Edinburgh Napier UNIVERSITY

Coursework Report

Cristian Anastasiu
40280345@napier.ac.uk
Edinburgh Napier University - Mobile Apps Development (SET08114)

Abstract

For this coursework we were told to do a mobile application at choice. The idea of the developed application was found online, at it was one for a car-sharing app. An app that allows users to register, log-in and add routes that they will do, from a city to another, and, if the had one or more spare seats, other users could join.

Keywords – carshare, amigo, carpool

1 Introduction

The application was given an identity, and called "Amigo" with a little alpha sign in a green circle as it's logo.

Figure 1: **Logo**

2 Software Design

2.1 Classes

As approach on the application was to make it as minimal and easy to use as possible, it only has two classes, RideDetails and UserDetails. They both push and pull data from the database with a Firebase implemented class, FirebaseDatabase. The same goes for the authenticator, which uses FirebaseAuth. Because of the little use of classes, they only have a few methods: getters, setters and constructors.

2.2 Database

The database design is made of two JSON childs, rides and users. Each item stored in users stores the information of a user and has the user id as it's name. Each item stored in rides is pushed in via a FirebaseDatabase method, push(), in order to make every ride added unique, and it then stores every added ride's details. The write permission is granted to everyone, in order to allow every user that registers to store it's details.

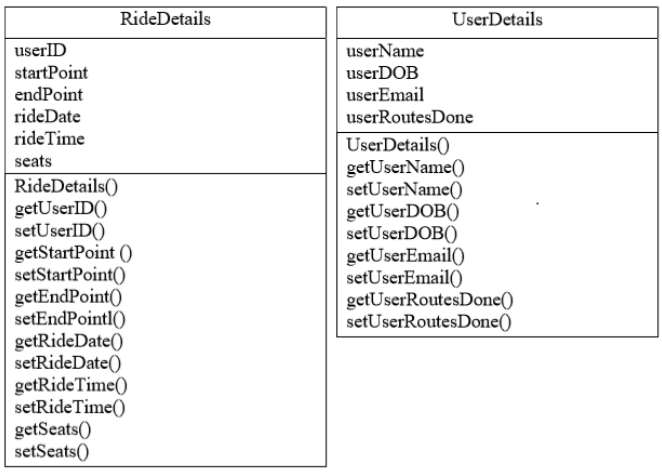

Figure 2: **Class Diagrams** - RideDetails and UserDetails Class Diagram

Listing 1: JSON Database Permissions

```
1  {
2      "rules": {
3          ".read": true,
4          ".write": true
5      }
6  }
7
```

3 Implementation

There have been loads of features implemented but the key ones that make the application easy to use are the bottom navigation bar, the "stay loged in" feature and the card view.

The feature that probably helps the user the most is the "Keep me loged in" feature. It is automatic, so you won't have to toggle it and is made possible with the FirebaseAuth. FirebaseAuth caches a login and the simplest way to implement the feature is to check if there is any user cached in.

The bottom navigation bar helps the user swap between actions without changing it, through fragments. It gives the application a modern feel and simplifies adding and viewing rides.

The main feature of the app is the ListView on the "List" fragment. It's basically the app itself. This is where you can see the routes you and other users add. Each item in it is

a custom routelayout.xml, that creates each card in the list view(now chaning it's name in a "card view").

Listing 2: Keep me loged in feature

```
1   {
2       firebaseAuth = FirebaseAuth.getInstance();
3       firebaseUser = firebaseAuth.getCurrentUser();
4       //those two are set up in a setup() method, not in the ↩
        onCreate() method
5
6       if(firebaseUser != null){
7           finish();
8           startActivity(new Intent(LoginActivity.this, ↩
        MainActivity.class));
9       }
10  }
11
```

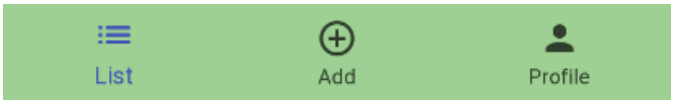

Figure 3: **Navigation Bar** - Allows users to change between the three options

Figure 4: **Card View Window** - Screenshot from the window that displays the routes

4 Critical Evaluation

While looking up for ideas I stumbled upon this particular one that intrigued me to develop, as I know a lot of people that rely on hitchhiking back home. It didn't have any description pointing out it's features, so I started creating an idea of how it would be used.

4.1 Comparison to Original Concept

As the original concept was just a rather vague term, the idea was to start doodling small interfaces to see what users might want from it. Authenticating came up, so that was added as the first action. It then led to either registering or loging in. As the application is on it's early stages, the data stored is minimal, so registering only takes a user's name, email, date of birth and password. After loging in, you would load up the routes, and be linked to a bottom navigation bar that leads to either adding a route or viewing your profile. While sketching up the idea, I was informed that such an application already exists(BlaBlaCar).

4.2 Comparison to Other Applications

As other applications had whole teams of developers and designers working for it, it lacks a lot of features compared to them. Such features include Facebook registering/loging or location picking, smooth interface with beautiful designs and so on.

4.3 Possible Improvements

Right now, the application is at an early stage. All the base functions of it are added, though extras could easily improve it's usability. As you could have possibly seen, the register doesn't require any register confirmation. That could easily help with spam accounts and overpopulating the database with useless items. Another one could be adding user profiles in both the profile fragment and the card shown in the card view. The ImageView has been added, though it has no use right now. So does the small "Routes done" feature. It is there, it's just not working. As observed, the card view only displays the routes, and doesn't allow users to join them. That has to be implemented for the app to have an actual use. A search bar could come in handy for the users to find routes that start from their town/city, ignoring the ones that are miles away.

5 Personal Evaluation

While creating the application I faced a handful of challenges. Some of the things I learned are: using Fragments, Card Views, understanding and implementing JSON, adding both a bottom navigation bar and a top menu bar and the list goes on. I found out that Google has provided great facilities for both JSON and authentication on it's Firebase. It had great documentation about everything and was just a pleasure to use. The most difficult thing to implement must have been the card view. I looked up loads of tutorials online on how to do it, but Firebase saved the day once more with it's Adapter. After completing this task I feel a lot more confident on developing mobile applications, and I am satisfied with the level Amigo as achieved.