

# MOAPD Assignment 1

Alaa Abdul-Al & Joachim Alexander Kofoed

March 2023

## 1 Design Choices

We have all of our classes, fragments etc. in `src/main/java/dk/itu/moapd/scootersharing/<ITU user>`. To keep resource files organized, they are all put in the `/res` folder. In here the different resources are separated in subfolders. For example, strings are put in `/res/values/strings.xml`, colors in `/res/values/colors.xml`, vector icons in `/res/drawable/`, and UI content in `/res/layout/`. For the content on the main page, we have chosen to put it into a `content_main.xml` instead of the `activity_main.xml`. This is to separate the content of the page from potential other things.

We are using `LinearLayout` most places. We have chosen not to use `ConstrainLayout` so far because it is more complex to use and because we do not need it. However, we may end up using it at some point during the development of the project as some pages may get more complex than the pages we currently have.

## 2 User Interface

We have opted for a very simple and user friendly user interface (UI). There is a landing page with three buttons: *Start Ride*, *Update Ride*, and *List Rides*. When clicking the *List Rides* button, a list of all rides stored in the data structure gets displayed. When clicking the button again, this list disappears. Optimally, when clicking the *List Rides* button, a new page should be shown with a list of all rides. This is due to the fact that the database might have a lot of rides stored, and hence there could be a need of scrolling through these. To make this look as pretty as possible, displaying this in the landing page below the buttons is not the best option. Due to time constraints, we have not been able to develop this feature. When clicking the *Start Ride* button, a new page is displayed with two fields allowing the user to start a new ride by inserting name and location and clicking *Start Ride*. In order to go back to the landing page, the user will have to swipe right. This is a built-in feature in android phones. Lastly, when clicking on the *Update Ride* button, a new page will be displayed with the option to only update the name of the Scooter being used. Updating the location is not a possibility.

A link to our figma design page can be found [here](#).

### 3 Future User Interface

We plan to have a total of six pages, namely: *Log in/register*, *landing page*, *city map/rent scooter (with qr code)*, *current scooter/ride*, *my profile*, *rental history*. We try to keep the design and interaction with the app as simple as possible due to time constraints.

The idea is that you start by accessing the login page, which will include a *register* text button to take you to the register page. When logging in, the landing page will consist of some buttons that can take you to the different pages. By clicking on the *start ride* button, you go to the city map page, which will show a map of all scooters with a button to scan a QR code to start ride. Furthermore, the the landing page will have a button which can take you to a page which shows you information about your current scooter/ride. This will however be considered during the course. Maybe we end up making this as a popup when clicking on a scooter on the map. A profile page will also be made, in which you can set personal information, as well as updating your payment details. Lastly, there will be a page that shows all your past rides.

### 4 Extensions

We have made some extensions to our Scooter Sharing app. Most of our extensions have been UI related. We have made a big focus on making our Scooter Sharing app look similar to that of the screenshots in the exercises. Some extensions are the icons in the text fields and buttons as well as placeholders in text boxes. We have also had a focus on having a user friendly and aesthetically pretty app. So, we have made sure that buttons, text fields, etc. are aligned and self explanatory. We have used the Material Design website a lot to make sure that these extensions have been designed correctly and in the best way.

### 5 Test and Evaluation

To test our app, we have used both the emulator built into Android Studio as well as a real phone. We have used the emulator to test out small increments to quickly see the new changes. After several small increments we have used the real phone to test the app to see if it works properly on a real device. In order to test if the app works properly on different screen sizes, we use the emulator to test the app on different phones. This is an important aspect of developing an app.

### 6 Problems

We have had some problems implementing the adapter as well as working with Constraint Layouts. The constraint layout would have been more efficient to use, however it simply took too much time to understand and implement. Hence we opted for the Linear Layout as described earlier. Another problem we believe we faced is the fact that we unfortunately do not have time to do the challenge exercises due to us being busy with our Bachelor Project. Hence, we do not have so many eye catching extensions except the ones mentioned.