Mobile App Development Assignment #02

05/05/2023	Group 10	Mads Cornelius Hansen coha@itu.dk

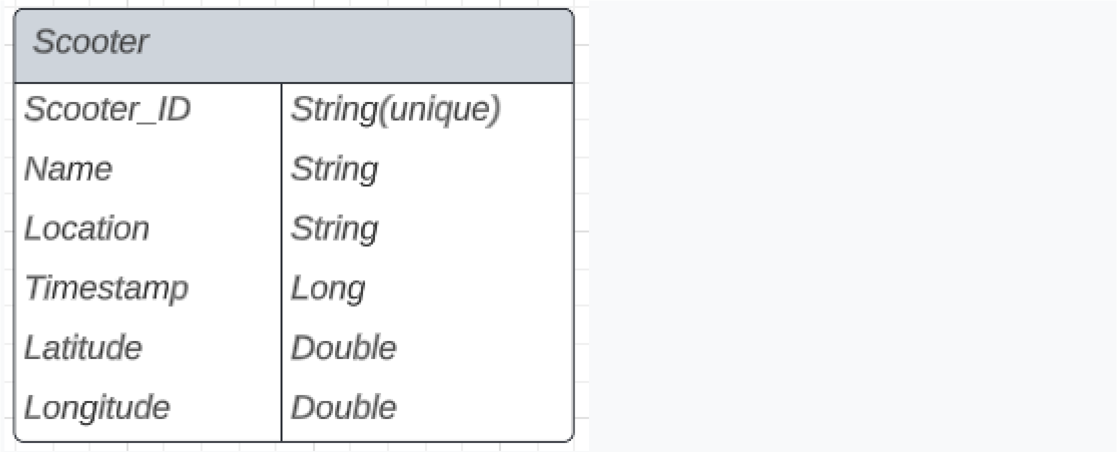
Design choices

Layout

My layout is designed such that I have one activity showing bottom navbar and top bar with a fragment's container in between. This means that all pages in my app are fragments which I change by using the findNavController().

Database

My Realtime Database is hosted on firebase and only has a single table consisting of scooter which consists of scooter objects. Like shown below on figure 1 a scooter object is built with a unique ID, a name, location which is the closets address in string form, Timestamp and the latitude and longitude coordinate values.



Scooter	
Scooter_ID	String(unique)
Name	String
Location	String
Timestamp	Long
Latitude	Double
Longitude	Double

Figure 1 Scooter ER diagram

App structure

Currently my app is built up by a Login screen (1), List of all scooters (2), google maps (3), QR scanner (4), scooter profile (5), rent a scooter (6), camera (7) and park scooter (8) see figure 2.

The login page (1) uses the firebase authentication to validate any users using the app. Once a user logs in it will navigate them to List of all scooters (2). On the page List of all scooters

(2) all scooters are displayed with name, location, and image. It's possible for a user to swipe a scooter to see its profile page or to navigate to google maps (3) or QR scanner (4) via the bottom nav bar. On the google maps (3) page shows the users and their current location and shows all scooters as pins and geofences(feature) as orange circles. The QR scanner (4) can scan any of the QR codes supplied in the QRCodes folder and navigate to the belonging scooters profile page (5). scooter profile (5) displays all the content of the chosen scooter, this includes picture, name, and location. On the bottom of the profile page are two buttons which navigate back or to rent a scooter (6). Once the user has confirmed that they intend to rent that scooter the user will be taken to rent a scooter (6). This page is similar to profile page (5) but has a price and a timer filed which constantly updates both. This page also only has one button which stops the renting of the scooter and leads to camera (7). On the camera (7) users can take a picture of where they parked the scooter which will be saved for the next user. Once the picture is taken the user is taken to park scooter (8) here the scooters new picture and locations is displayed. The users can choose to retake the image or park the scooter there, where after they are navigated back to the main page.

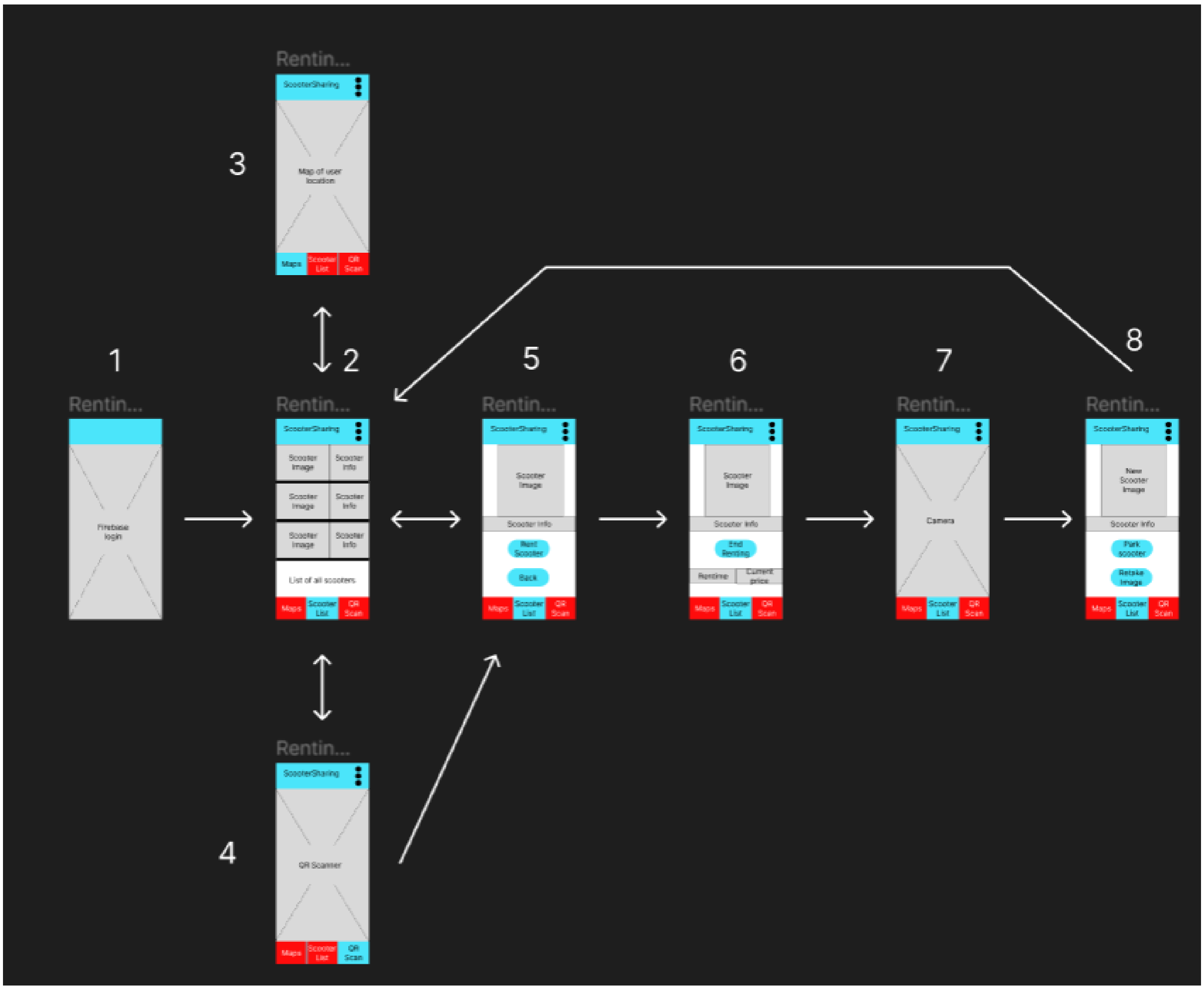


Figure 2 Image of current app layout

Here are the links leading to the Figma page where the layout is displayed.

Figma layout : https://www.figma.com/file/wotJWOJjm8T7wfrfXakeak/ScooterSharing-Draft?node-id=102%3A280&t=CdkMDlzciHFXuI32-1

Figma prototype : https://www.figma.com/proto/wotJWOJjm8T7wfrfXakeak/ScooterSharing-Draft?node-id=102-396&scaling=min-zoom&page-id=102%3A280&starting-point-node-id=102%3A396

User guide

Rent a scooter.

Any numbers like (2) references Figure 3

To rent a scooter there are two ways of getting started. Either swipe a scooter on the scooter list or scan one of the QR codes.

From there you are led to the scooters profile page, here the user should press the Rent Scooter button (2). Here you will be promoted with a dialog on if you want to rent a scooter (note if you are not inside a geofence a debug dialog will come first allowing you to overwrite the limit)

Then the user is navigated to renting a scooter (3) here the price and time of how long the scooter has been rented is displayed. Once you wish to park the scooter and pay press the button on that page (3). Here you will be promoted with another dialog on if you want to end rent the scooter (note if you are not inside a geofence a debug dialog will come first allowing you to overwrite the limit).

Once the button is pressed the phone's camera is opened so they can take a picture of where the scooter is parked. Once the image is taken the user is navigated to the final step (4). Here the scooter's new picture and address is displayed and only once the park scooter is pressed the scooter's location and image will be updated and the renting of the scooter has ended.

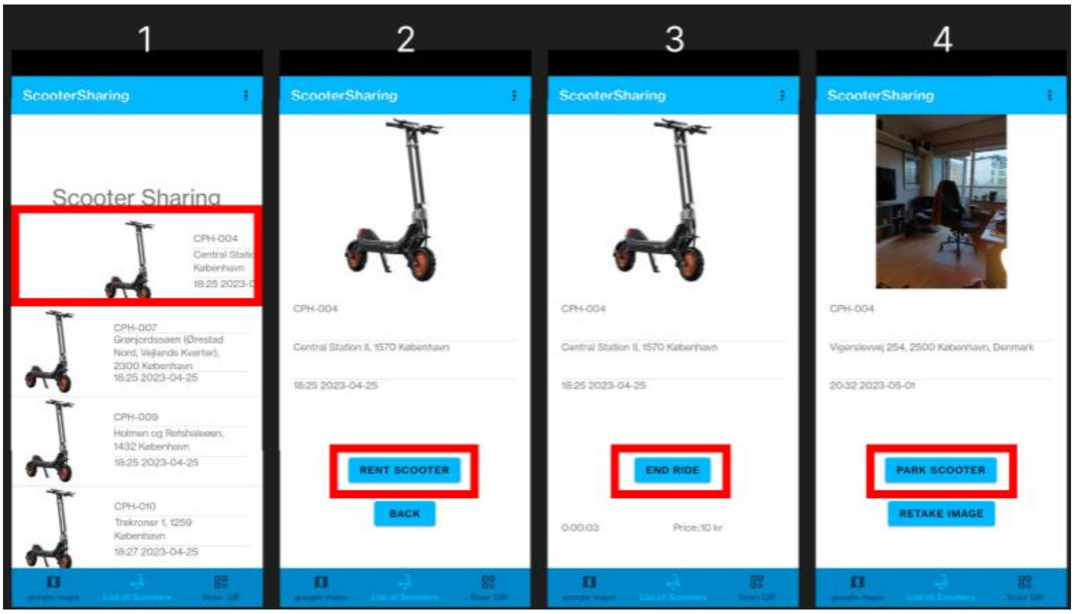
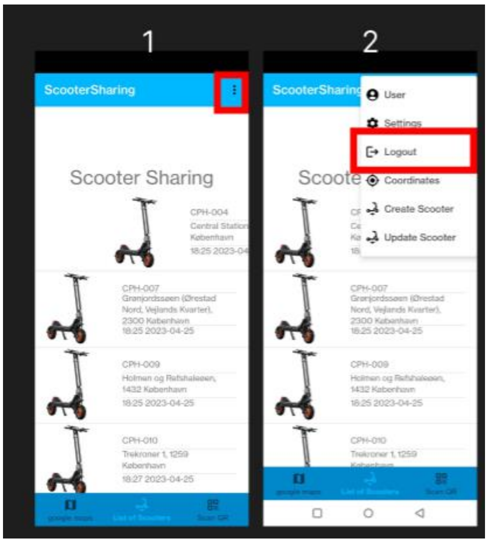


Figure 3 walkthrough of renting a scooter.

Logout

To logout press the 3 dots in the top right corner where the logout option will be presented.



Google Maps

The google maps can be access via the bottom nav bar. And should display geofences and scooters.

QR Scanner

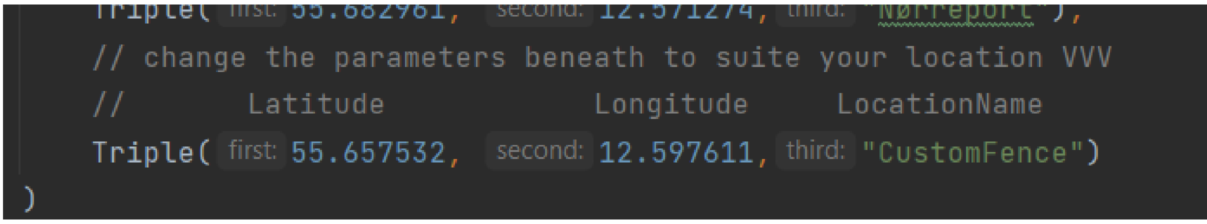
The QR Scanner can be accessed via the bottom nav bar. And should scan the codes quickly and then navigate the user to the belonging scooter page.

Feature

As my feature I implemented geofencing which means I register when a user enters og exits a certain area. I decided to use it as an allowed parking zone, meaning that scooters can only be rented or parked inside of these zones. The zones should be marked on the google maps page but please note that sometimes the maps and the zone don’t match precisely so move closer or further toward the center to get it to react. I have a small toast the pops up when you enter or exit a geofence. But your geofence status can also be found by pressing the Coordinates button inside the top right burger menu.

How To add geofence

Inside the folder utils there is a file called GeoFenceFile. This file is responsible for all related to the list of geofences and their location. If you want to add an extra geofence go down and find the *locationList.* At the bottom is a custom geofence, change the values to fit where you want a geofence or add your on Triple. It's also possible to change the radius of a geofence by changing the value var *geofenceRadius* at the top of the file.



```
    Triple( first: 55.682961,  second: 12.571274, third: "Nørreport"),
    // change the parameters beneath to suite your location VVV
    //       Latitude             Longitude     LocationName
    Triple( first: 55.657532,  second: 12.597611, third: "CustomFence")
)
```

Problems

Turning the phone

I have a problem when a user turns the phone no matter where they are navigated back to the main page with the list of all scooters. I know I can solve this by saving the current fragment inside the savedInstanceState and then reset the phone state onResume or onStart. But because of time constraints it was a fix / feature that I decided to down prioritize to make sure the I met the minimum requirements.

Users enter the google map too quickly.

This Is a small problem that is not fatal but if the user enters the google maps before it knows the user's location, they are placed on Bahamas. When google maps knows the location it is no longer a problem and works fine.

Scooters can be rented from any geofence.

While I only allow uses to rent scooters when they are inside a geofence, I do not consider the users distance from the scooter. Meaning that the scooter could be in an entirely different geofence than the user. Again because of time constraints the was a feature I didn't implement, but I did think of two possible solutions. One would be to calculate the distance between the user and the scooter and set a limit of 10 meters. Since I can get the user and scooter coordinates, I would just have to calculate the distance between two points. The second option would be to give each scooter then name of the geofence they are parked in. then I could compare the users geofence and the scooters to see if they matched.

Address sometimes being a wrong address.

This is more of a problem caused by the API than my code. I'm using the API geocoder.getFromLocation which works completely fine most of the time and gives an address nearby. But sometimes it returns an address a couple of streets away and in the most extreme case I found was 10 km away. I don't know why it only does it sometimes but I'm assuming it's a problem on the ape's side.

Random crashes

I have no clue what causes this, sometimes when I press a button, turn the scene, or change fragment the app crashes. If I try to replicate my actions the app works fine. It doesn’t happen often, just sometimes. Just thought it was worth mentioning.

QR codes

All QR code can be found inside the project folder called QRCodes in png format.

Original plan vs finished layout.

My original plan is shown in figure 4 and my finished layout figure 5.

I completely overestimated how much I would be able to complete before the handin. While I also didn’t follow my original plan with having a rental history, renting multiple scooters and an entire account section. I did however keep the bottom navbar, but it navigates between google maps, scooter list and QR scanner instead of Renting options, rental history and currently renting.

I did however also add a burger menu which allows the user to log out and if implemented could navigate the user to an account page.

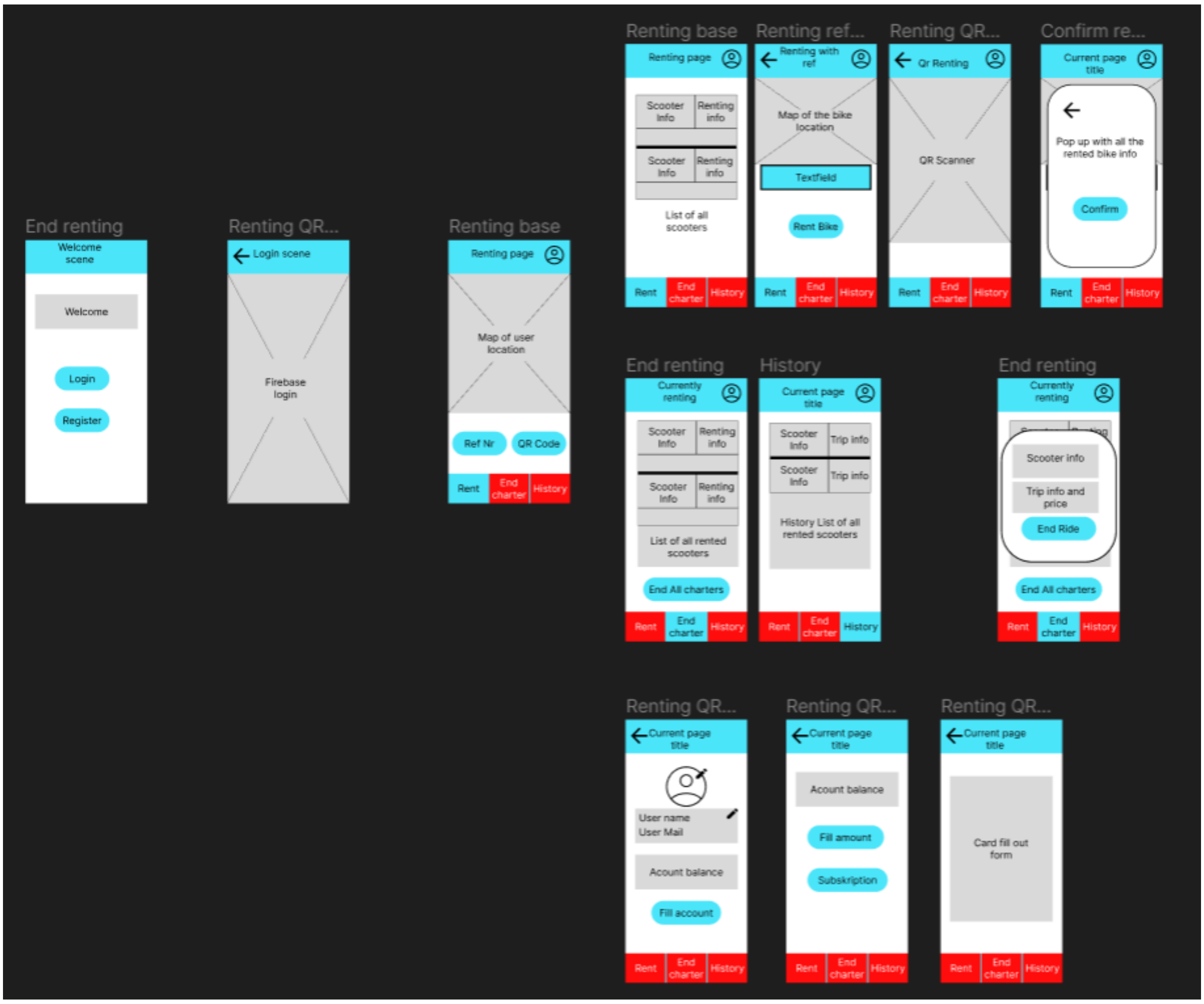


Figure 4 Original app layout plan

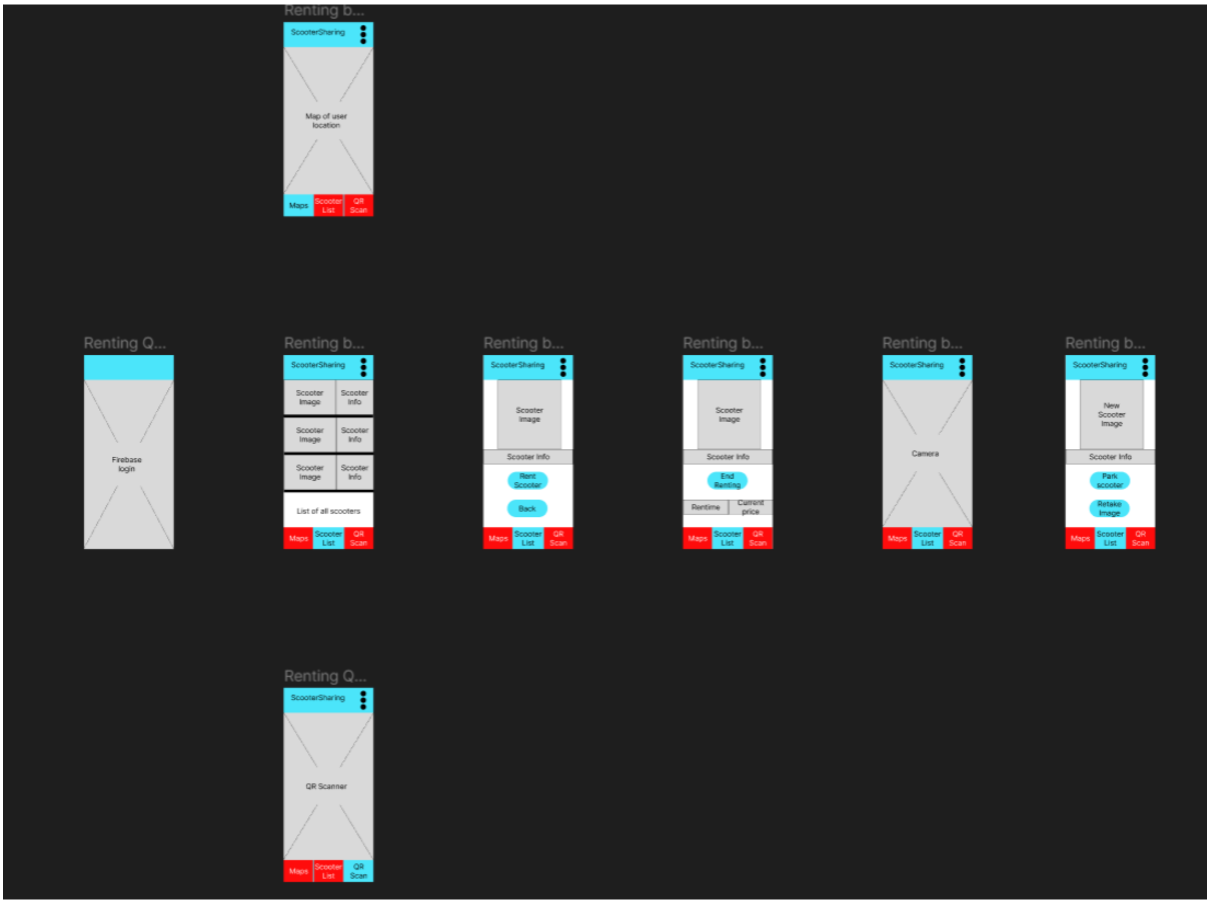


Figure 5 finished app layout.