

Technical Project Report - Android Module

GhostRunner

Subject: Computação Móvel

Date: Aveiro, 03/06/2020

Authors: 85070: Daniel Marques
84690: Miguel Dinis

Project abstract: GhostRunner is a mobile app similar to a “run tracker” but enhanced with better qualities and features, in order to record tracks/trails on the map that are accessible for the other friend users. Keep track of each trail performed and its information.

Table of contents:

[1 Application concept](#)

[2 Implemented solution](#)

[Architecture overview](#)

[Implemented interactions](#)

[Project Limitations](#)

[New features & changes after the project presentation](#)

[3 Conclusions and Resources](#)

[Lessons learned](#)

[Key project resources](#)

[Reference materials](#)

1 Application concept

GhostRunner is a mobile app similar to a “run tracker” but enhanced with better qualities and features, in order to record tracks/trails on the map that are accessible for the other users.

Not only that but the Home Page can tell you your daily goals, like calories burned or your trails stats (runned distance, time, altitude and maximum speed).

In this page are saved and shown every trail done, giving you details and the routes about each one of them.

The Profile Page displays the personal information about the runner user (name, weight, height, kcal burn goal and a QR code to be used in the runners network), this information will be useful for calculations through the app. This information can easily be edited just like the profile photo.

Then in the Runners Page can be found every friend added and all fitness information and the trails of each one. This page also contains a QR code scanner that can scan the QR codes of each runner, adding them to the user friend’s network.

GhostRunner's main page is the interactive map that takes you right to your location.

Given the current location and clicking the show trails are displayed several trails (trails from the user and his friends) that can be scrolled and on click in each one the map is centered to the specific trail showing its route the main information about it and the record. Clicking on the start trail button, immediately starts the timer for the run and then can be saved.

At this time the user is running against the record runner for this trail in the form of a ghost, showing his location in realtime on the map like he starts the race at the same time.

Alongside the timer is a speedometer to keep track of the speed. Instead of starting a trail already created the user can create one by clicking on the add button.

When the user stops the timer is taken to a page to create the trail, associating a photo to the trail, a name, description, etc, adding it to the user trail list in the home page.

Features

- Real time position tracking retrieved by GPS.
- Display of current user position in a map, existing saved positions from ghosts (with markers) and trails that can be selected.
- Creation of trails based on current position (Start point) and a final position indicated by the user.
- Timer that can be paused or terminated.
- Speedometer sensor that appears when running a trail.
- QR code generator and scanner (to found user profiles).
- Get images from camera sensor or android gallery.
- Photos from trails are saved on Firebase Storage and always retrieved from there (not local storage).
- Realistic statistics retrieved

- All data created in the app is stored in Firebase firestore and retrieved from there too. (Trails, Users).
- Google authentication for registering unique users.

2 Implemented solution

Architecture overview

The app was divided in 4 main modules which are also the 4 main pages.

Although it doesn't make sense in the context and structure of our app, these modules were made in four distinct fragments, this made the transition between the tabs of the navbar easier, it could be done with activities but we decided to keep the structure of the example of the navbar provided by Android Studio.

The first module is the page that presents the tracking of all the daily values achieved and statistics of the trails made. These values include the total distance covered, the calories burned, the time spent and the average speed of the daily routes and the remaining calories until the goal, and are shown in a RecyclerView with an Adapter that we developed to show the values in the best way, in this case with the background image of the trail and the overlapping main values.

This RecyclerView is also used on the map page.

The second module depicts everything that has to do with the map, where the trails are created and where the values are taken from to be shown later.

Here each user will be able to see the trails he and his friends created, and run them (competing against his ghost friend), or on the other hand he will be able to create a new trail, but regardless of the choice, the timer and speedometer are shown immediately and counting. All trails performed are described on a page to properly identify them and then sent to the database.

In the third module is the page of the friends runners, they will only appear if their QR codes are exchanged, by each part or by both. When you click on a friend, information about his physique and the created trails are displayed.

The last module is the presentation of all the user's profile information such as the profile photo, which can be changed by accessing the camera or gallery, its weight and height and his QR code that is automatically generated based on his ID.

To guarantee the structure and consistency of the data in the application and in the database, two essential models were created:

- **Trail Model:** in this model the trail is characterized by having an id, a name, a location, a brief description, its duration, the distance, the average speed, the date of its creation, the url of a photo taken or chosen from the gallery, the coordinates of the beginning and end of the trail, all the coordinates that make the polylines and a flag indicating if the trail is new (it has a parent trail) or was a “started trail”, this information is necessary to know the times to be compared on the ranking page of a parent trail.

- **User Model:** in this model the user is identified as having an id, a name, his profile url photo, the google account photo is chosen if the user logged in with it and can be changed on profile page, his weight and height, his calorie goal and the list of all his friends.
- **Firebase:** All application data is stored in the *firebase* database including photos of the trails and users (firestorage cloud) in order to be possible a synchronization between different friendly users, in other words, it is possible to be instantly presented with trails created by users and their statistics. One of the advantages of firebase, that proved to be quite useful, is the possibility of creating an object in java (in our case the User model and Trail model) and automatically firebase converts these objects into pairs of keys and values. It should be noted that the google authentication service is also used and that it is strictly necessary in our application as it is the starting point for all interactions between users.

Implemented interactions



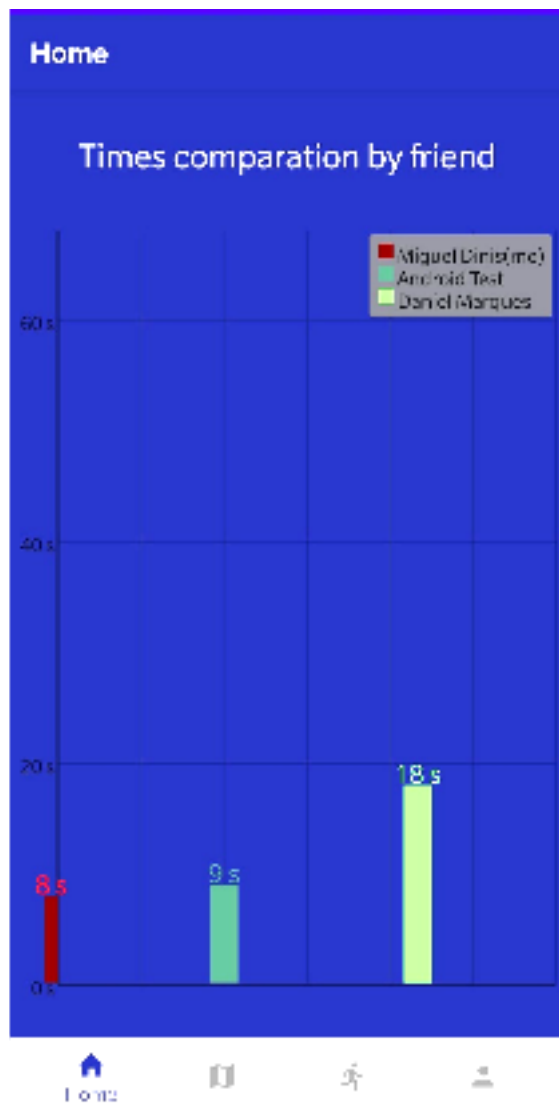
Project Limitations

Although we developed and presented a well-structured application with some features, time was a factor that limited error correction in our application that, eventually, might occur with excessive use and even the development of new features implemented in the previous application developed in Flutter (ex: Text To Speech and more statistics information).

We took into account the professor's proposal at the end of our presentation, in which a ranking system among the several runners was suggested, and we implemented it.

New features & changes after the project presentation

In order to have a way to compare performance between different friend runners on the same trail, we developed one page that can be accessed by clicking on a trail ("My trails" zone in "Home" page). After the click, the user is directed to a page with bar graphs with the time it took him to travel his trail and the time of friends who also traveled it.



3 Conclusions and Resources

Lessons learned

One of the problems encountered was how we were going to implement the ghost and the solution we found was to get the ghost marker to move every second to the next position saved in the database.

The use of a database such as firebase and a cloud service like firestore have proven to be quite useful and versatile, however were also one of the most difficult aspects to implement in the project because there was a need to realize in advance how all the components would relate.

It was a surprise when the several modules developed (Runners and Trails information, Map display, ...) started to come together, once connected to the Firestore database.

This allowed the trails of each one to start appearing in the appropriate places as well as the trails of the friends who belong to the user network.

Mobile computing is being adapted by users at a faster pace than desktop computing and will comprise a bigger factor in the workplace.

Mobile apps have their advantages, such as cost reduction, new revenue channels, brand image strengthening, etc, and this is great because any user can work/use the app from the comfort of any location and time, and this subject allowed us to know the different modules existing in an application in order to develop a well structured application for a given purpose.

Key project resources

- Code repository: <https://github.com/MiguelDinis/CMProject2>
- Ready-to-deploy APK: <https://github.com/MiguelDinis/CMProject2/tree/master/APK>

Reference materials

Our references were basically the Android Code Labs shown in class, firebase official tutorials, StackOverflow and for the chosen theme we had the idea of making a system in which people could compete against a ghost, be it the user himself or a friend, just like in car games.



As inspiration for the development of the UI, the following images were focused on:

