Technical Project Report - Android Module

HealthHelper

Subject:	Computação Móvel
Date:	Aveiro, 10th of June 2020
Authors:	84734: Fábio Daniel Ribeiro Alves 84831: Sérgio de Aguiar
Project abstract:	HealthHelper was created as an application with the intent of allowing people to have somewhat of an idea on how sedentary their lifestyle is. It accomplishes this by tracking the user's movement in real-time to assess if they should be moving more often or not. Additionally, the user can check previous routes taking while traversing the world we live in and even create markers at points of interest.

Table of contents:

1. Application concept

This app's objective is to help the user manage his health by tracking how sedentary he is, his walked distance and previous data. Another objective is saving points of interest as map markers so that the user can revisit them in the future if he so desires.

The application's target users are those who live a more sedentary life, allowing them to track some aspects of their life to motivate them to try to correct themselves.

The application allows its users to immediately know whether they should be more active. This is accomplished using suggestive colors on the main menu's main feature, the sedentary ratio tracker. Additionally, if a user sees the application describe them as having a "Bad" ratio, it will stimulate them to change it.

2. Implemented solution

Architecture overview

Sensors

- The application uses a timer with a timer task that implements a sensor listener. This way the application can listed to the needed sensors (5 samples every second), updating the stored info every second even when navigating to another activity.

Camera

- For the camera, the application makes use of an intent to invoke the existing android camera application, saving the image in the external storage with full resolution.

Graphs

- For the graphic we use a library called GraphView. This library allowed us to quickly generate a graph for the time data collected.

FirebaseAuth

- The application interacts with FirebaseAuth in the Sign in and Register pages to complete their respective objectives. Additionally, there is also interaction in the Main Menu page where the user signs out by returning to previous screens, and in any screen where FirebaseFirestore is used to know the current user's uid.

FirebaseFirestore

- The application interacts with FirebaseFirestore in the Main Menu, Tracking, Favorites and Past Data pages. All four enunciated pages feed off data from it, yet only the Main Menu and Tracking add new values (times in the Main Menu page, marker info and polyline coordinates in the Tracking page).

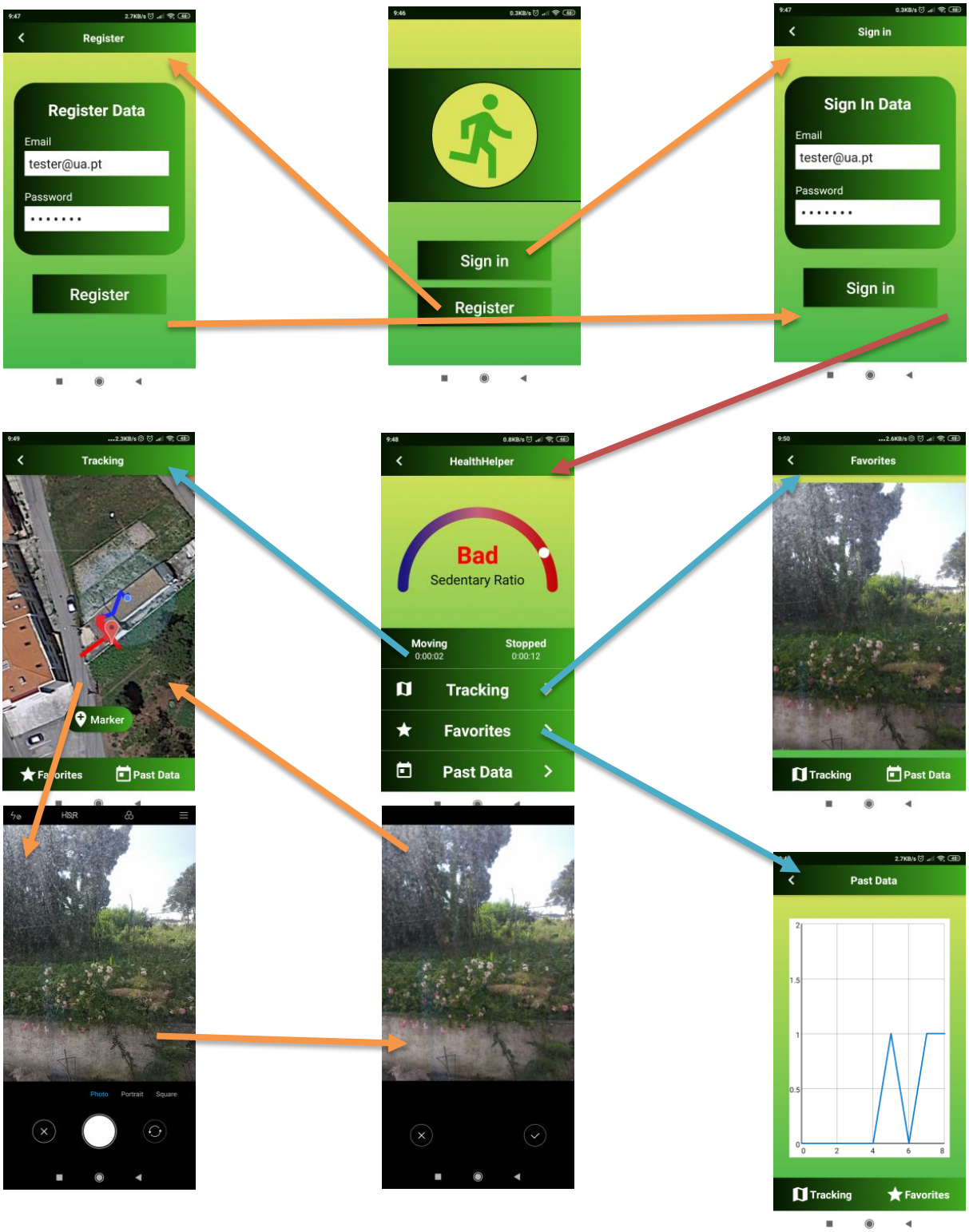
Persistence and Synchronization

Over the course of the application's usage, a variety of information not present in the application is required. This information though, is received through asynchronous calls which need to be handled carefully.

When handling authentication (Sign In and Register with FirebaseAuth), the application makes use of a listener to know when the result has arrived and then treats it based off the result (success or not).

When handling getting data from FirebaseFirestore, the application once again makes use of listeners to update the local variables as the data flows in. As an example, in the Main Menu page, the counter values are displayed as a placeholder '-:--:--' until this operation has finished.

Implemented interactions



Project Limitations

Distinction between types of movement (Pedometer had an anomalous behavior and was not counting steps correctly. Due to this, we could not tell if a user was running/walking or displaying other kinds of movement).

New features & changes after the project presentation

Added the use of both FirebaseAuth and FirebaseFirestore, both of which are essential to the application's usage.

Added a polyline on the map that displays the user's last route (in red, with info gotten from FirebaseFirestore); and one that displays the user's current route (in blue, updated as the user moves).

Added the ability to add markers to the map and associate them to a picture (taken at the moment of marker adding).

Added a graph that displays the user's movement data through time (info gotten from FirebaseFirestore).

Added marker photos being displayed in the favorites page.

Added time moving/stopped counters. This data is also both saved to the FirebaseFirestore and displayed on the screen via a sedentary ratio tracker and two timer displays (one for each counter).

3. Conclusions and Resources

Lessons learned

As for issues faced when working on HealthHelper, we had three: the usage of fragments, updating information as it is gotten from the database and the addition of views programmatically.

As for the usage of fragments, their usage was hindering development as the time taken to learn how to use them was far greater than the time we had to finish the project. Overall, we value code maintainability very highly and if something such as this appears as a solution, we would rather avoid its usage seeing as future maintenance wouldn't be as easy.

As for updating information as it is gotten, we were having some issues when it comes to returning information from FirebaseAuth and FirebaseFirestore auxiliary functions. To combat this, we decided to implement classes to serve as wrappers for certain data structures we would use so we could create our own listeners using interfaces.

As for adding views programmatically, we had issues with the addition of ImageViews onto the HorizontalScrollView present on the favorites page. As it stands, images are being correctly displayed inside the scroll view's child, which is a LinearLayout, however we have no control over where or how the images appear, to the point of the screen starting off as empty when images are loaded. The user has to scroll to be able to actually see the existing images, plus there is an extreme amount of space between each image in the view. We could not fix this, due to being low on time.

Overall, we believe that android development is not as powerful or versatile as Flutter and think that the CM class would benefit from only having a Flutter module in the future, possibly replacing the android module with one that approaches the usage of Dart/Flutter on other platforms.

Key project resources

- https://github.com/sergio-aguiar/HealthHelper-Android
- https://github.com/sergio-aguiar/HealthHelper-Android/tree/master/app

Reference materials

Library: GraphView

- https://developer.android.com/courses/fundamentals-training/toc-v2
- https://developer.android.com/guide

Library: ArcSeekBar

- https://github.com/MarcinMoskala/ArcSeekBar

Library: Firebase (FirebaseAuth and FirebaseFirestore)

- https://firebase.google.com/