MotoTrack

Final Report – Proposed Mobile App

Paul Cvasa

Year 2, Computer Science in English

Programming for mobile devices

16.05.2021

Supervisor: PhD. Lecturer Liviu Octavian Mafteiu-Scai

1. Abstract

For this course, I have developed a mobile application, using Android studio, to fulfil the problem of managing personal motorcycles. In this report I will write in-detail information about my app, different goals and target audience, reviews of similar apps and the development plan.

2. Purpose

The purpose of my app is to help users keep track of their motorcycles and their corresponding mileage.

3. Users

The target audience will mostly be motorcycle owners, aged 18-100(Chester Zawalik at 100 years old – oldest motorcycle rider) and with any profession, which are looking for an easy way to remember their motorcycle's info. Also useful for keeping track of the number of kilometers ridden on the odometer.

4. Introduction

In the last years, more and more smartMotoTrack is an Android application, coded to help users maintain their personal motorcycles. This app also helps remember the current running state of a vehicle.

The visual part of the app will have two main colours: black and white. It will also use some light green accents in some parts.

The user should be able to navigate between these screens: start screen (application logo, name of the app and the login form), the dashscreen(from where you can logout, access the virtual garage or add a new motorcycle) and the main screen which is the virtual garage with all the owned vehicles. Another screen will be the about page with information about the app.

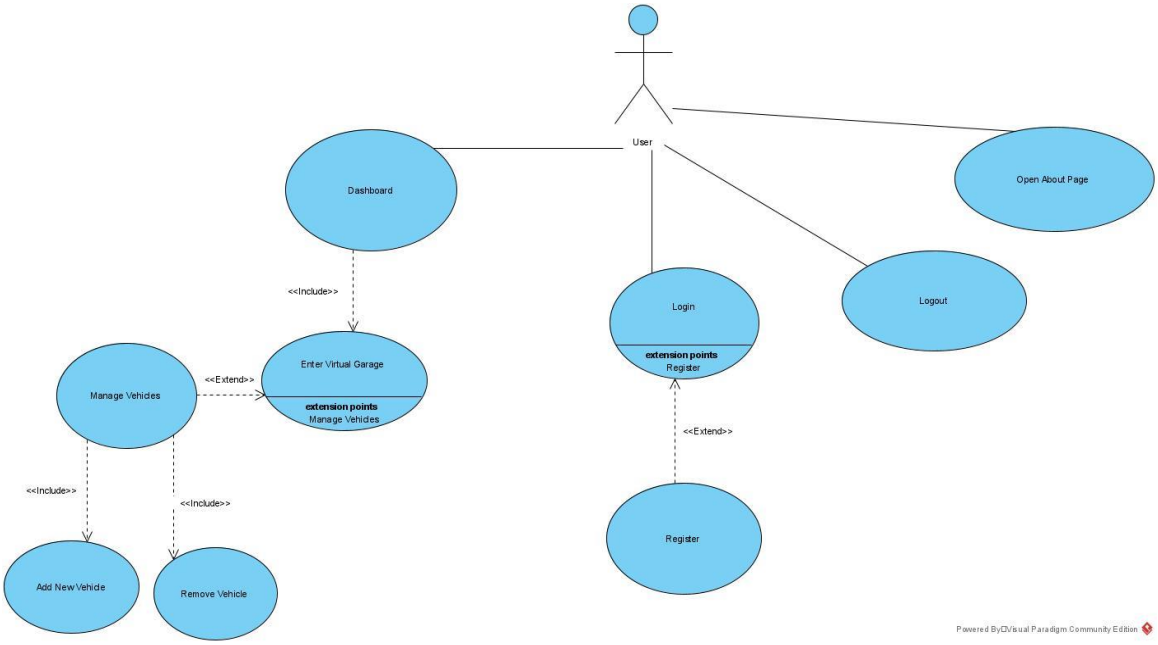
There are a couple of apps on the market that manage vehicles, like My Car – Vehicle Manager but it's not motorcycle exclusive.

5. What is the original contribution of the author?

The originality of my app from my point of view consists of the following:

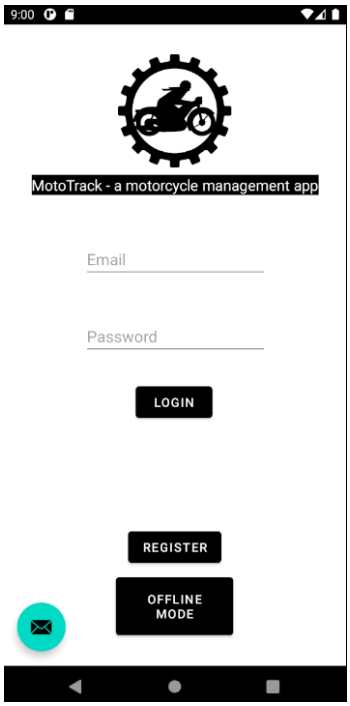
- The motorcycle – type of vehicle exclusive design
- The simple and easy to use interface
- The Offline Mode button for quick use of the application's functionality
- The secure login system using Firebase's Authentication

6. Functionality



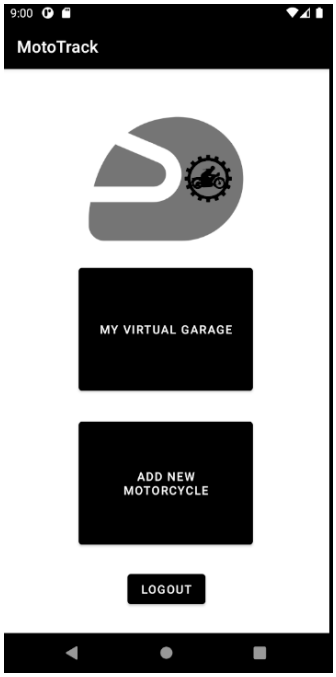
7. Running the app (User's Manual)

After starting the app by pressing the app's icon, it opens the login screen.

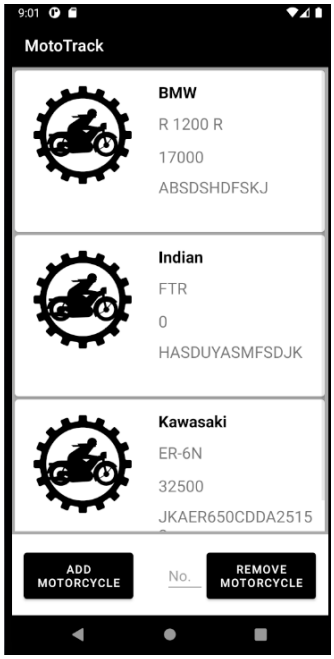


Here you can enter your account credentials and press Login to open the Dashboard, press Register to open the create account screen, press Offline to use the app without an account or press the light green mail button to display the support email.

Here you can create a new account, using your name, email and password. After completing the required field, press Register button to create the account. If you already have an account, press the Login button to return at login screen. Here you also have the light green button for displaying the support email.



On the Dashboard screen, we have three buttons, the Virtual Garage button, which will open the list of motorcycles added, the Add New Motorcycle button which will open the Add Motorcycle Screen and a Logout button to exit the current account and return to Login screen.



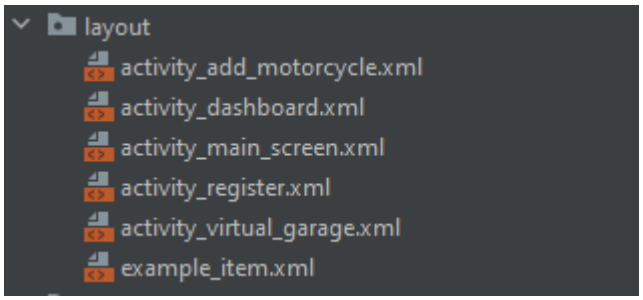
In the Virtual Garage Screen, you will have all the motorcycles that you added, to view them, just scroll up and down to see the complete list. At the bottom, you have two buttons: first one is Add Motorcycle, which will open up the Add Motorcycle screen and for the second one, you have to type a number starting from 0, which will represent the motorcycle which you want to remove. After typing the desired number, press Remove Motorcycle to delete it.

On the Add Motorcycle screen, you will be prompted to type four specifications about the motorcycle you wish to add. First is the Brand of the motorcycle, second is the Model of it, third is the current Mileage and the last one is the VIN (Vehicle Identification Number). After completing the fields, press the Add To Virtual Garage button to save the motorcycle in the virtual garage.

8. App's Structure (Technical Manual)

I used Android Studio paired with the Kotlin programming language to create this app. For my authentication system and the motorcycle database I used Google Firebase.

For the layout, I created six activities:



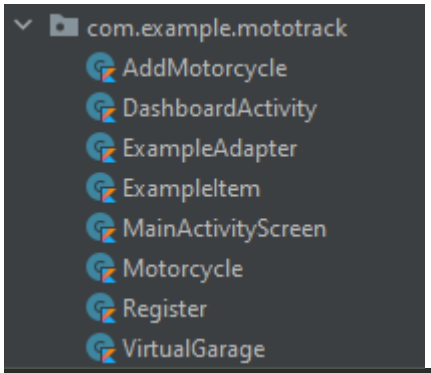
Each activity has a screen, except exampleitem.xml, which is responsible for the design of the motorcycle item in the Virtual Garage.

```
<Button
    android:id="@+id/button_insert"
    android:layout_width="136dp"
    android:layout_height="69dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginBottom="14dp"
    android:onClick="addMotorcycle"
    android:text="@string/add_motorcycle"
    android:textColor="#FFFFFF" />
```

Here is the code for the Add Motorcycle button, which calls the addMotorcycle function below.

```
package com.example.mototrack

import ...

class AddMotorcycle : AppCompatActivity() {

    private lateinit var binding : ActivityAddMotorcycleBinding
    private lateinit var database : DatabaseReference
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityAddMotorcycleBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.addMotoBtn.setOnClickListener{ it: View!
            val motoBrand = binding.motoBrand.text.toString()
            val motoModel = binding.motoModel.text.toString()
            val mileage = binding.mileage.text.toString()
            val vin = binding.vin.text.toString()

            database = FirebaseDatabase.getInstance().getReference( path: "mototrack")
            val Motorcycle = Motorcycle(motoBrand, motoModel, mileage, vin)
            database.child(vin).setValue(Motorcycle).addOnSuccessListener { it: Void!
                binding.motoBrand.text.clear()
                binding.motoModel.text.clear()
                binding.mileage.text.clear()
                binding.vin.text.clear()

                Toast.makeText( context: this, text: "Succesfully Added", Toast.LENGTH_SHORT).show()
            }.addOnFailureListener{ it: Exception
                Toast.makeText( context: this, text: "Failed To Add", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Each layout has a dependent Kotlin file responsible for the functionality of that layout screen.



The VirtualGarage Kotlin file is responsible for the functionality of the list of motorcycles. And also, here I codded the Firebase functionality to retrieve the motorcycles from the online database.

This is it:

```
val rootRef = FirebaseDatabase.getInstance().reference
val mototrackRef = rootRef.child( pathString: "mototrack")
mototrackRef.get().addOnCompleteListener { task ->
    if (task.isSuccessful) {
        for (ds in task.result?.children!!) {
            val motoBrand = ds.child( path: "motoBrand").getValue(String::class.java)
            Log.d( tag: "TAG", motoBrand!!)
            //item.setText1(motoBrand)
            val motoModel = ds.child( path: "motoModel").getValue(String::class.java)
            Log.d( tag: "TAG", motoModel!!)
            //item.setText2(motoModel)
            val mileage = ds.child( path: "mileage").getValue(String::class.java)
            Log.d( tag: "TAG", mileage!!)
            //item.setText3(mileage)
            val vin = ds.child( path: "vin").getValue(String::class.java)
            Log.d( tag: "TAG", vin!!)
            //item.setText4(vin)
            val item = ExampleItem(motoBrand,motoModel,mileage,vin)
            mMotorcycleList.add(item)
        }
    } else {
        Log.d( tag: "TAG", task.exception?.message!!) // potential errors!
    }
    mAdapter.notifyDataSetChanged()
}
```

This took a bit too much time for what it does, but that's how we learn...

9. Conclusions and Future Work

Future features that I would like to add:

- Add a map with all the users that wish to find a gathering nearby
- Friend request system
- Ability to create a motorcycle gathering
- Ability to send messages between the users
- Add all the motorcycle brands and models in a drop-down search list in the Add Motorcycle screen

This project helped me to learn many new things about mobile app development, I am proud of what I have created, even though, to be fair, if I started earlier to work on it, I would've had more time to implement other features, but I will do it anyway in the future because I enjoyed to work on this project.

10. References

www.apps.apple.com/ro/app/my-car-vehicle-manager/id1165749302

www.visual-paradigm.com

www.developer.android.com/studio

www.firebase.google.com/

www.kotlinlang.org

www.developer.android.com/guide/topics/ui/layout/recyclerview