UNIVERSITY OF TARTU

Institute of Computer Science

Computer Science Curriculum

Silver Liivamägi

Running tracker – an android collaborative sports application

Bachelor's Thesis (9 ECTS)

Supervisor: Satish Narayana Srirama, PhD

Tartu 2018

Jooksu jälgija – androidi rakendus rühmaspordiks

Lühikokkuvõte:

Bakalaureusetöö kirjeldab Androidi rakenduse loomist, mis on mõeldud kasutamiseks erinevatele jooksmisentusiastide gruppidele. Aplikatsioon sisaldab endas baasfunktsionaalsust, mis katab enamuse jooksjate vajadused. Rakendus pakub kasutajale lihtsat jälgimisseadet, et mõõta oma kiirust, aega ja läbitud distantsi treeningu jooksul. Programm on võimeline salvestama ja hiljem visualiseerima treeningu andmeid, koos ülejäänud treeningute ajalooga.

Lisaks pakub programm jooksuhuvilistele uut võimalust sotsialiseerimiseks ja uute jooksukaaslaste leidmiseks. Selle saavutamiseks on kasutajatele antud võimalus teha avalikke ja privaatseid postitusi selle kohta, kus ja kunas grupijooks võib toimuda.

Võtmesõnad:

Android, GPS, Firebase, jooksmine, andmed

CERCS: P175 (Informaatika, süsteemiteooria)

Running tracker – an android collaborative sports application

Abstract:

This bachelor's thesis details the creation of an Android application which is intended to be used by different groups of running enthusiasts. The application includes basic functionality that covers the needs of most runners. The program also provides a simple tracker for users to keep track of the speed, time and distance they have covered during their exercise. The application can save the exercise data and later visualize it for the user along with rest of the users running history.

Furthermore, the application gives running enthusiasts a new way to socialize and find running partners. This is done by giving the users the ability to make public and private posts about where and when a group run activity is going to take place.

Keywords:

Android, GPS, Firebase, running, data

CERCS: P175 (Informatics, systems theory)

Table of contents

1. Introduction

1.1 The problem

Smartphones and social media have had a profound effect on the way we interact with each other. It has affected the way we meet new people, the way we communicate and the way we find out and go to different events [1].

People have also become more active and share more about themselves. These changes in our society have helped popularize health and sporting applications. The rise in popularity of these applications has also been aided by smartphones becoming more capable of tracking human movement and sporting watches becoming cheaper and more affordable [2].

Sporting applications is category which already has a lot of competition and there are apps that are relatively good and widely used. But the author has yet to find an application that tackles the problem of creating public and private running events. Furthermore, all the other applications are in some way connected to an already existing social media platform like Facebook or Google+. Making it harder to find new connections and friends. This is something the author has set out to change with this application.

1.2 The solution

The aim of this thesis is to create Android application which would allow its users to perform the following tasks:

- The application should make it easier for the user to start with their running or jogging exercises.
- It should track user's movements on a map and also measure the speed, distance and time of the exercise.
- Furthermore, the application should enable the user to easily socialize with other joggers by creating public or private events.
- The application should visualize and present data about the most recent exercises the user has completed.
- Also an option to view all history should be present.

1.3 Outline

Chapter 2 gives an overview of the technologies used in the creation of the application. It also covers the differences between the work created in this thesis and other similar works.

Chapter 3 describes the final application.

Chapter 4 contains the usability test.

Chapter 5 discusses the future development options for the application.

Chapter 6 contains all the references.

Chapter 7 contains the appendix.

2. State of the art

2.1 Introduction

The first part of this chapter goes into detail about all the technologies that have been used in the creation of the thesis. Technologies that have more competition in their field, have an explanation, why the author chose them over others. The second part of this chapter explores work that is similar to the one created in this thesis. It contains a table that compares applications and under it an interpretation of the information in the table.

2.2 Technologies used

2.2.1 Android

Android is a mobile operating system that was originally developed by Android Inc. but is now owned by Google. Patterned after the Linux kernel, the Android operating system was released as open source code. Development for the Android operating system may be done through Windows, Linux or Mac. Although primarily written in Java, there is no Java Development Machine (JDM) in the platform. Instead of allowing Java programs to run through the JDM, Google developed Dalvik, a virtual machine specifically for Android. Dalvik runs recompiled Java code and reads it as Dalvik bytecode. Dalvik was designed to optimize battery power and maintain functionality in an environment with limited memory and CPU power [3].

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast—every day another million users power up their Android devices for the first time and start looking for apps, games, and other digital content [4].
Because of the wide user base of the platform and previous experience in developing smaller applications for Android, the author has chosen it over its competitors.

2.2.2 Android studio

Android Studio is Android's official IDE. It is purpose built for Android to accelerate development and help build the highest-quality apps for every Android device.
Based on Intellij IDEA, Android Studio provides tools custom-tailored for Android developers, including rich code editing, debugging, testing, and profiling tools.

Android Studio also comes with the android emulator. The Android Emulator installs and starts apps faster than a real device and allows prototyping and testing applications on various Android device configurations: phones, tablets, Android Wear, and Android TV devices. The emulator also alows the developer to simulate a variety of hardware features such as GPS location, network latency, motion sensors, and multi-touch input [5].

2.2.3 Firebase

Firebase is a cloud service provider which offers backend as a service (BaaS). Firebase allows developers to quickly sync data and make it available on an application, which is being used by multiple users. Besides offering a real time database, Firebase also provides, among many other services, the ability to create push notifications and user authentication [6].

Firebase Authentication provides a service to authenticate users using passwords, phone numbers or popular identity provides such as Google, Facebook and Twitter. Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with custom backends [7].

Firebase also provides a cloud-hosted database. Data is stored there as JSON and synchronized in real time to every connected client. If the developer wishes to build a cross-platform application, then Firebase can be setup so that all of the clients share one Realtime Database instance and automatically receive updates with the newest data [8].

Firebase Cloud Messaging (FCM) allows software developers to send push notifications to their applications' end users through an application programming interface. With push notifications, the cloud service acts on behalf of the app and only connects to the mobile device when there are new notifications. Using message targeting, FCM is able to deliver messages to applications in three ways: to single devices, to groups of devices, or to devices subscribed to topics [9].

2.2.4 Java

Java is a widely used general-purpose programming language purposely designed with object-orientation in mind. An object in Java can take advantage of being part of a class of objects and inherit code that is common to the class. A Java method can be thought of as one of the object's capabilities or behaviors. Java is the most popular programming language for the Android smartphone applications. Java can be used to create complete applications that may run on a

single computer or be distributed among servers and clients in a network. The source code is compiled into what Java calls bytecode, which can be run anywhere in a network on a server or client that has a Java virtual machine (JVM). The JVM interprets the bytecode into code that will run on computer hardware [10].

Quite a few options exist for choosing a language when developing for Android. Java, Kotlin and Dart are three languages most commonly associated with Android development. For developing the application the author chose Java because it has been around the longest in Android development and because the author was most familiar with this language out of the three mentioned above.

2.2.5 Gradle

Gradle is an open-source build automation system. At the heart of Gradle is a rich extensible Domain Specific Language (DSL) based on Groovy. Gradle pushes declarative builds to the next level by providing declarative language elements that can be assembled as the developer wishes. It provides utmost flexibility to adapt Gradle to every projects unique needs. Gradle scales very well. It significantly increases productivity, from simple single project builds up to huge enterprise multi-project builds. The Gradle Wrapper allows to execute Gradle builds on machines where Gradle is not installed. [11].

2.2.6 Google Maps Android API

The Google Maps Android API gives developers the ability to add maps based on Google Maps data to their applications. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. The API also allows the developers to add markers, polygons and overlays to a basic map, and to change the user's view of a particular map area. These objects provide additional information for map locations, and allow user interaction with the map [12].

2.2.7 Google Location Android API

One of the unique features of mobile applications is location awareness. Mobile users take their devices with them everywhere, and adding location awareness to applications offers users a more contextual experience. The location API comes with the Google Play services. This makes it easy to add location awareness to an application. The API provides automated location tracking, geo-fencing and activity recognition [13].

Google Play services location API allows the developer to request the last known location of the user's device. Usually developers are interested in the user's current location, which is often equivalent to the last known location of the device. [14]

2.2.8 MPAndroidChart

MPAndroidChart is a library for Android that allows developers to easily add different types of charts to their application. The library runs on API level 8 and upwards [15].

2.2.9 Picasso

The Picasso library makes it easy for developers to add images to their Android applications. Many common pitfalls of image loading on Android are handled automatically by Picasso:

- Handling ImageView recycling and download cancelation in an adapter.
- Complex image transformations with minimal memory use.
- Automatic memory and disk caching [16].

2.2.10 CircleImageView

CircleImageView is a library for making circular images for profile images. The library uses a BitmapShader and does not:

- create a copy of the original bitmap.
- use a clipPath.
- set Xfermode to clip the bitmap (which means drawing twice to canvas)

As this is just a custom ImageView and not a custom Drawable or a combination of both, it can be used with all kinds of drawables, i.e. a PicassoDrawable from Picasso) [17].

2.3 Similar work

A brief comparison to three popular sports trackers on the Android play store (Table 1).

Table 1. Comparison with similar applications

Name of the application / Functionality	Running Distance Tracker +	Sportactive	Sports Tracker	Application created in this thesis
Tracker that shows time, distance and average speed	yes	yes	yes	yes
Tracker displays users movements on a map	yes	yes	yes	yes
A feature to plan future runs	Paid extra	no	yes	no
Ability to view run history	yes	yes	yes	yes
Ability to edit users profile	yes	yes	yes	yes
Ability to search for people	no	no	yes	yes
Individual work out plan	no	yes	no	no
Dashboard with history of users health details	no	yes	no	no
Sharing done workouts	no	no	yes	no
Creating public events	no	no	no	yes
Creating private events	no	no	no	yes

Many sports tracking applications already exist and they all manage to provide different approaches and configurations. Some applications focus more on the social aspects of running and others are more oriented to give their users a detailed overview of the user's health and progress. Another factor that separates applications is the requirement to register an account.

The application created in this thesis will focus on trying to give the users a different way to socialize with other runners. The main difference to other applications with social integrations is the ability to create public and private running events. The application should act as standalone social network for runners.

3. Design and architecture of the application

3.1 Introduction

This following chapter is separated into three sections. The first section contains the functional requirements of the application. The second part encompasses every view that is in the application, and goes into detail describing the looks and functionality of each one. The third and final part in this chapter describes the Firebase backend. A detailed overview of each table in the database is given.

3.2 Functional requirements

1. User should have the opportunity to login using a username and password.
2. New users can register to use the application.
3. Users should be able to change the password of their account.
4. The users should be able to track their running distances and the path of the run should be visualized on a map.
5. The users should be able to track the time a run has taken
6. The user should be able to track the average speed of the run.
7. While tracking their run, the user should be able to turn off the phones screen.
8. Users should be able to view their running history
9. Users running history should contain data about distance, time, speed and date.
10. The user can make a post which contains of a title, map, description, date, time and an image.
11. An option to choose who can view the post should be available
12. After registration users can set up their profile
13. Users can click on posts to see detailed information about the event.
14. Users can edit their profiles, username, profile picture and password.
15. Only the users who created a post can delete it.
16. Users can search for other users
17. Users should have the ability to send friend requests to other users.
18. Users should have the ability to accept or decline friend requests
19. Users should receive notifications when somebody adds them as a friend.
20. The user should have the ability to view their friends list.

21. Application must give information about errors, successes or confirmations in readable form. There must also be info logs.

3.3 Application's frontend

The application consists of the following views:

1. Welcome view
2. Login view
3. Registration view
4. Account setup view
5. Main menu view
6. GPS tracking view
7. Plan an event view
8. View planned events view
    a) Public events view
    b) Friends events view
9. Profile view
10. History view
11. Friends view
12. Friends list view
13. Change password view
14. View all users view

3.3.1 Welcome view

Welcome view is the view that is displayed to the user when they open the application and are not logged in. From this view the users can choose to log in to an existing account or create a new one.

3.3.2 Login view

Login view allows the user to log in to an existing account using an email and password (Figure 2). If the email and password do not match, the user will be notified of this.

3.3.3 Registration view

Registration view allows the user to register a new account with an email and password (Figure 1). The email has to be a valid email and also the password must be at least 6 characters long.

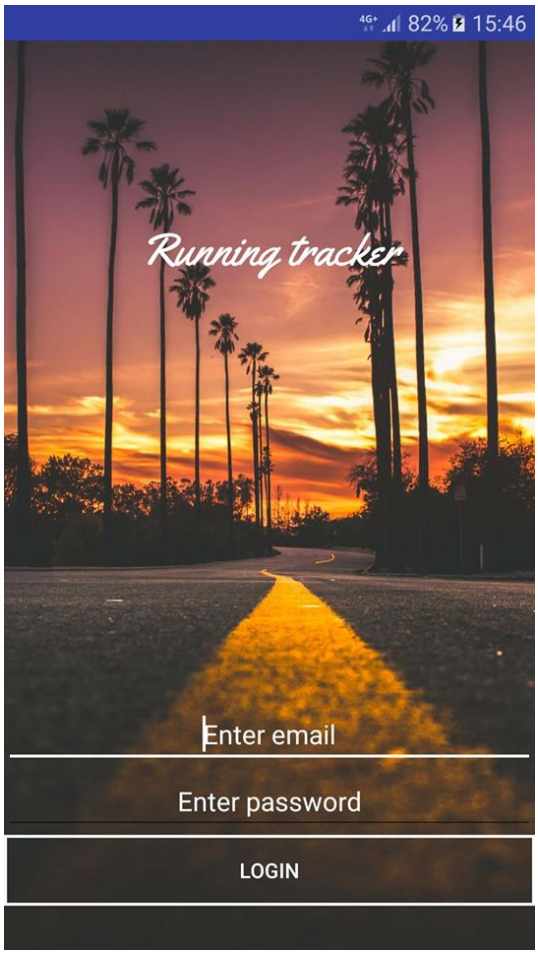

Figure 2. Login view

Figure 1. Registration view

3.3.4 Account setup

Account setup view is a view that the user will be prompted to after finishing the registration. Here the user can choose their profile picture, enter their name and other personal information. Once the user taps the “Done” button their profile will be updated in the database. The user will be forwarded to the main menu view.

3.3.5 Main menu view

Main menu is a view the user will be greeted by every time they log in. The view consists of three buttons labeled (Figure 4):

- Start tracking
- Plan an event
- View planned events

In the toolbar of the view are two more buttons. The button on the right side will open the history view of the application. The button on the left side will open a navigation drawer that contains five buttons labeled (Figure 3):

- My account
- Friends
- Find friends
- Change password
- Logout

This view acts as a central hub for the whole application. From here the user can navigate to every other view of the application.

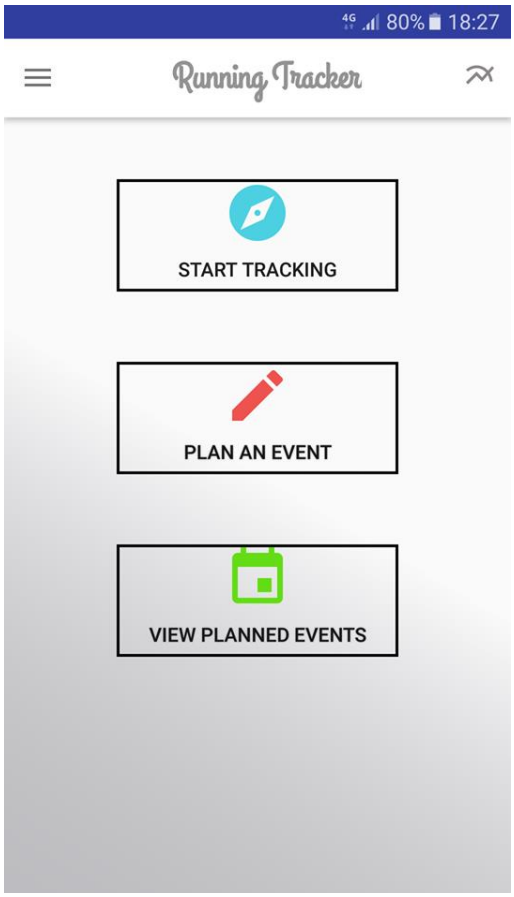

Figure 4. Main menu of the application

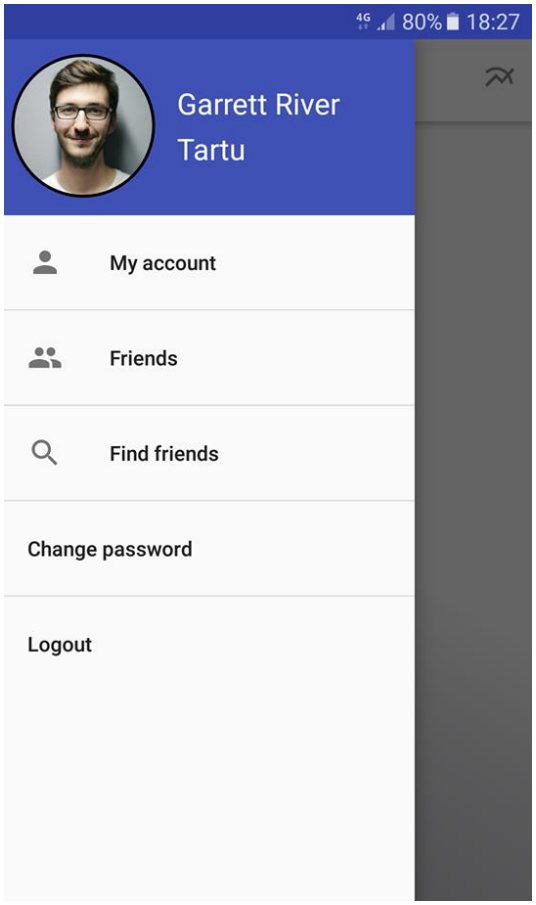

Figure 3. Navigation drawer

3.3.6 GPS tracking view

The GPS tracker consists of three main components, the map fragment, buttons and three different metrics (Figure 5).

The map fragment enables the visualization of the user's location and the path that they have travelled during their trainings.

The view has three text views for displaying time, speed and distance data to the user.

Lastly the view has three buttons for interacting with the tracker.

- The "Start" button starts several components in and outside the view. The three metrics that display speed, time and distance will be reset and started. Also a service that feeds the GPS tracker view with location data about the user will be started.
- The "Stop" button stops the service that feeds the location data, it also resets all the displayed metrics. Next all the relevant training data is collected and sent to the Firebase Realtime Database.
- The "Cancel" button also stops the location service and resets all the metrics but it does not send the gathered data to the database.

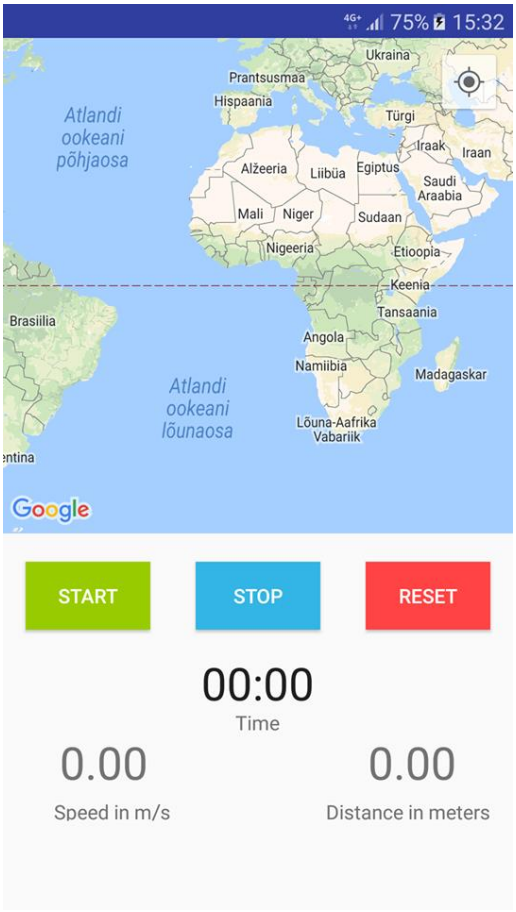

Figure 5. GPS tracker view

3.3.7 Plan an event view

Plan an event view gives the user the ability to make running related events for other users to join. These events can be made either public or private. If the event is made public, everyone who uses the application will be able to see the event. If the event is made private only the friends of the user, who created the event, can see it.

The view allows the user to set various types of information about the intended event. The view starts with a textbox for setting a title for the event. It is followed by a map fragment where the event creator has to set the start and end point of the intended run. The application will then draw the shortest path between these two points to give users who are potentially interested in the event, an idea of the scope of the run. Next the user has the ability to set a description for the event followed by two text views to set the start time and date for the event. At this point the user can choose to make the post that they are creating either public or private by toggling a switch. Lastly the user can choose an image from their device to personalize their event's post (Figure 6).

Figure 6. Event creation view

3.3.8 Public events view

Public events view is stream of public events displayed in the form of cards. These cards are ordered by the time they were created (Figure 8).

Cards consist of a title, image, start time and date, description and the name of the person who started the event. If a user clicks on a card, they will be forwarded to a detailed view of the card. A detailed view will show the user the anticipated route the run will take along with the start time and date, title and description (Figure 7).

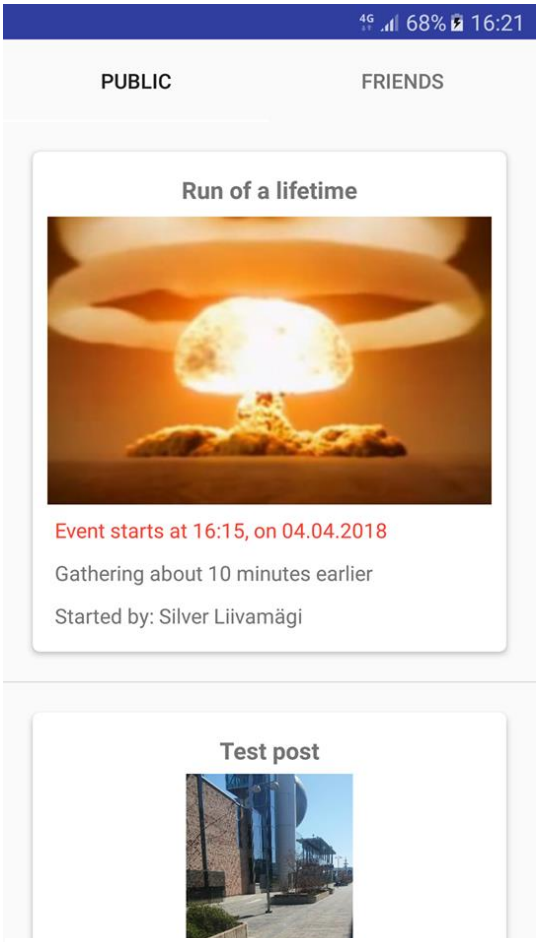

Figure 8. Public posts view

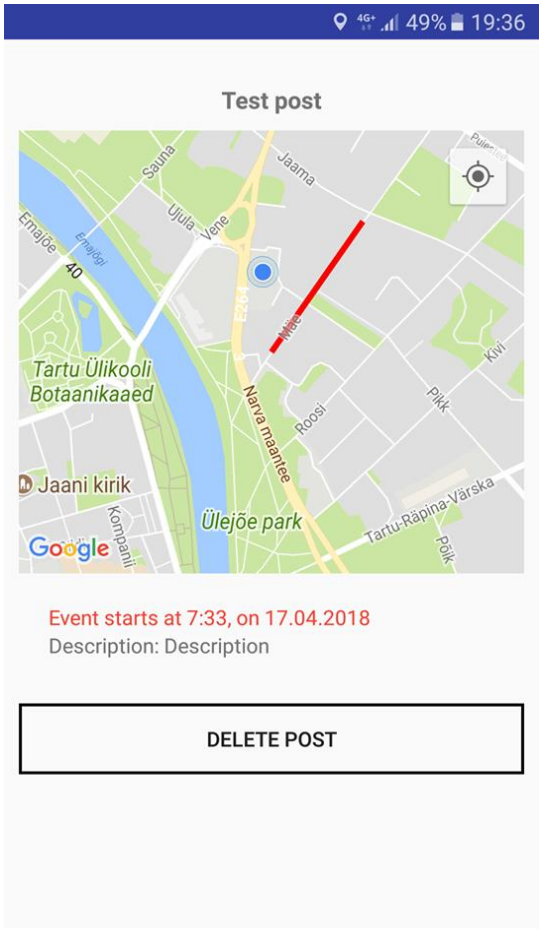

Figure 7. Detailed view of a single event

3.3.9 Friends events view

Friends events view is a stream of private events displayed in the form of cards. These cards can only be viewed if the user is friends with the event's creator. The cards are ordered by the date they were created. The cards behave the same way as the public events cards. Meaning they consist of the same elements and also have a detailed view related to them.

3.3.10 Profile view

Profile view is a view where the user can see and edit their personal information. By tapping on their profile picture the user can select a new image from their device. If the user taps on the background image they can also choose a new image for that, from their device. Users can edit and view their birthdate, location and add a status or introduction of themselves. Any changes to the profile will be registered only if the user clicks the “Done” button. (Figure 9Figure 9).

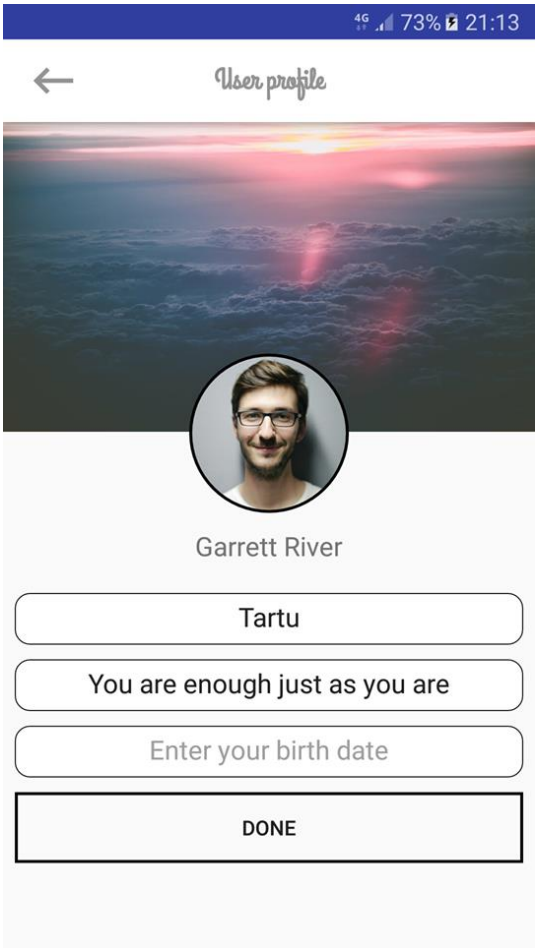

Figure 9. Profile view

3.3.11 History view

History view is composed of two parts: a bar chart displaying either the last 7 or the last 14 trainings and a list view of all the trainings.

The bar chart displays the most recent 7 or 14 trainings, the x-axis shows the number of the run and the y-axis displays the meters the user traveled (Figure 11).

Below the bar chart is a list of the whole history of the user's trainings. A single run is displayed in a card that that shows the date when the run took place, the distance the user ran, their average speed and the time it took (Figure 10).

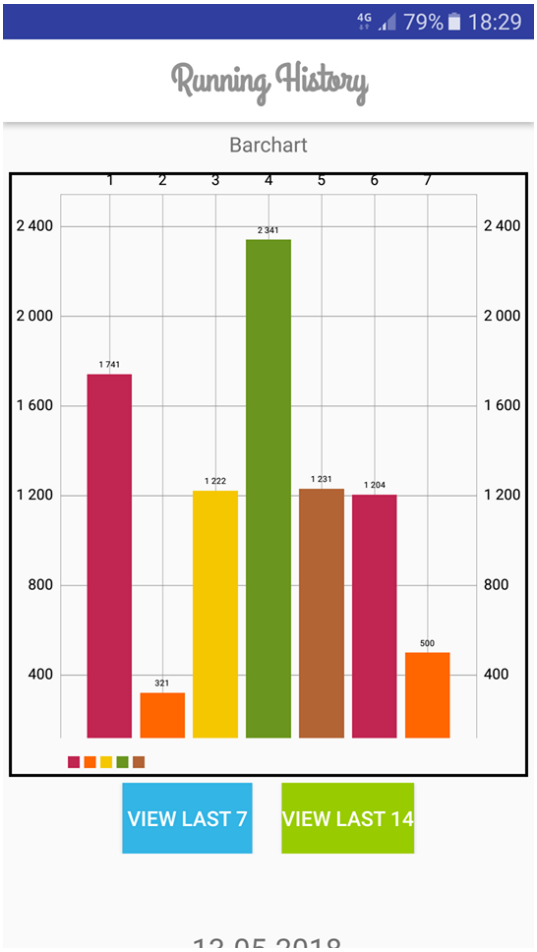

Figure 11. History as a bar chart

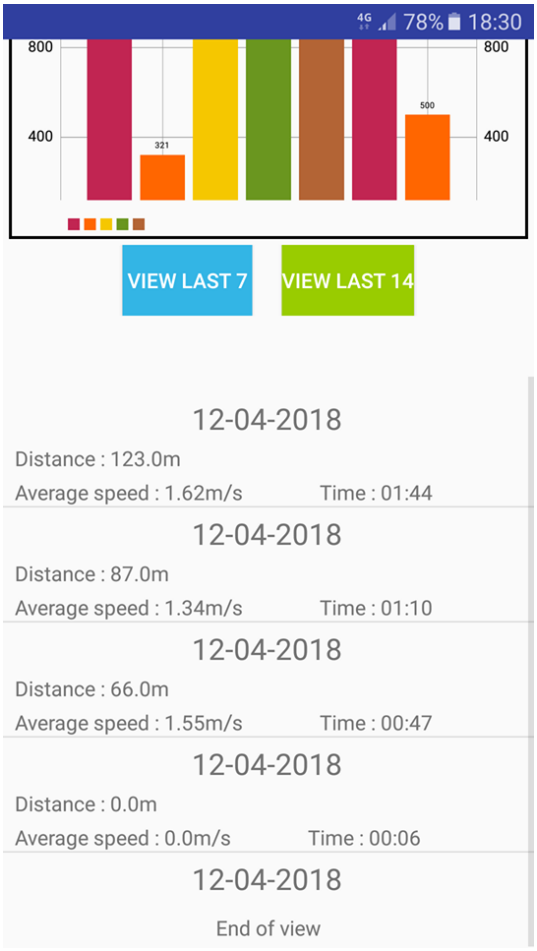

Figure 10. History as a list view

3.3.12 Friends list view

Friends list view shows the user all the other people they are friends with. The information is presented to the user in a list of cards. Each card displays basic information about their friend. If the user clicks on the friend's card, they will be forwarded to their friend's profile view (Figure 13).

3.3.13 Friends request view

Friends request view displays to the user all the friend requests that have been sent to the them. The information is presented to the user in a list of cards. Each card displays basic information

about the person (Figure 12). If the user clicks on the card they will be forwarded to the person's profile view, where they can either accept or decline the friend request (Figure 14).

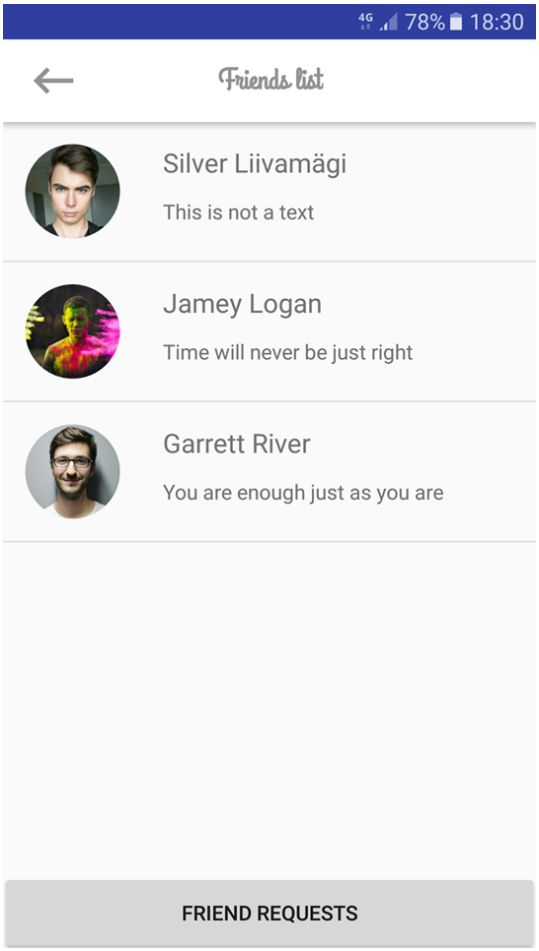

Figure 13. View of the friend's lists

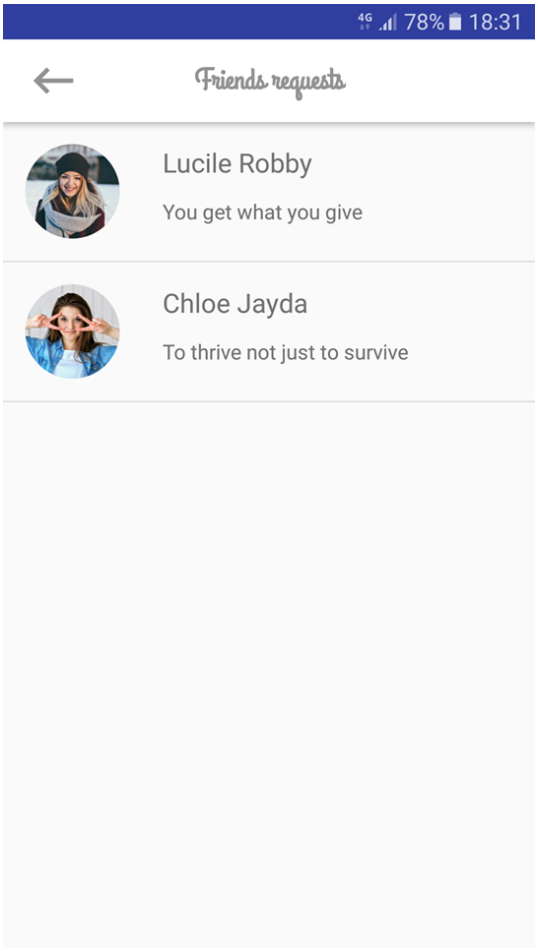

Figure 12. View of friend requests

3.3.14 View all users

All users view mainly consists of a list view that displays all the users that have registered on to the application. The displayed users are quarried from the database 10 at a time so that the whole list of users is never loaded in. Additionally, users can use the search option, at the top of the view, to find people. Each account is displayed to the user inside a card. A card houses basic information about the user. If the users click on a card they will be forwarded to the person's profile view, that they clicked on (Figure 14).

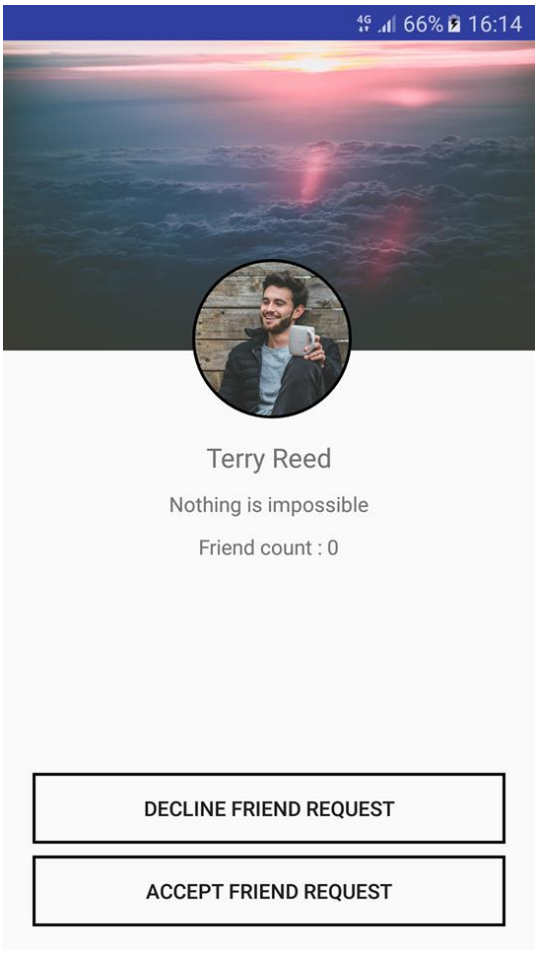

Figure 14. Detailed view of a user's profile

3.4 Firebase database

The database consists of seven different tables. Each table has its own purpose and usage in the application (Figure 15).

- "ActivityDatabase" is for storing data about users running activities. Data gets sent here when the user clicks on the "Stop" button inside the GPS tracker view.
- "Friend_req" is for storing information related to friend requests. When a user clicks to add another user as a friend, data is sent here. If the user who receives the request accept it, the request is deleted and a new relation is added to the "Friends" table. If the users who received the request declines it, the request is deleted from the "Friend_req" table and no other table gets altered.
- "Friend" table is for storing friend relations between users. Data is stored here if one user accepts another's friend request. If one user unfriends another, the friend relation is deleted from the table.
- "FriendsPosts" table contains all the data about posts that have been made private. Data is stored here if the creator of the post decides to share the post only with their friends.
- "Users" table contains all the users registered to the application. Data is stored here every time a new account is created. Data is edited in this table when a user changes information about their profile.
- "rafutfirebase" table contains data about every post in the public stream. Data is stored here if the creator of the post decides to make the post visible to all users.
- "notifications" table contains information about notifications that have been sent in the application. Data is added here when a user clicks to add another user as a friend.

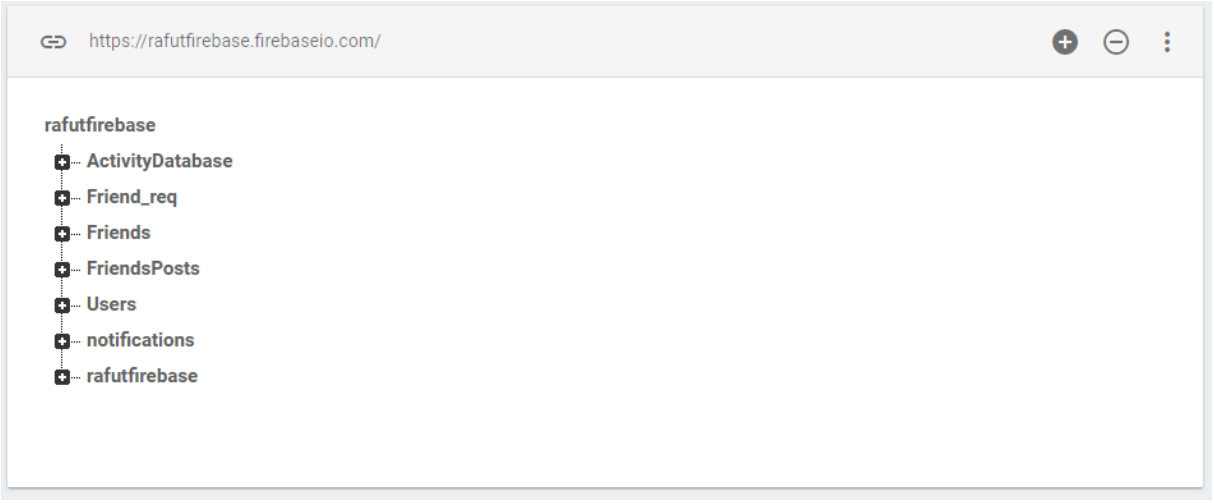

Figure 15. Overview of the database

3.4.1 ActivityDatabase

The ActivityDatabase table contains the exercising data of every user. Under each user's id there is a list of completed runs, each represented by a unique key. A key contains data about the date, distance, speed and time of the exercise (Figure 16).

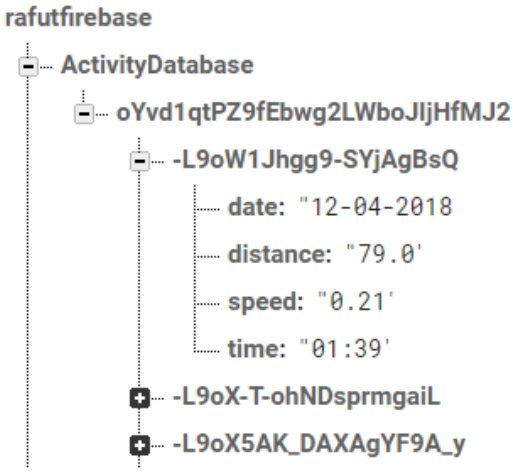

Figure 16. Overview of the ActivityDatabase

3.4.2 Friend_req

The Friend_req table contains information about users who have been sent or who have received a friend request. When user one clicks on "send request" on a user twos profile, a new node will be added into the table under both users' id's. User one will have a new child with the id of user two and the request type "sent". User two will also receive a new child with the id of user one and the request type "received" (Figure 17).

Figure 17. Overview of the Friend_req table

3.4.3 Friend

The friend table contains information about the friend relation between two users. When user one accepts user twos friend request both users will receive a new child under their ids. The new child contains the friend's id and a date when they were added as a friend (Figure 18).

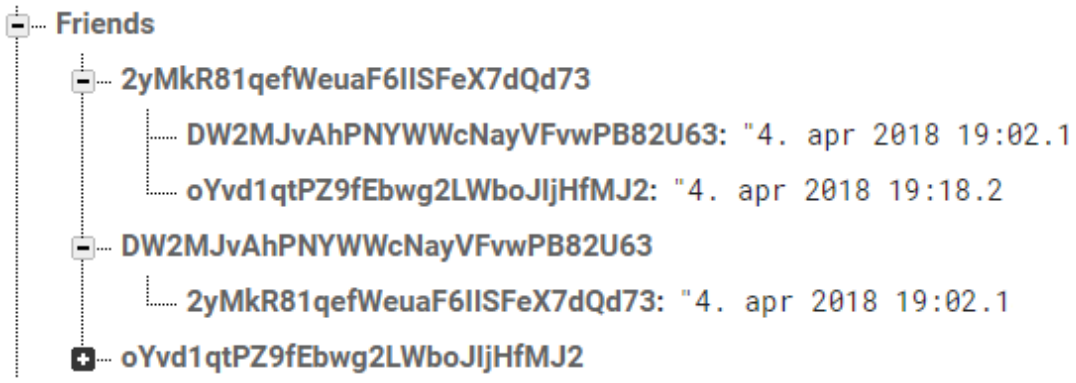

Figure 18. Overview of the Friends table

3.4.4 FriendsPosts

The friends post table contains every post that has been shared among friends. When the database is queried for friends posts the query looks at the posts uid. If the uid is in the user's friends list, they will be shown the post. Each post has associated with them a date, description, time, title, uid, username map details and an image. The image is link to a picture in the firebase storage. The image is saved there before the post is saved to the database. Map details are for the posts detail view (Figure 19).

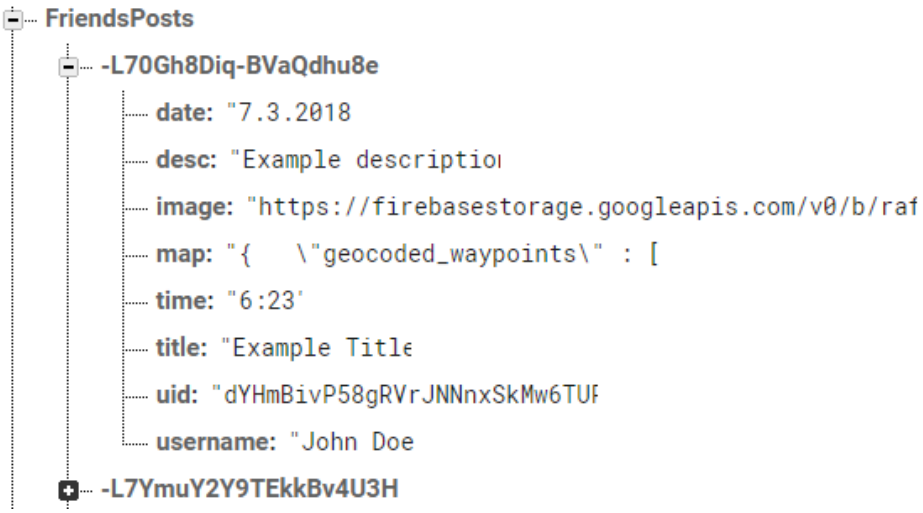

Figure 19. Overview of the FriendsPost table

3.4.5 Users

Users table holds information about all the users who have registered to the application. Each user has associated with them a name a uid and a device token. The device token is used for sending notifications. Rest of the fields are not compulsory, like the image, thumbnail image or status. In addition to the aforementioned fields users can also have fields for a birth date and location (Figure 20).

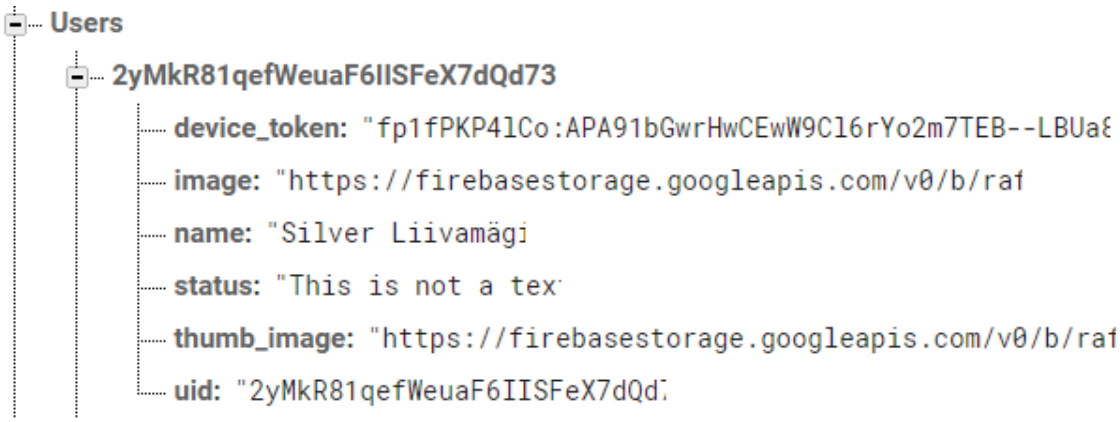

Figure 20. Overview of the Users table

3.4.6 Rafutfirebase

The rafutfirebase table holds information about all the posts that have been shared among all the applications users. Each key in this table corresponds to a post. Posts in the public stream contain the same values as the privately shared posts. Data is queried from this table to the public posts view five posts at a time and in the order of date they were created in (Figure 21).

Figure 21. Overview of the rafutfirebase table

3.4.7 Notifications

The notifications table contains information about all the notifications that have been sent in the application. Under each users' key is a list of all the notifications that have been sent to them. A notification contains two keys, a key for the user id who sent the id and the type of the notification.

When data is added to the notifications table, it triggers a function in the Firebase backend. The function accesses the table and retrieves the id of the user who sent the notification, the id of the user who receives the notification and the id of the notification itself. It also accesses the Users table and receives from there the device token. Once all the required data is collected, the notification is constructed and sent to the device (Figure 22).

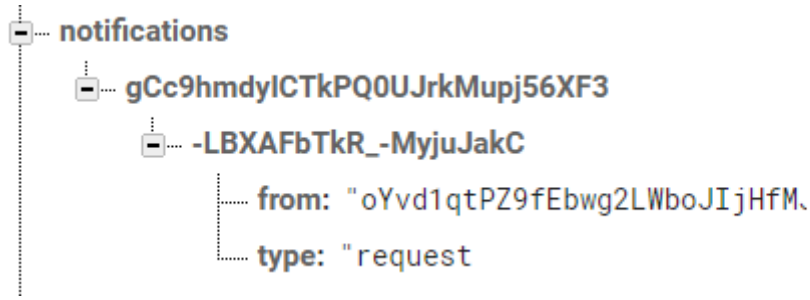

Figure 22. Notifications table

3.4.8 Conclusion

This chapter detailed the Android application that has been created in this thesis. The main focus of the application is to give its users a running tracker with a new way of socializing. The application consists of 16 different views and uses Firebase as its backend. Into the Firebase database the application saves data about its users, user's friends, posts and completed runs. The firebase backend also helps take care of sending notifications.

4. Usability testing

A group of four people for chosen to test the quality and usability of the application. The aim of the test was to get the opinions of potential future users and locate potential problems the users might face with the application. The test group consisted of two people from the age range of 20-26 and the other two were in the range of 40-46 years old. The devices used during testing were all running Android 7.0.

Before testing started the participants were given a quick overview of what the application is and what it can be used for.

Tasks which were asked to be perform:

- Register an account and setup a profile.
- Add someone as a friend.
- Accept someone else's friend request.
- Create a post.
- View posts created by other users.

Questions that were later asked from the participants:

- How easy on a scale from 1-5 did they find navigating the application.
- Would they use the application in the future.
- What sort of changes would they like to see made to the application.
- What were their impressions of the application.

During testing the author was next to the participant, who was testing, to send to them, at the right time, a friend request. Other than that the author avoided from interfering or answering questions during testing.

4.1 Users' opinions and conclusions

Based on the conducted tests users did not have many problem navigating the application, on a scale from 1-5 users evaluated their experiences as fours and fives. Most users felt like they could use the application as it stands, but at the same time pointed out that for a better user experience they would like to see some more features added. Among these features was an ability to set a distance range in kilometers to who can see the created posts. Another thing that

two testers noted was that in their opinion the design of the application could be improved. One user added that they could see themselves refraining from using the public post system, but would still use the tracker and the history features of the application.

In conclusion the application received mostly positive feedback from the test group. The suggestions that the test group pointed out are in the authors opinion valid and should be taken into account for future developments.

5. Conclusion and future work

As a result of the thesis, an Android application was developed which provides its users a way to track their sporting activities. The application displays the user's movements on a map and also keeps track of the speed, time and distance travelled during the exercise. If the user chooses to save the exercise data, the application can visualize it along with the rest of the recently saved data. Furthermore, users can create posts and through them invite other people to join for a run or for a similar sporting activity. Posts can be made either public or private. Private posts can only be seen by the people the user has chosen to add as a friend, public posts can be seen by everyone using the application.

5.1 Future work

As with most applications and software projects they are never quite finished and require constant updating and improvements. As the usability testing proved, the application created in this thesis is no different, there are still several features and tweaks the author wishes were implemented.

- Range for public posts

The user should be able to set a distance range to the posts they wish to see in their public posts stream. This would make it so that the user would only see posts relevant to their location.

- Detailed user profile

The application should have the ability to keep information about other variables of the user's health, such as weight and body mass index.

- Changing units

The application should give the user an option to change between metric and imperial units.

- Different modes

The application could include a feature for walking, hiking or bike riding.

From the start to the end of the thesis, the author has learned a lot about new approaches and methods, when it comes to software development. As such, to scale this application up and to

reach more users, the author feels that some of the core technologies of the application should be changed. The first few months of this year have introduced two new technologies that should be used in the future development of the application. These being Flutter and Cloud Firestore, the first would allow the application to reach both Android and iOS users, from the same codebase. The second would replace the current Firebase database because Cloud Firestore offers faster queries and scales better than the Realtime Database.

6. References

[1] "*6 Ways Social Media Changed the Way We Communicate*" Circaedu.com [Web]. Retrieved from: http://circaedu.com/hemj/how-social-media-changed-the-way-we-communicate/ [Accessed 19th of April].

[2] "*Health and fitness app usage "grew 330% in just 3 years*", Netimperative.com [Web]. Retrieved from: http://www.netimperative.com/2017/09/health-fitness-app-usage-grew-330-just-3-years/ [Accessed 19th of April].

[3] „*Android*", Techopedia.com, [Web]. Retrieved from: https://www.techopedia.com/definition/5415/android . [Accessed 10th of February 2018].

[4] "*Android, the world's most popular mobile platform*", Developer.android.com [Web]. Retrieved from: https://developer.android.com/about/index.html [Accessed 10th of April].

[5] "*Everything you need to build on Android*", Developer.android.com, [Web]. Retrieved from: https://developer.android.com/studio/features.html [Accessed 10th of February 2018].

[6] "*Push notifications using Firebase Cloud Messaging*", Tothenew.com [Web] Retrieved from: http://www.tothenew.com/blog/push-notifications-using-firebase-cloud-messaging/ [Accessed 30th of April]

[7] "*Firebase Authentication*", Firebase.google.com, [Web]. Retrieved from: https://firebase.google.com/docs/auth/ [Accessed 30th of April].

[8] "*Firebase Realtime Database*", Firebase.google.com, [Web]. Retrieved from: https://firebase.google.com/docs/database/ [Accessed 10th of February 2018].

[9] "*Firebase Cloud Messaging (FCM)*", Techtarget.com, [Web]. Retrieved from: https://whatis.techtarget.com/definition/Firebase-Cloud-Messaging-FCM [Accessed 30th of April]

[10] "*Java*", Theserverside.com, [Web]. Retrieved from: https://www.theserverside.com/definition/Java [Accessed 30th of April].

[11] "*Gradle overview*", Gradle.org, [Web]. Retrieved from: https://docs.gradle.org/current/userguide/overview.html [Accessed 11th of February 2018].

[12] „*Maps Android API Overview*", Developer.google.com, [Web]. Retrieved from: https://developers.google.com/maps/documentation/android-api/intro [Accessed 17th of April].

[13] „*Location and context overview*", Developer.android.com, [Web]. Retrieved from: https://developer.android.com/training/location/index.html [Accessed 17th of April 2018].

[14] „*Getting the Last Known Location"*, Developer.android.com, [Web]. Retrieved from: https://developer.android.com/training/location/retrieve-current.html#java [Accessed 17th of April].

[15] „*MPAndroidChart*", Github.com, [Web]. Retrieved from: https://github.com/PhilJay/MPAndroidChart [Accessed 6th of April].

[16] „*Picasso, a powerful image downloading and caching library for Android."*, Github.com, [Web]. Retrieved from: http://square.github.io/picasso/ [Accessed 17th of April 2018].

[17] „*CircleImageView*", Github.com [Web]. Retrieved from: https://github.com/hdodenhof/CircleImageView [Accessed 4th of April].

7. Appendix

7.1 Applications code

Code repository https://bitbucket.org/Experimentum/rafutfire

APK file: https://bitbucket.org/Experimentum/rafutfire/downloads/app-debug.apk

7.2 License

Non-exclusive licence to reproduce thesis and make thesis public

I, Silver Liivamägi,

herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

„Running tracker for Android with social group events“,

supervised by Satish Narayana Srirama,

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 03.05.2018