Chalmers University of Technology

OOP Projekt

Requirements and Analysis Document (RAD)

Rent A Car - Stulb

Albert W, Hannes T, Jamal M, Johan S, Josef N
24/10-2021
v.2.0

1. Introduction

The project aims towards making renting cars easier and better by providing one platform all over the world. A platform where users can both rent out their own private car, rent someone's private car or for companies to reach out to more and new customers through an application easy to use for all. In today's society owning your own car has become less and less convenient where different circular car owning options have become more popular. This has created a need for an app where you can easily rent a car for a shorter amount of time like a day or a longer time if needed. By enabling private persons and companies to list their cars we create a competitive market where the users are made sure to get the best prices and lowest fares. Part of the purpose is to make car-sharing better and more convenient so that less people will be in need of buying a car so that we can lower the amount of lighthouse gas being emitted into the atmosphere.

There are a lot of beneficiaries of Rent a Car with foremost people who are in need of renting a car but also car rental firms and people that own one or multiple cars which are not used all the time. It will facilitate for both domestic and foreign travellers as the app could be used worldwide and therefore gathering cars to rent in one platform. Another one of the most important beneficiaries is the climate, which through reduced new production of cars will avoid the cost of more emissions into the atmosphere.

1.1 Definitions, acronyms, and abbreviations

The following is a list of words used throughout this document and their explanations:

GUI ”Graphical User Interface” also ”User Interface”, the part of the program that the user sees and interacts with.

User Story short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system.

API ”Application Programming Interface”, an interface or communication proto-col between a client and a server intended to simplify the building of client-side software.

Hi-Fi ”High fidelity”, a sketch with much detail.

UML ”Unified Modeling Language”, A visual language for mapping and repre-senting systems. Can be used for modelling och class diagrams, sequence diagrams and more. In this project used for Domain and design model.

DoD "Definition of Done", a list of requirements that have to be met for a User Story to be considered done.

2. Requirements

2.1 User Stories

This subsection lists the user stories that define the application’s requirements, in terms of the end-user’s wanted functionality.

2.1.1 UserStory1: Epic - Done

Story Name: User Story - Epic

Description:

(Epic) As a User, I need to be able to see the cars are available to rent so that I can rent a car to do what I need a car to travel.

Confirmation:

The user can see a list of available cars, choose one and then rent it.

2.1.2 UserStory2: Done
Story Name: Search

Description:
As a User, I need a search feature so that I can search and iterate between different types of cars.

Confirmation:
- The user can Search for specific types of cars.
- The user can Search for different car brands.

2.1.3 UserStory3: Done
Story Name: Account Page

Description:
As a car rental firm, We need a page so that we can overlook our account and our ads so that we know what we are renting out.

Confirmation:
- The user can scroll through his ads.
- The user has one page for his account where he can see and change his information.

2.1.4 UserStory4: Done
Story Name: Car Owner

Description:
As a user, I want to be able to put my car up for rent so that I can make money when I don't use my car.

Confirmation:

- The user can advertise his car
- The user can add images to the ad
- The user can set a price for the rental
- The user can post their contact information
- The user can add an item

2.1.5 UserStory5: Done
Story Name: Customer Order

Description:

As a Customer, I need to place an order so that I know if I can rent a car.

Confirmation:

- The user can save his order and come back to it later.
- The user can change his order before I pay for it.
- The user can see a running total of the cost of what he has chosen so far.

2.1.6 UserStory6: Done
Story Name: Personal Information

Description:

As a user of the app, I need to be able to create an account and have my personal information saved so that I do not need to re-enter all information.

Confirmation:

- The user can create an account.
- The user can save personal information, including cards etc.

2.1.7 UserStory7: Done
Story Name: Car Details

Description:
As a user, I want to be able to read and write details about the car so that I can make an informed choice about the car I will rent.

Confirmation:
- The user can open a detailed view.
- The user can read a description.
- The user can see an image of the car.
- The user can get personal information about the renter.

2.1.8 UserStory8: Done
Story Name: First Time User

Description:
As a first time user I want to be able to search for cars and navigate / look through the app without the need to register an account.

Confirmation:
- The user does not need to be logged in or register just to scroll through the app.

2.1.9 UserStory9: Done
Story Name: Frequent User

Description:
As a frequent user, I want to be able to stay logged in so that I don't need to log in every time that I use the app.

Confirmation:

- The user stays logged in when they exit the app/switch views.

2.1.10 UserStory10 : Done
Story Name: Upload image from desired location (phone)

Description:
As a user, I want to be able to upload images of my car from my phone's own photo library so that I can show pictures of my car to rent it out.

Confirmation:
- The user is able to upload an image of choice from their phone library to the ad.

2.1.11 UserStory11 : Done for prototype .
Story Name: Show featured cars nearby

Description:
As a user, I want to easily see cars that are available in my local area so that I might rent a car close to where I am located.

Confirmation:
- The user is able to see featured cars listed near the Users location.

2.2 Definition of Done

The Following is the DoD used in the project:

1. GUI Implemented.
2. The code is compilable.
3. All tests are successful.
4. All changes have been committed and pushed to the master branch and the master branch is running without error.

2.3 User interface

In the beginning of the project a preliminary sketch of the user interface was created to help with the visualization of the program (See figure 1,2,3).

The application is built with three major views: home, browse and profile. The bottom menu makes it easy for the user to switch between the three different views at any time. Navigate to the home view at the bottom left, the browse and search view in the middle and finally the profile view at the bottom right.

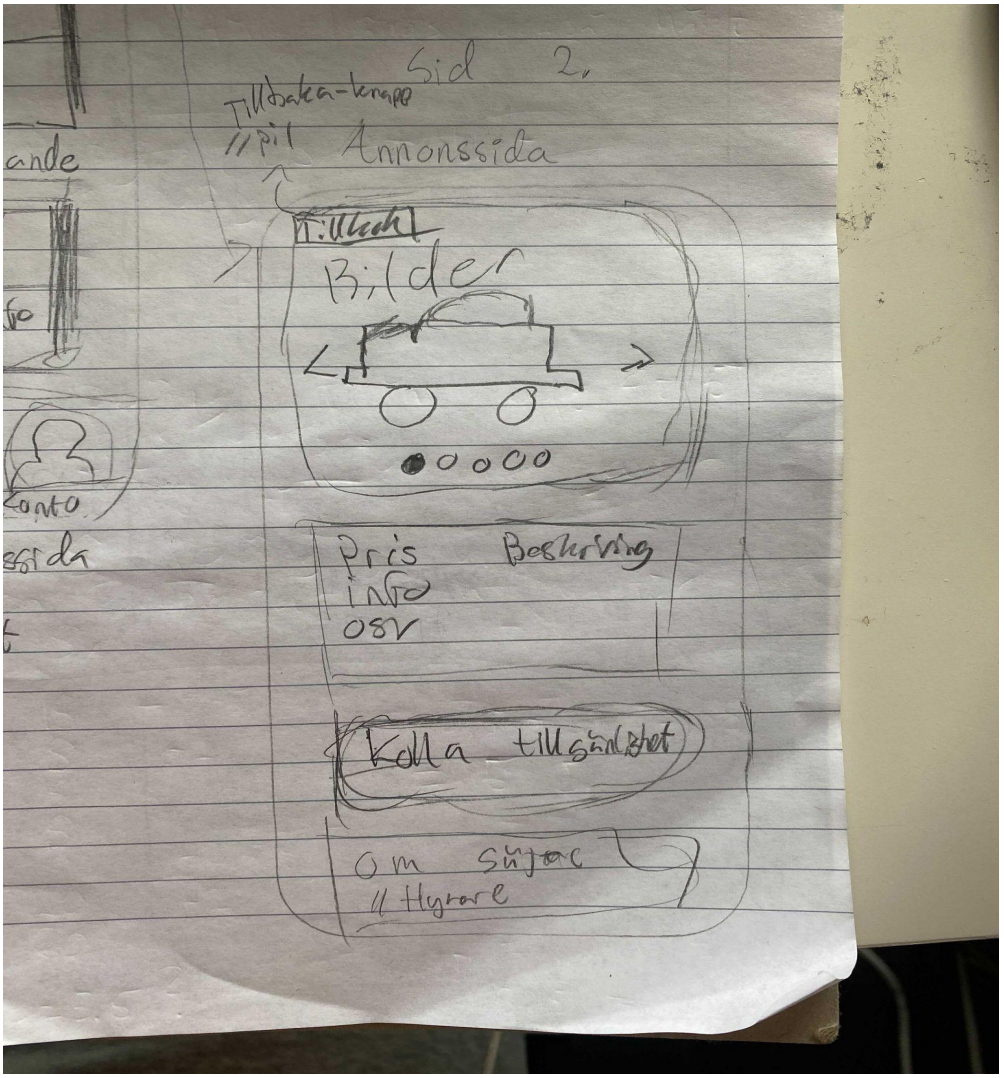

Figure 1: A Preliminary sketch of the interface

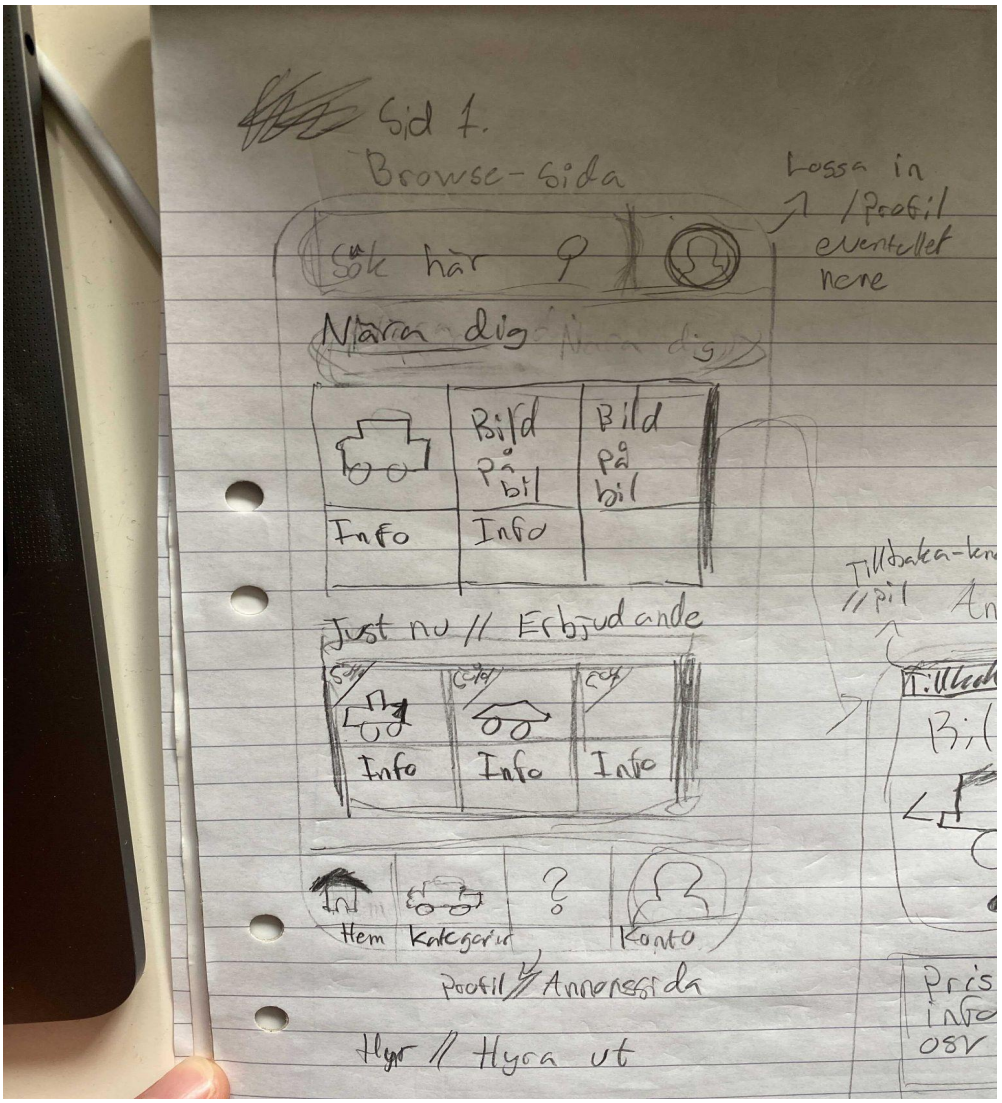

Figure 2: A Preliminary sketch of the interface

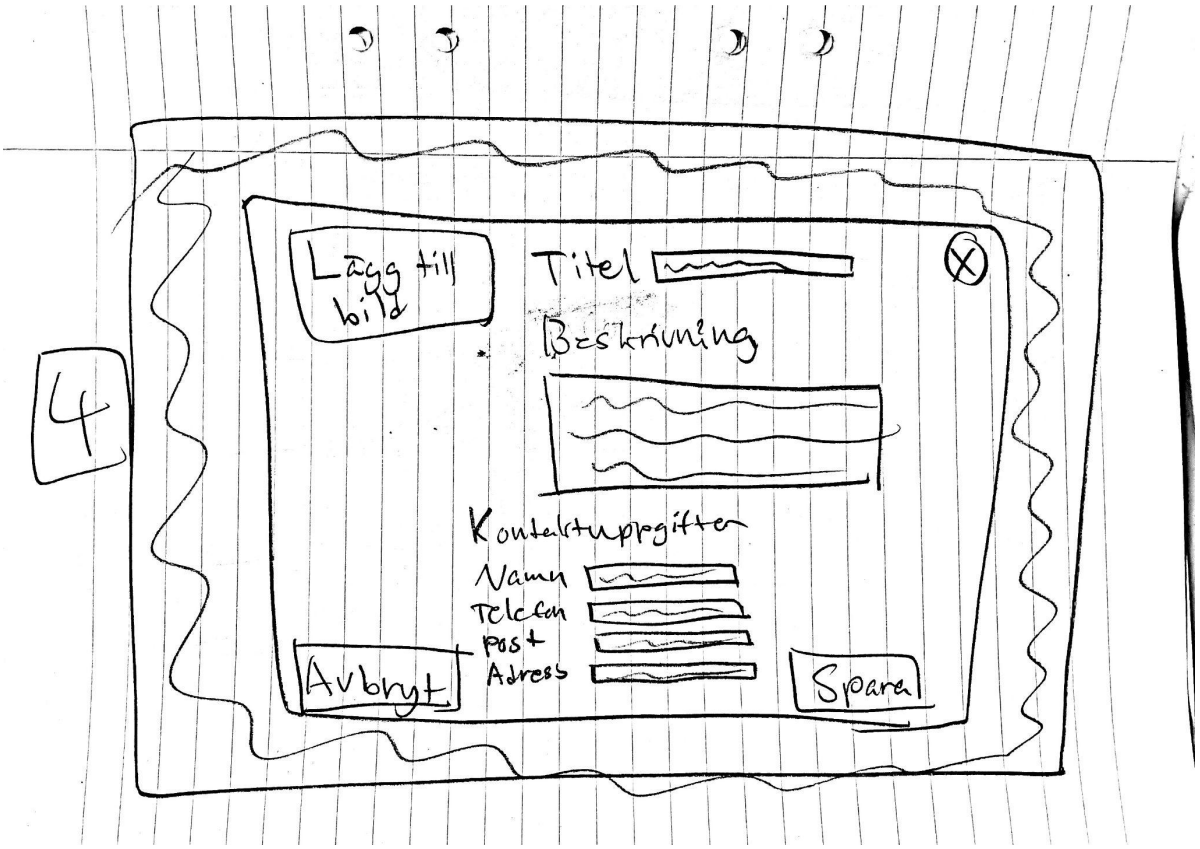

Figure 3: A Preliminary sketch of the interface

Later the interface was made more defined through the creation of a “High fidelity” (detailed) mock-up sketch, created in ”Figma” (See figure 4,5,6).

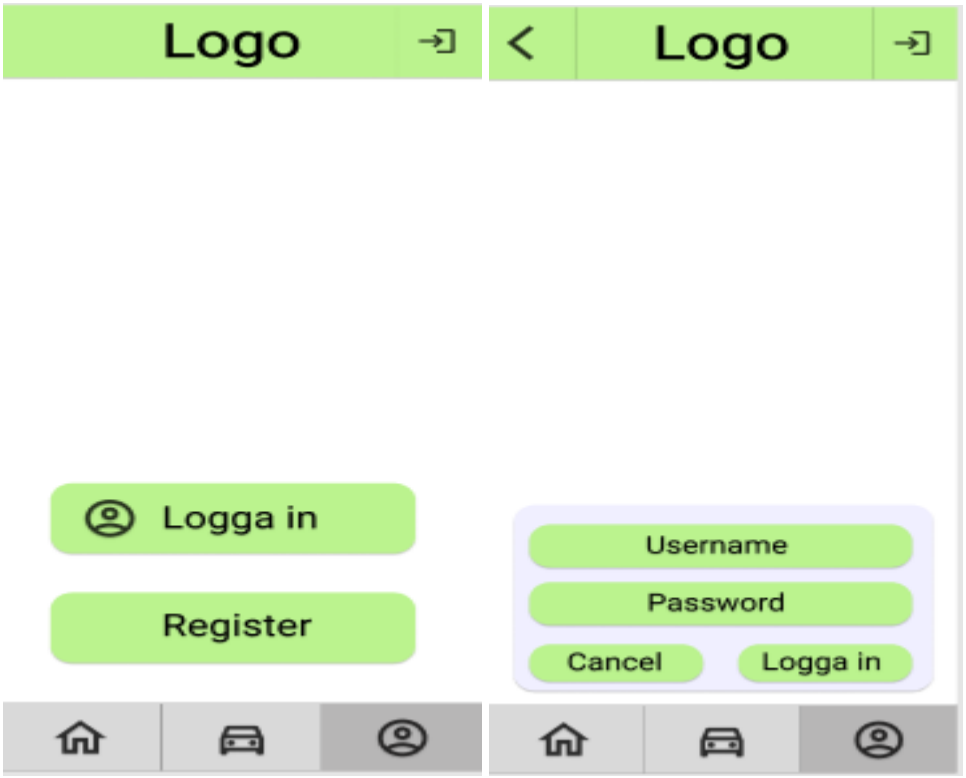

Figure 4: A Hi-Fi mockup of the interface

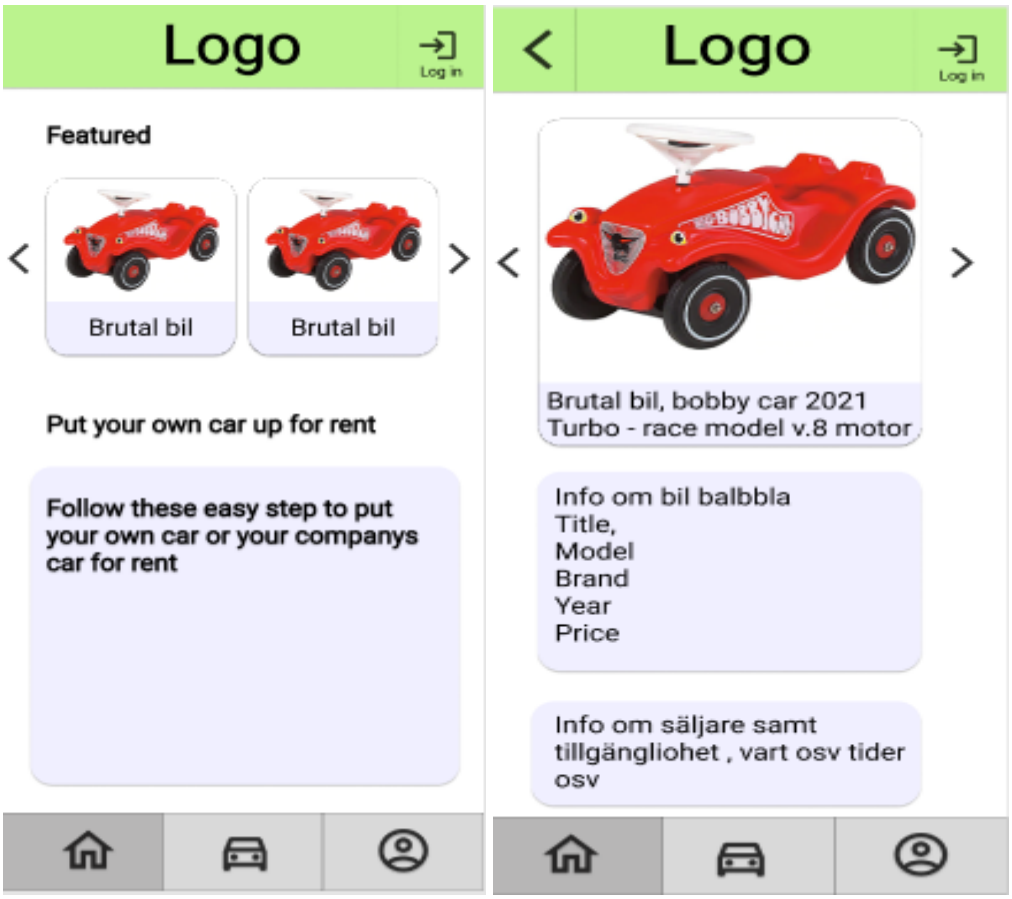

Figure 5: A Hi-Fi mockup of the interface

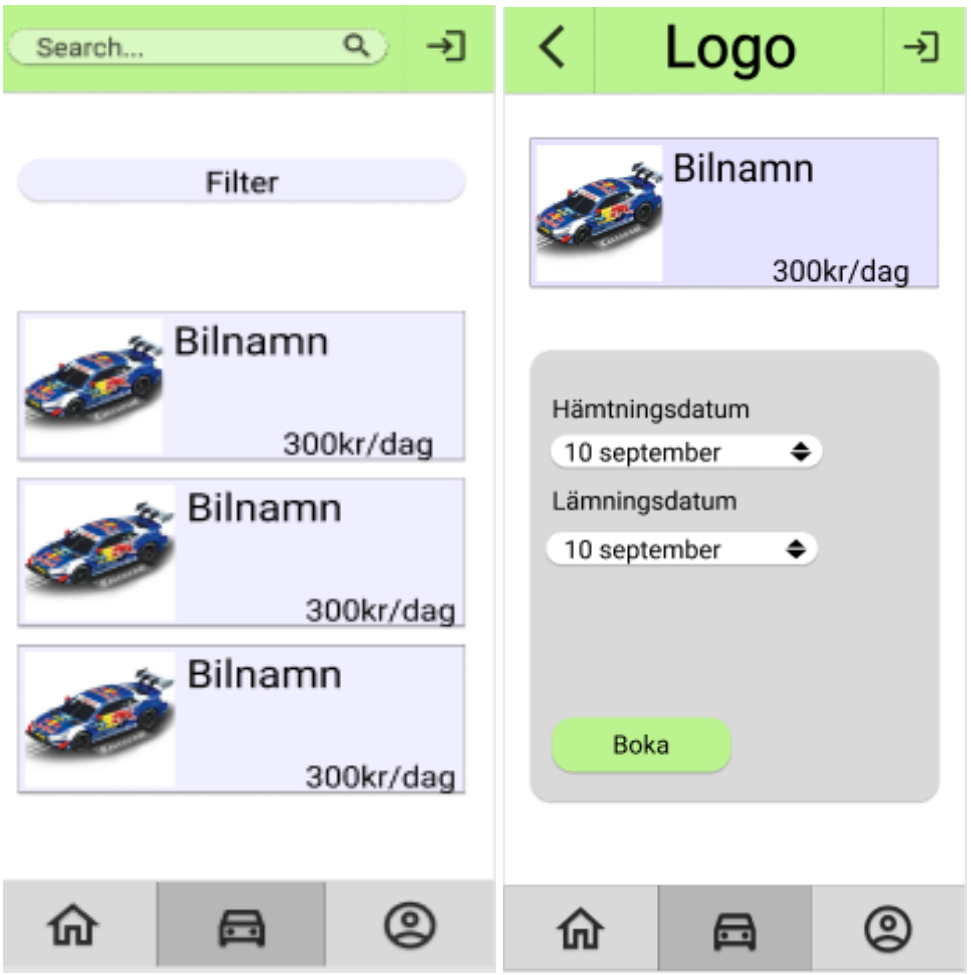

Figure 4: A Hi-Fi mockup of the interface

After getting a comprehensive overview of the application and the most important features that would be included, we started working on the design of the project in .xml files by using Android Studio to get the final product. (See figure 7,8).

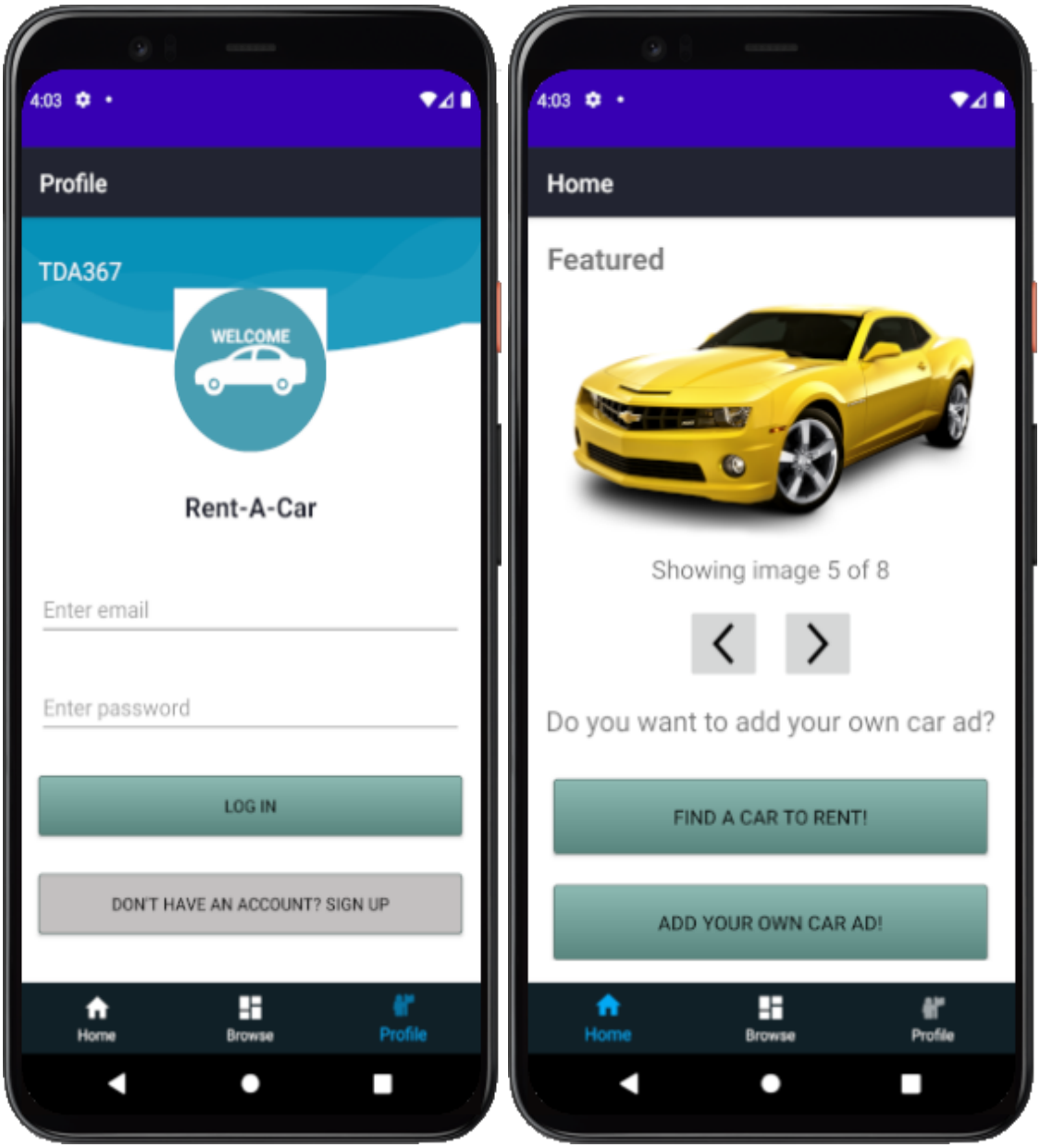

Figure 7: Profile and Home interfaces

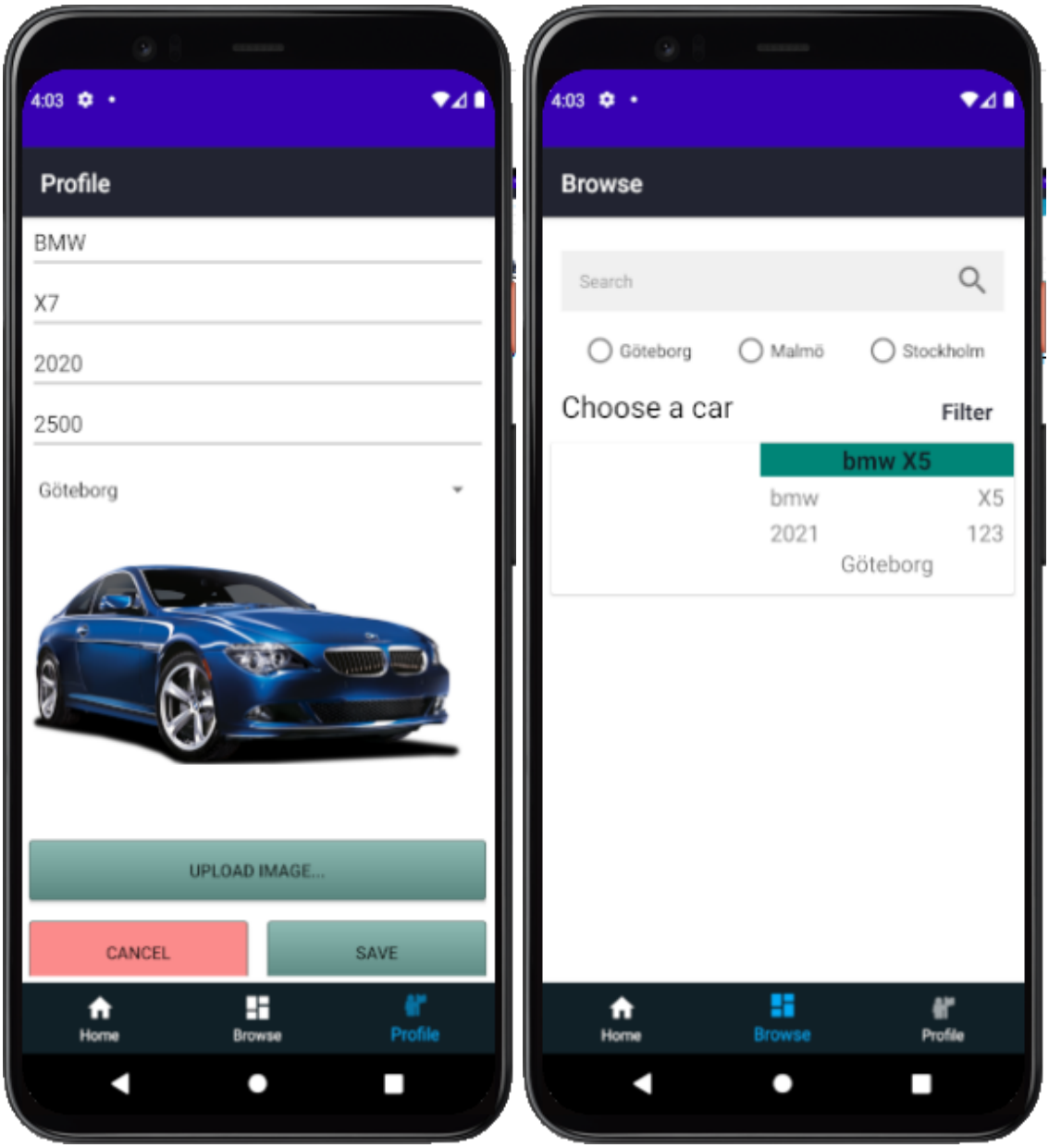

Figure 8: Add Car (Profile) and Browse interfaces

3. Domain model

The program is a type of car rental app that uses user input and input from various APIs to produce informative results which is then fed back to the user through a GUI. The user can also browse the cars without having to enter anything into the program, but he cannot rent a car without logging in or creating an account.

The user is the main information object of the program and the user is the owner of a number of properties. The properties in the app hold most of the information necessary in order to rent a car.

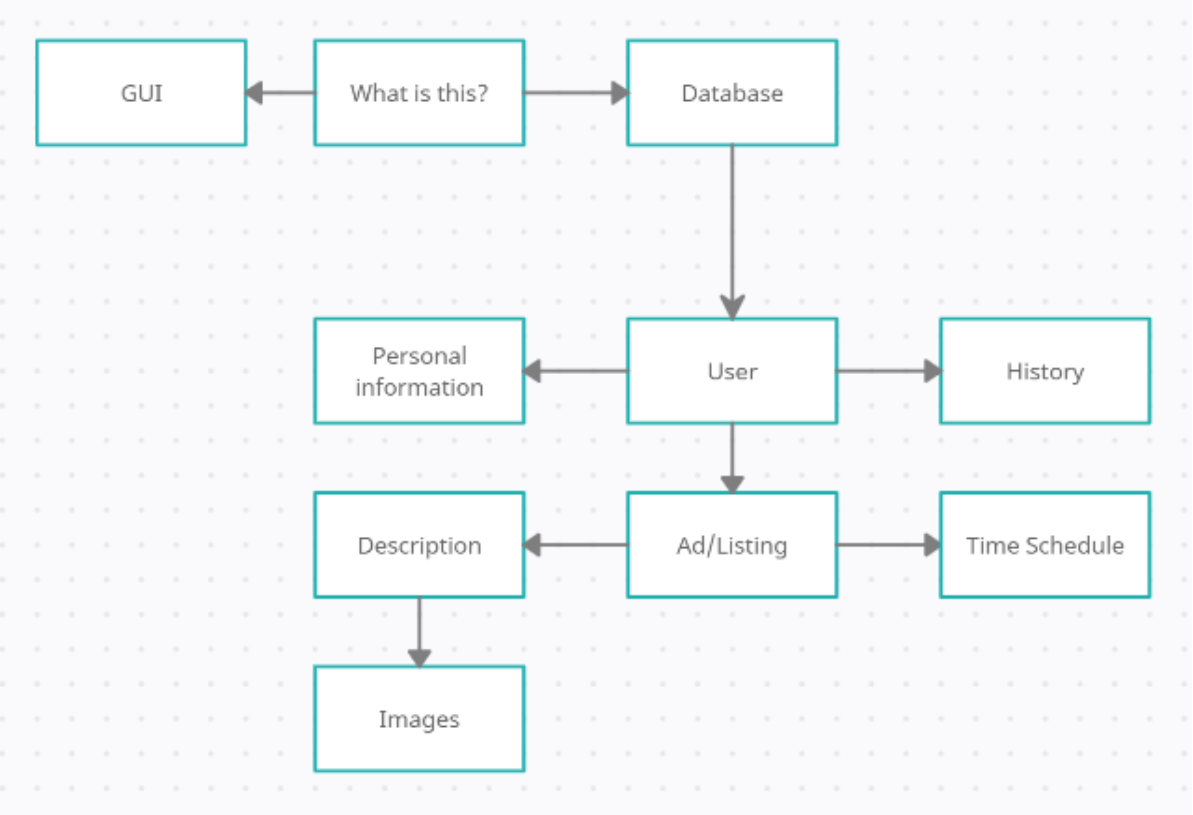

Figure: The Domain of the app

3.1 Class responsibilities

The program uses the MVC design pattern. The GUI part of our model makes up the View. It consists of our MainActivity and Fragments that are part of the Android API.

The Controller part is made up of ViewModels that are also a part of the Android API. This class is used to get information from the Model to our View.

The Model contains our data and logic for the program.

The User class is used to keep information about users such as name, email, card information, etc.

The Listing class is used to keep information about every individual listing in our program. It stores information such as what car is up for rent, price, what dates the car is available, etc.

References

[1] Figma, “Mockup tool,” 2019. [Online]. Tillgänglig: https://www.figma.com

[2] Firebase

[3] Android Studio

[4] Dependency analysis matrix java.
https://www.jetbrains.com/help/idea/dsm-analysis.html

[5] Android Studio emulator https://developer.android.com/studio/run/emulator

[6] Find bugs http://findbugs.sourceforge.net

[7] pmd https://pmd.github.io

[8]Junit https://junit.org/junit5/