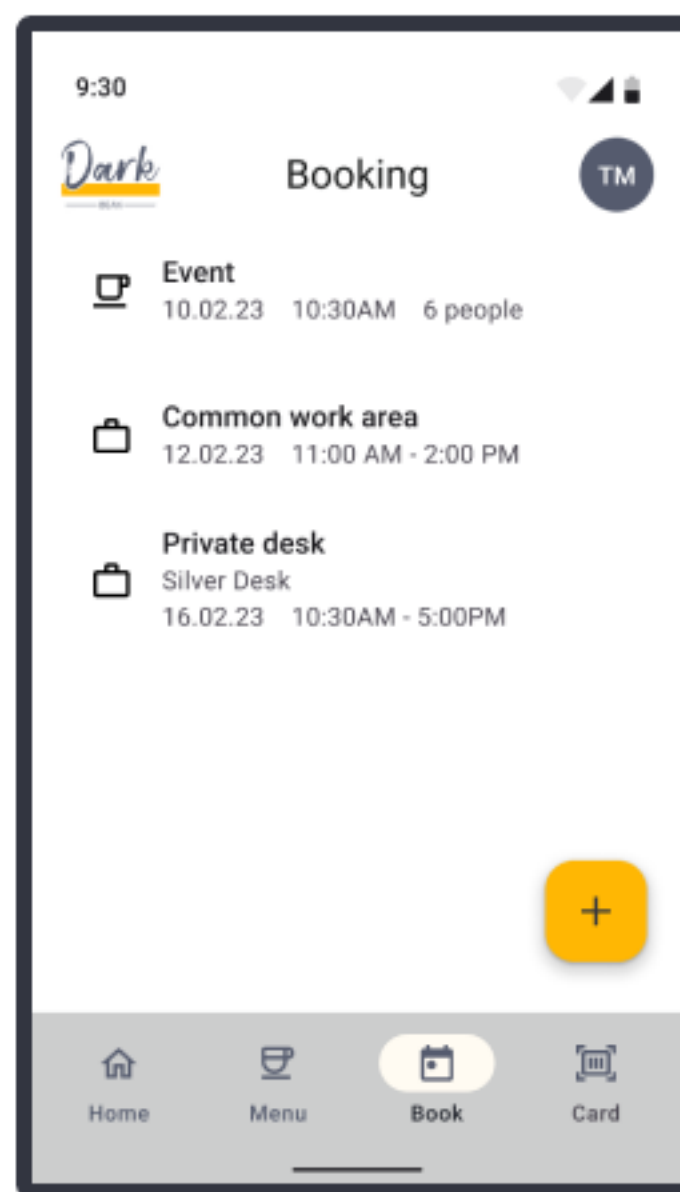




In the original idea for this screen, I did not include product pricing information. With the help of internal tests, I realized this shortcoming and corrected my mistake. In addition to the user experience, the design also gained, because red added dynamism and freshness to the list of products, so that the screen no longer looks boring and default. This screen also shows the bottom sheet, which allows the user to read a more detailed description of the product.

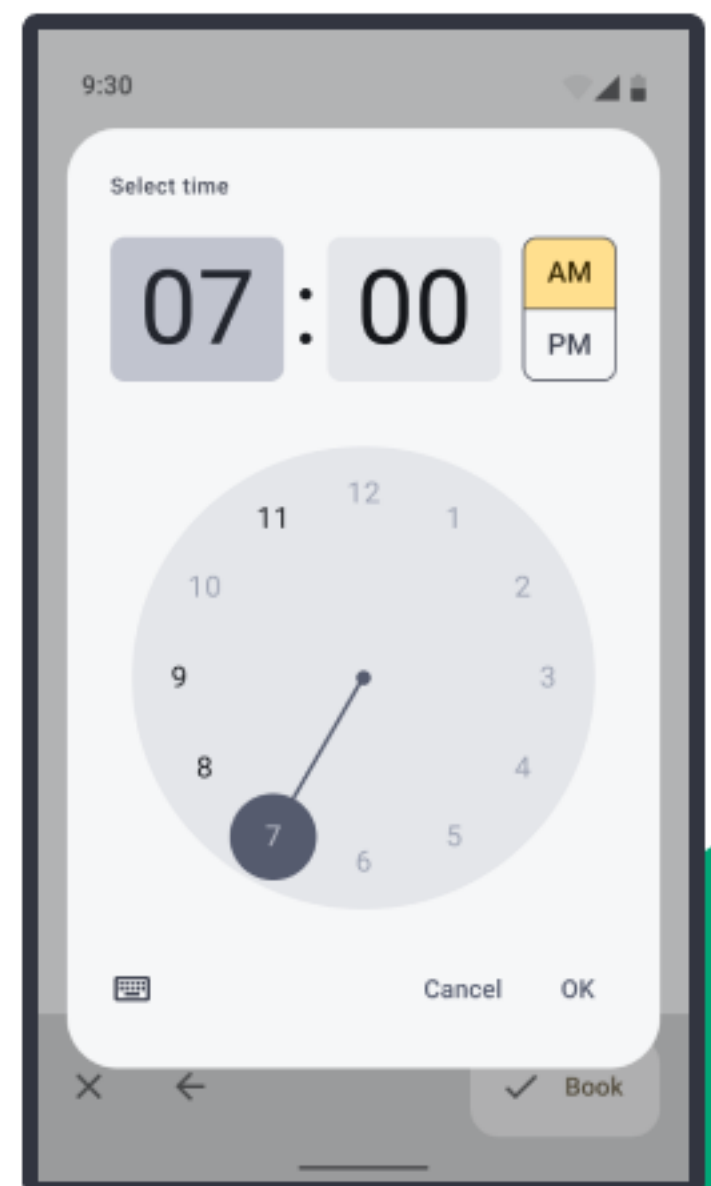
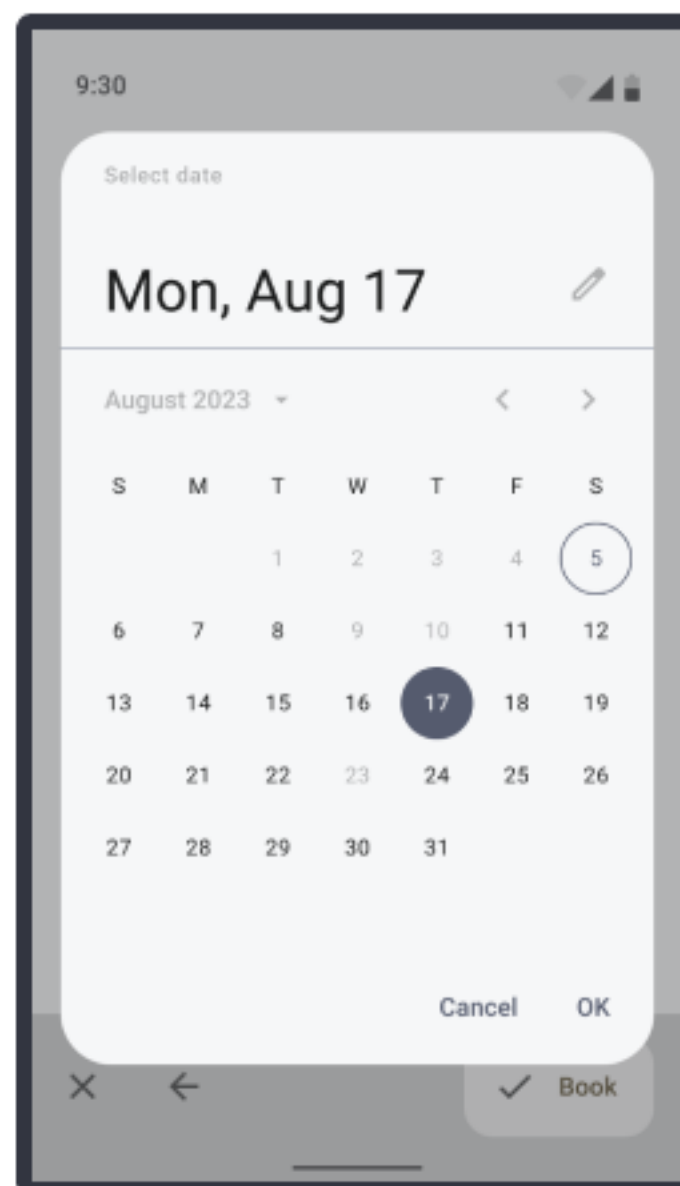


This is the screen that allows you to start creating a new reservation (through the strongly highlighted FAB with a plus icon). The booking list on this screen uses a basic list component from Material Design. In order to go to the edit mode, the user must hold the given reservation longer. User can only edit one booking at a time. This screen also shows the refreshed square FAB, which is not available in edit mode, which simplifies the logic of the application.



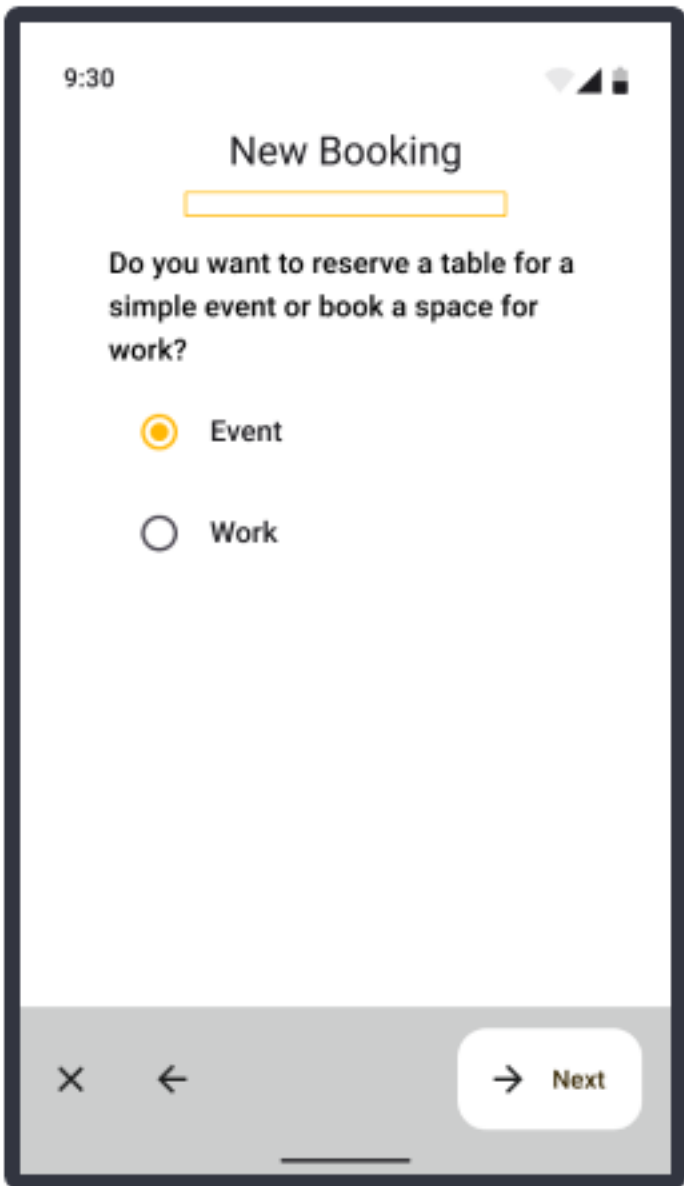
Booking list in editing mode

Date and time pickers are generic. I didn't feel the need to customize anything except the colors. I know from experience that dealing with date and time logic is hard enough already, so any aesthetic change will only double the torment. As for the logic mentioned, I think that reservations are only possible on full and half hours (10:00, 10:30, 11:00, 11:30, ...). Of course, you cannot make reservations for a date before the current day. In addition, I think it would be possible to set a booking limit for common work areas and regular events, so that the work of the cafe is not disrupted by too many remote employees, or by booking all available tables for guests.

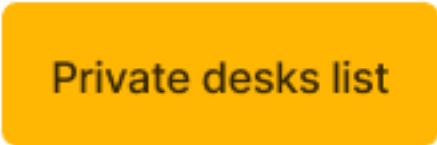
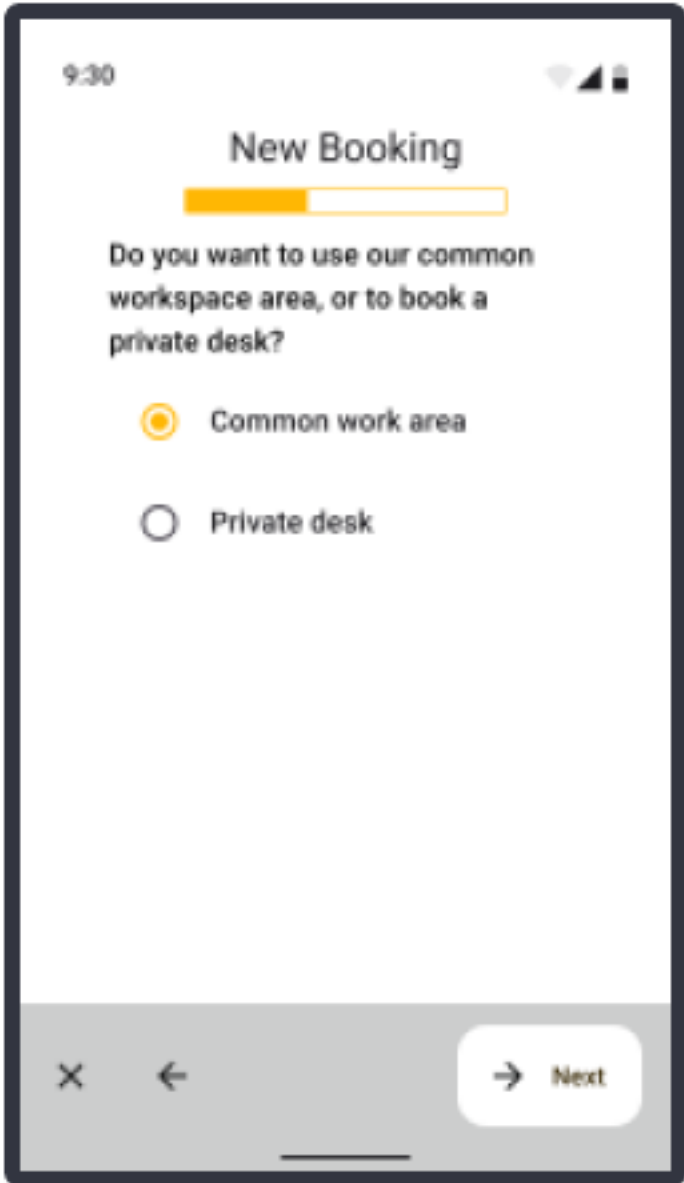




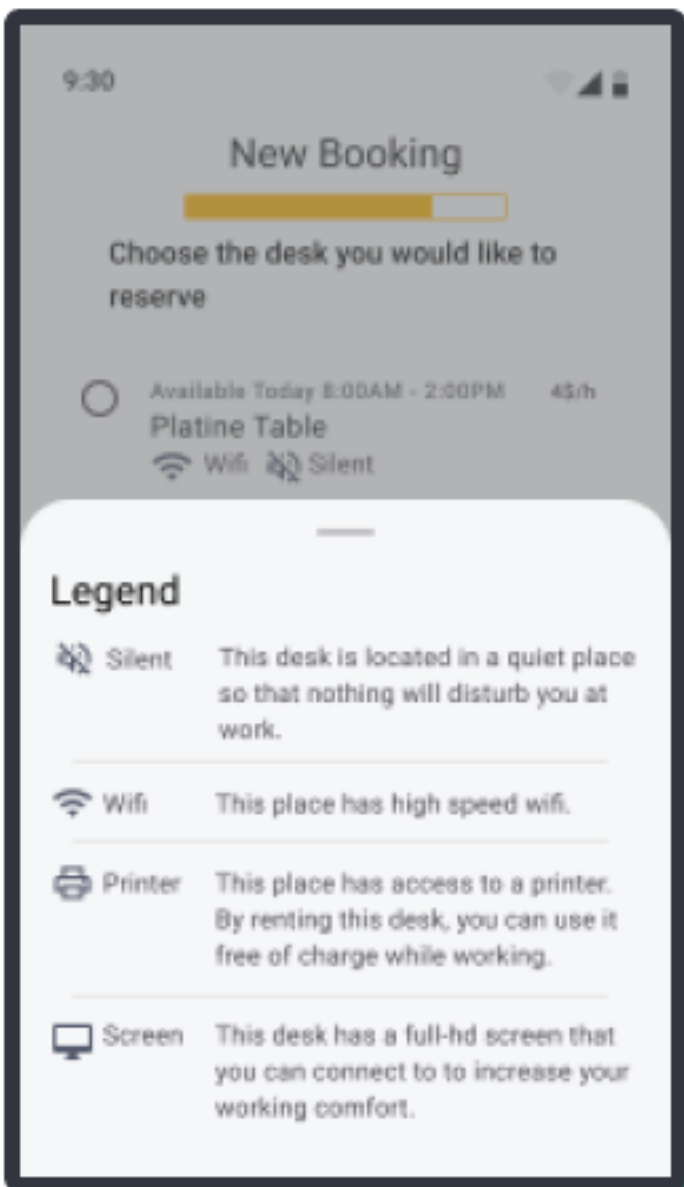
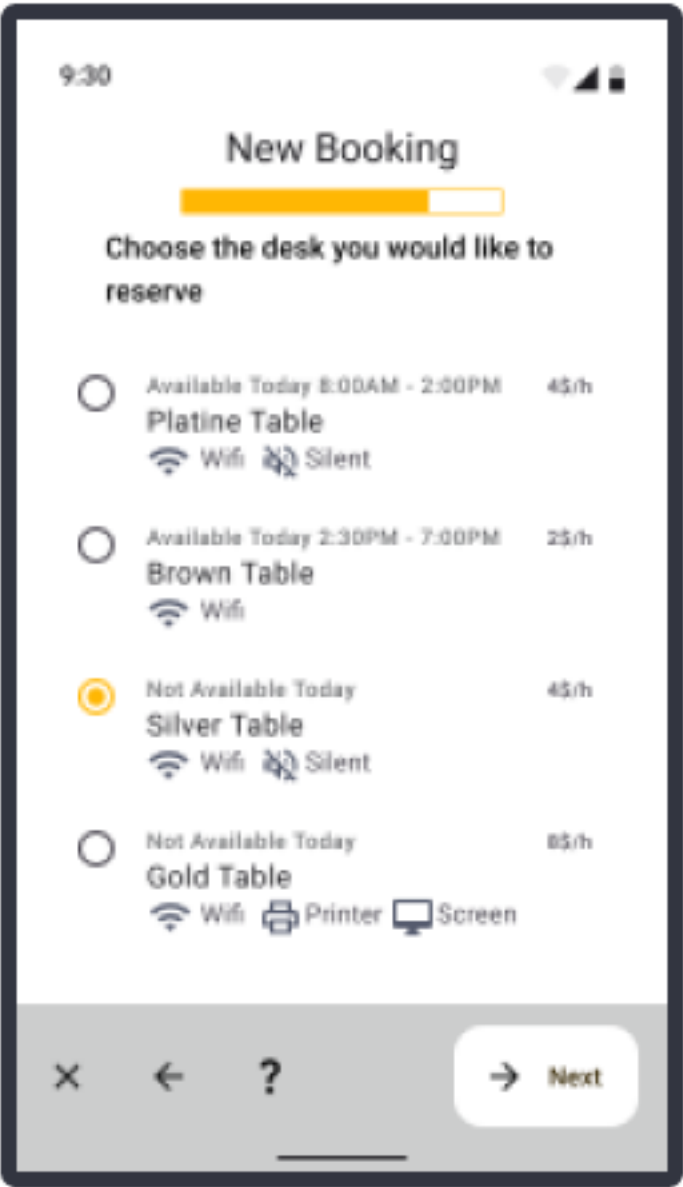
This is the first step of creating a reservation. The user decides whether he wants to book a place to create an event (e.g. a meeting of two friends over coffee) or to work remotely using the work area in the cafe. I used a radio buttons here because of their readability. By default, the event option will be selected, thanks to which the process of booking a place for ordinary customers has been maximally shortened and simplified. I found adding this step clearer and simpler than separating the two processes by using two separate FABs in the booking list.



This step only applies to remote workers. At this point, the user must choose between a free work zone and paid desks that have additional features. It reuses radio button here, and the default value is common area. Maybe that's why fewer users will check the offer of private desks, but I decided to focus on user comfort here, not increasing profits.



Again, I'm dealing here with a list created from the basic list component available in this framework. I used radio button again to visualize the selection and to keep the user's selection transparent. Internal tests drew my attention to the ambiguity of desk features. That's why I created a simple legend that will be shown after clicking on the question mark at the bottom of the screen. An additional feature here is information about the availability of a given desk on the day when the user makes a reservation.



The legend hidden behind the question mark



Booking details

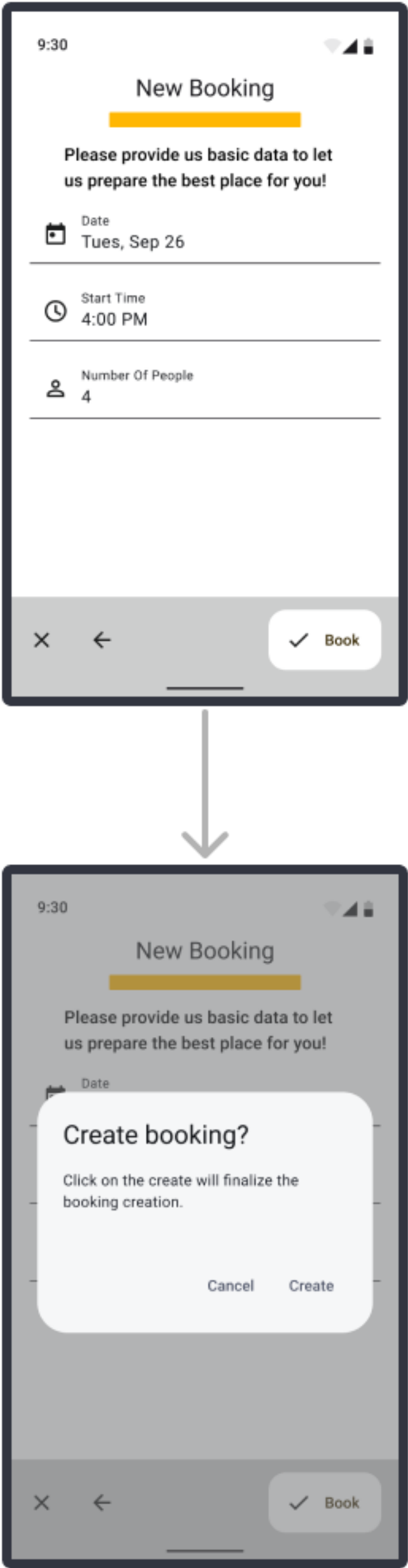
Checkout is almost identical for each road.

This screen differs significantly for the private desk option, where the price and information about the method of payment for the desk appear. I think that for the sake of simplicity, it can be assumed that when the user enters this screen for the first time, all fields will be empty and the displayed price will be \$0.

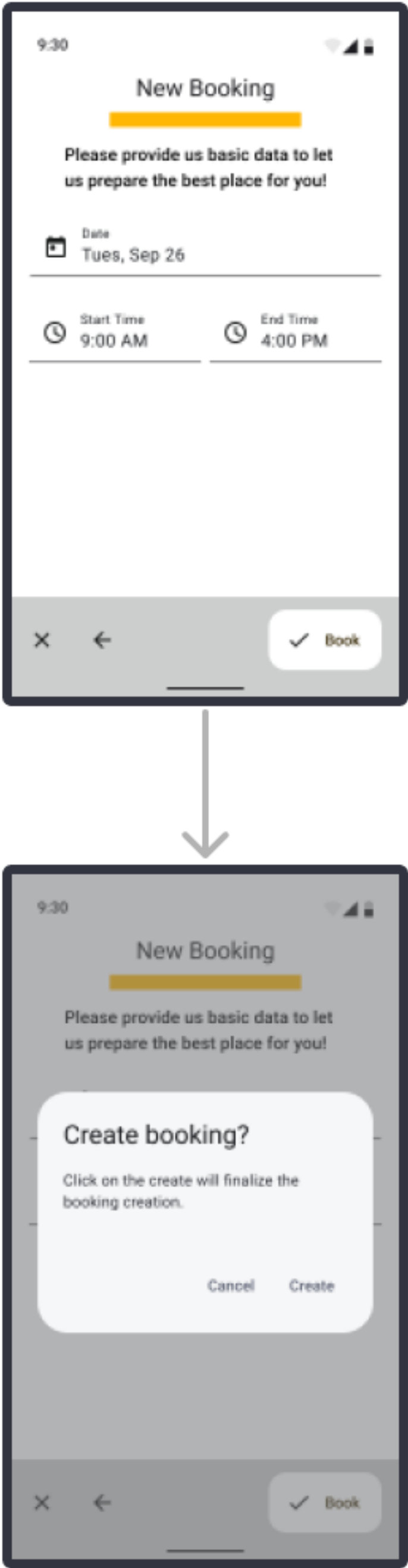
Before creating a reservation, a dialogue appears. It may not be necessary, but I believe that it gives the user a sense of security, plus it is an additional protection against accidental clicking to create a reservation button.

From the UX perspective, it is also important that action button changes. This together with the progress bar at the top lets the user know that he is at the end of the process.

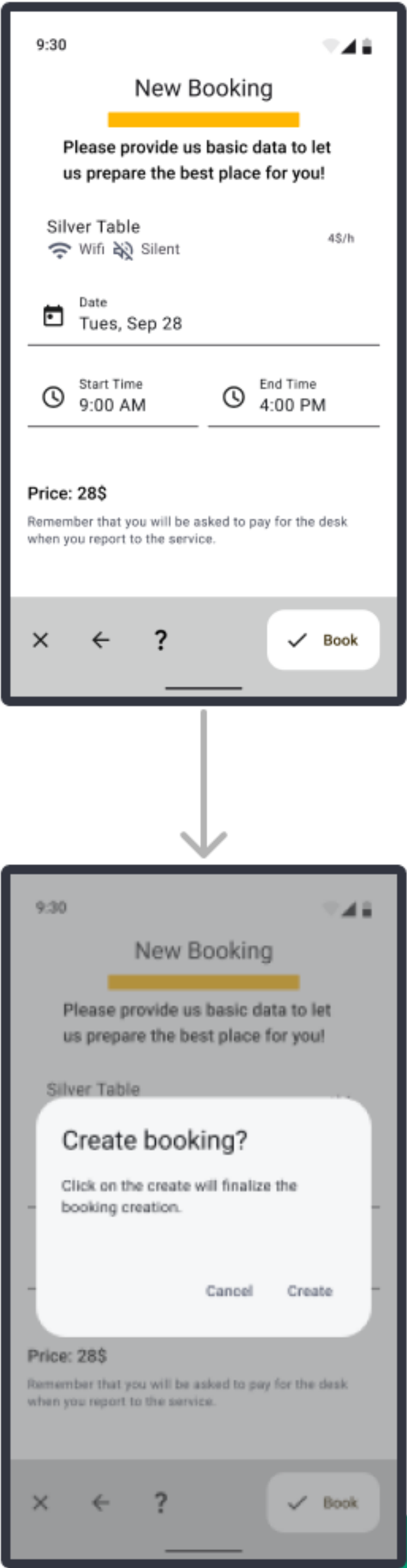
Event booking checkout



Common working area booking checkout



Private working desk booking checkout



## This is the end

Thank you for your time!

Going through this document is quite a feat, especially because of the mass of texts and descriptions - although I must admit that the design of the document often set me a limits on how much I can write. Anyway if you still have enough strength and motivation, on this last page I put my ideas, plans and visions of what should happen next with this project.

## Future ideas

- The next step should be external testing of this application to be able to catch errors that are not noticeable to me now.
- It would be useful to prepare the project in terms of development. So, in addition to writing down the colors and styles, I should also prepare some information and examples of components that will be used throughout the app.
- An important issue that has been slightly overlooked is the topic of the application on the side of the cafe. It would be useful to create its design, taking into consideration the whole dynamics of contact between the cafe and its customers.
- It would be a good idea to create an iOS version of this application, taking into account the default font for this system and various changes that will allow this application to better adapt to the nature of this system.