

Technical Project Report - Android Module

# Rent@Bike

Subject: Computação Móvel

Date: Aveiro, 2021-01-27

Authors: 80214: Duarte Dias  
84760: Sérgio Gasalho

Project abstract: This document aims to explain our idea of application (Rent@Bike), as well as all the implementation and setbacks found in its implementation. Rent@Bike is an application that lets you rent a bike from the system or even allows you to rent your own bike. We will start by addressing the concept of our idea in more detail. Next, we will discuss the implementation of the solution as well as its interactions and limitations. Finally, we will talk about the conclusions drawn from this work.

## Table of contents:

### [1 Application concept](#)

[What is Rent@Bike?](#)

[App's Purpose](#)

### [2 Implemented solution](#)

[Architecture overview](#)

[Implemented interactions](#)

[Project Limitations](#)

### [3 Conclusions and Resources](#)

[Lessons learned](#)

[Key project resources](#)

[Reference materials](#)

# 1 Application concept

## What is Rent@Bike?

Nowadays, with all the environmental problems we face, we must try to opt for sustainable mobility alternatives. The bicycle is a great ally because it doesn't pollute, doesn't need fuel and promotes a healthy lifestyle. However, it's always necessary to have an initial investment and when it is bought it's not always possible to have a bike with us all the time. What if we could change that? The goal of our solution is that everyone can enjoy the benefits of a bicycle without having to travel with their bikes or even to purchase one. It doesn't matter if you are looking for a bike to go to work, to explore a new city, take a road ride or even to go mountain biking, our solution gives you that choice! Our app allows you to do this in a very simple way, for that you just have to choose a bike suitable for your use, choose the duration and finally pay for the rental duration. Done!

## App's Purpose

Rent@Bike's main objective is to provide the user with an intuitive app, that allows him to choose a bike with the desired specifications, choose the duration for the rental according to what the user is looking for and finally pay for that rental. Our application provides:

- **Add a bike to the system:** The person who wants to rent his bike will have a page that allows him to add the desired bike, thus inserting its characteristics, photo, price and dates on which it can be rented.
- **List:** A list of all the bikes available for rent nearby.
- **Map:** A map of the user location, with all the bikes that are available for rent nearby.
- **Fast Access:** You can quickly access a specific bike using the bike's qrcode (using camera)
- **Fast Sharing:** You can quickly share the bike's QRcode generated on the app.
- **User Location:** You can easily access your location to see which bikes are nearest.
- **User History:** You can easily access your rental history, by accessing the profile page.

## 2 Implemented solution

### Architecture overview

Our application consists of a main Activity (MainActivity) that contains a Bottom Navigation Bar that allows you to navigate between the three main Fragments. The first is for the map (MapFragment), the second is for the bicycle list (ListFragment) and the third is for the page for adding a bicycle to the system (AddFragment).

- The map page contains two FloatingActionButton, one to move the map to the current location and another that redirects to the qrcode scanner page. The page contains the bicycles and you can be redirected to each one by clicking on each marker.
- The list page consists of a RecyclerView that shows all the bikes the system has available. Associated with each bicycle is a class (Bike) that defines the bicycle.
- The last fragment contains a RecyclerView that shows, in each item, its characteristics in order to be filled and then transformed into a class (Bike).

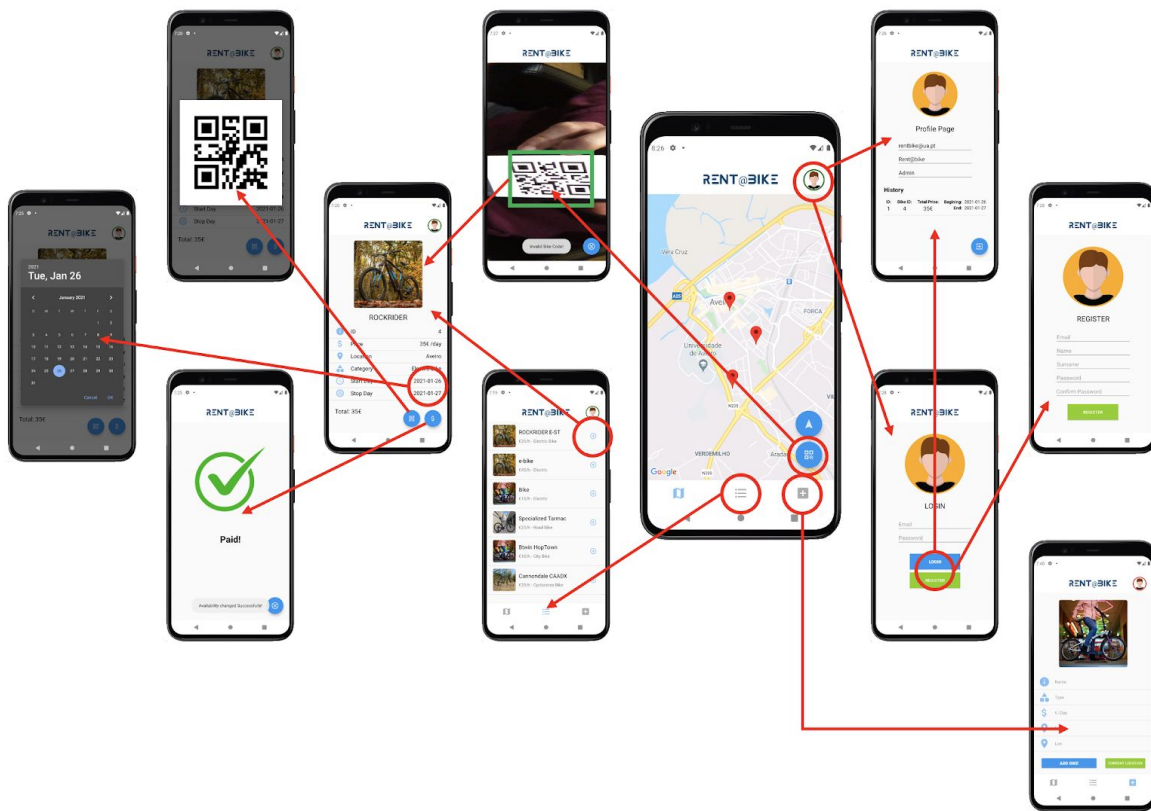
On the top bar, we have an icon (profile) that allows you to redirect the user to their profile area. The profile area is divided into three Activities, the login (LoginActivity), the registration (RegisterActivity) and the profile itself (ProfileActivity). These pages follow the iteration described in the figure below.

We have the qrcode scanner page (ScanActivity) that looks for qr codes in the image captured by the camera and if there is a match, it redirects to the bike page, otherwise a message appears saying that the code is not valid. This Activity implements the BarcodeReader.BarcodeReaderListener interface.

Next we have the page of a given bicycle (BikeActivity) that also contains a RecyclerView that shows each of the bicycle's characteristics (id, price per day, category, etc.) and that also allows you to select the rental dates. To select the dates, a DialogFragment page is displayed (DatePickerDialogClass), where a calendar appears to choose the days. The BikeActivity page also contains two FloatingActionButton that redirect either to the qrcode page (a qrcode is generated for that bike) that is of the AlertDialog type or to the bike rental completion page (BuyActivity).

Regarding firebase, it's used to register and safely authenticate users in our system, to store profile information, to store the history of the bikes rented and to store information regarding bikes available in our system. Regarding bikes, we use Cloud Firestore Databases, that allows us to fetch all the bikes, change its availability and add new bikes. Regarding users, we also use Cloud Firestore Databases, that allows us to fetch the user information of the current user authenticated via Firebase Authentication. Regarding history, we also use Cloud Firestore Databases, that allows us to fetch all the history of a user, and add to the history a bike from the bikes database.

## Implemented interactions



## Project Limitations

Our project has some limitations, these are as follows:

- The bicycles available in our system do not have dates related to their availability, as long as they are available, you can choose any range of days.
- It is not possible to change profile data (photo, name, surname).
- It is not possible to associate a photo when inserting a bicycle in the system, there is a standard photo.
- On the QRcode scanner page, to leave the page without detection, it is necessary to use the "quit" button instead of the "back" button, otherwise the app crashes.
- There is still no page for the invoice (data + qrcode) related to a rental, but that's a future improvement.
- In the future, the app may have access to the location of the bikes so that the owner can always know where his bikes are.

## 3 Conclusions and Resources

### Lessons learned

One of the hard problems that we have is leading with the asynchronous calls in Java, in Flutter it is much easier to lead with asynchronous calls, but after realizing how it is done the first time in Java, it is easier to replicate in other situations.

Another problem we had was related to the permissions to have access to both memory, location and the camera.

For future developments, we can implement firebase storage, to save images of the bikes, and load these images for the application, we know how to do it, but because of time limitations we couldn't do it.

We also can prove that the android native app has a better performance than the flutter application.

### Key project resources

- Code repository: [https://github.com/psagasalho/Rent\\_at\\_bike\\_Android](https://github.com/psagasalho/Rent_at_bike_Android)
- Ready-to-deploy APK:  
[https://drive.google.com/file/d/1FYRdHjCsZ3rk\\_kdwQrYtKR1z0LQfZAsn/view?fbclid=IwAR12S4Rqq0dCKpnQftmrPOc2JygTuIMsL-MH-WNEw4eOPZXyyVVbCupyWsE](https://drive.google.com/file/d/1FYRdHjCsZ3rk_kdwQrYtKR1z0LQfZAsn/view?fbclid=IwAR12S4Rqq0dCKpnQftmrPOc2JygTuIMsL-MH-WNEw4eOPZXyyVVbCupyWsE)

### Reference materials

*Links:*

<https://medium.com>

<https://www.tutorialspoint.com/index.htm>

<https://firebase.google.com/docs/android/setup>