Technical Project Report - Android Module

BikeTrack

Subject:	Computação Móvel
Date:	Aveiro, 27-01-2021
Authors:	72385: Miguel dos Santos Lopes 80323: Lucas Seabra
Project abstract:	Track user precise location when travelling with a bicycle. Keep information of your previously made tracks, take pictures, and share it with other users.

Table of contents:

1 Application concept

The bike market is constantly increasing and people from all around the world are buying a bike to be able to move around without the constraints of a motorized vehicle. This increasing demand is a consequence of either people wanting to be more environment friendly or due to the outbreak we are now facing.

With this in mind, this application was developed in order to satisfy the needs of the more demanding cyclists that want to keep track of their routes with precision.

This application aims to keep track of all the routes of a cyclist. With this app, the users will be able to keep track of their previous routes, saving data such as travelled distance, average speed and time taken to complete the track.

The app will also be able to generate a QR code in order to share a track of your liking with your friends and they will be able try the same tracks you did previously.

The app also includes a camera so that the user can take a picture of any place he likes during his ride.

Therefore, the main focus of the application is to record a track of the user, the distance he travelled and share tracks with their friends and also show the pictures taken during those rides.

2 Implemented solution

Architecture overview

The main component of the project was to use the user location, display it, and save it for future consulting.

For the implementation of the application, we made use of various Android Libraries available to use. Those libraries allowed the use of the most important aspects of the project, that are Google Maps, Room Database, NavigationUI, Location API, Directions API, Camera and Firebase.

In order to extract precise locations from the users, Google Maps, combined with Location API, was used to display the exact position from the user on the map. The Location API, also offers simple to use methods to extract user speed and traveled distance, which were also used to save that information to a specific data model.

To save routes from the user, Room Database was used and it saves the data model “TrackInfo”, in which information from the track is stored, such as distance, time, average speed and all the location points necessary to draw the user route.

Room Database was implemented only for persisting the data relative to the tracks users previously made. To implement it, we used all of the Android Architecture Guidelines, from the Room Database to the Android ModelView. By using these components, synchronization of the data was simple to maintain, because, every time an update existed in the database, those changes were reflected to the ViewModel LiveData and displayed instantly.

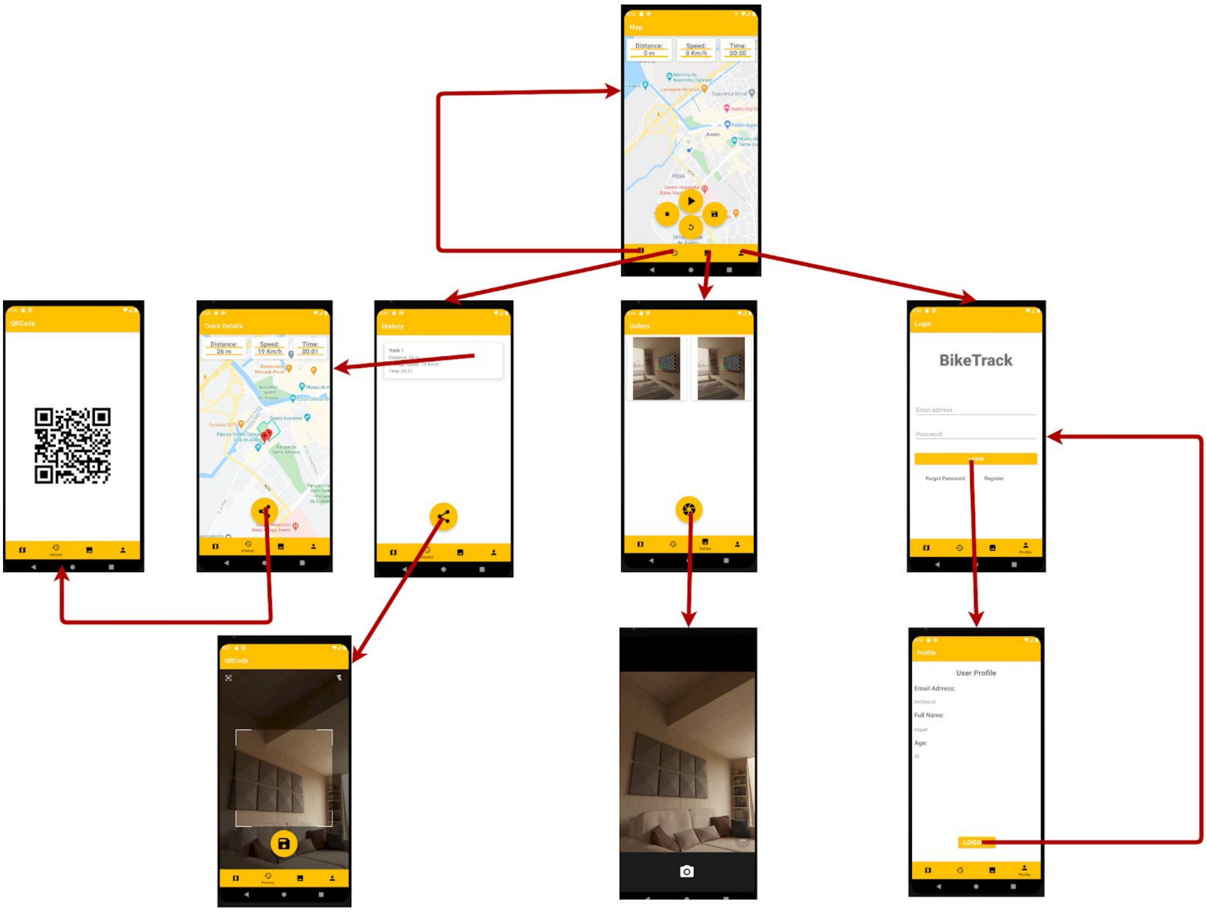
Users can also take pictures with the camera, but, for the pictures, no specific database was used to save them. Instead, the device internal storage was used to keep the images, in a folder that is only accessible by the application itself, these images are then displayed on a custom gallery, specifically tailored to display only images that were taken on the application.

The images, just like the track's information, need to be displayed instantly, after being taken. For this, a synchronization strategy was also needed.

In this case, Android ViewModel was also used, but without the access to a database, just accessing the internal device storage, and the ViewModel was used to monitor that storage for when changes occurred and display them instantly to the user.

For the Firebase interaction, a specific data model was also defined, the User model, that keeps the information of a specific user stored, such as name, age and e-mail.

Implemented interactions



Project Limitations

The main objective of the project was achieved but there were some extras that we aimed to implement, but due to, mainly, time constraints, that implementation was not possible.

One of the main extras was the possibility to follow a path/route, shared by another user. There could be an easy implementation to solve this problem in which we could only highlight the start position, end position, and the route travelled, but we wanted to give meaningful feedback to the user, for instance, if they were keeping the original route, if they were keeping the same average speed as the original user and even if they did track with a better time. This idea was designed to give more purpose to the sharing feature and to provide some kind of competition between users. Also, the share feature is only available through the reading of a QR Code, and we wanted to extend this to share function to support sending tracks via web.

The track history feature is also missing an extra we wanted to have, which is, make use the Google Places API to display a picture of the location of the track, to provide users some visual feedback for the corresponding location.

Since we also have the support for a user to sign in with Firebase authentication, we wished to make the image saving feature cloud based instead of just locally.

3 Conclusions and Resources

Lessons learned

Along the development of the application, the problems encountered were various, some easier to address, others not so much.

For example, the integration of a Room Database using Android Components Guidelines, at first, can be overwhelming, but after understanding the core concepts behind it, it becomes very intuitive, and a powerful tool to interact with a Database, with minimal SQL code writing.

Some other, less troublesome problems were, in using the NavigationUI to instantiate the BottomNavigationBar, that ultimately caused some navigation issues along the development of the project, until we could fully understand how to use that resource properly.

The use of the NavigationUI also caused some more problems, namely on the camera, since we use the intent for capturing an image. This intent was called from a fragment and even overriding the the intent callback “onActivityResult”, NavigationUI uses nested fragments to instantiate a new fragment, which caused problems getting the corresponding image from the camera, because this callback goes for the parent Activity of the Fragment, and not directly for the Fragment making the access to that data troublesome, leading to save the images on local files a reading those files, instead of reading the bitmap directly, with LiveData on a ViewModel.

Apart from these specific problems, the rest of the development went smoothly with some other minor problems that always happen when developing any application.

One component that we found easy to use was the Google Maps API and Location API in order to provide live tracking, comparing the same implementation on the Flutter module.

Key project resources

- Code repository:
    - https://github.com/mlopes95/CMHomeworks/tree/main/Project
- Ready-to-deploy APK:
    - https://github.com/mlopes95/CMHomeworks/tree/main/Project

Members Contribution

- **Miguel Lopes - 90%:**
    - Real-Time Tracking with Google Maps and Location API
    - Room Database usage to save/delete track information (Using Android Components Architecture)
    - Internal Storage usage to save images
    - Custom Gallery
    - Generating QR Codes to share track information
    - Using QR Codes to import track information
    - Report
- **Lucas Seabra - 10%:**
    - Firebase authentication for user login

Reference materials

https://developer.android.com/codelabs/android-training-create-recycler-view#4

https://developer.android.com/training/camera

https://developer.android.com/guide/navigation

https://github.com/androidmads/QRGenerator

https://github.com/yuriy-budiyev/code-scanner

https://developer.android.com/reference/androidx/core/content/FileProvider

https://developers.google.com/maps/documentation/android-sdk/map-with-marker

https://github.com/jd-alexander/google-directions-android

https://www.youtube.com/watch?v=fZIZ81fn7b4 https://developer.android.com/codelabs/advanced-android-training-device-location#0

https://developer.android.com/codelabs/advanced-android-training-places-api#0

https://codelabs.developers.google.com/codelabs/advanced-android-training-google-maps#0

https://developer.android.com/training/data-storage/room

https://developer.android.com/codelabs/android-training-cards-and-colors#0

https://android.jlelse.eu/launch-screen-in-android-the-right-way-aca7e8c31f52