TFO "MAX VALIER"

PROJECT DOCUMENTATION

Runnergy

Fitness Tracking App

Fabian Feichter

supervised by
Michael Wild

School Year 2015/2016

Contents

1 Introduction

1.1 What is Runnergy?

Runnergy is an Android application, which is able to record an activity, which you covered during a sporting activity, for example, running or cycling. Before you start, you tell the app to start tracking your activity and when you are finished, Runnergy calculates the distance and the average speed and lets you view the route on a map.

If you decide to do the same activity one more time, the app tells you in certain intervals if you performed better or worse compared to a saved activity you did in the past. You compete against yourself. It is possible to compete against a specific, the best, closest-to-average or the worst performance of an activity.

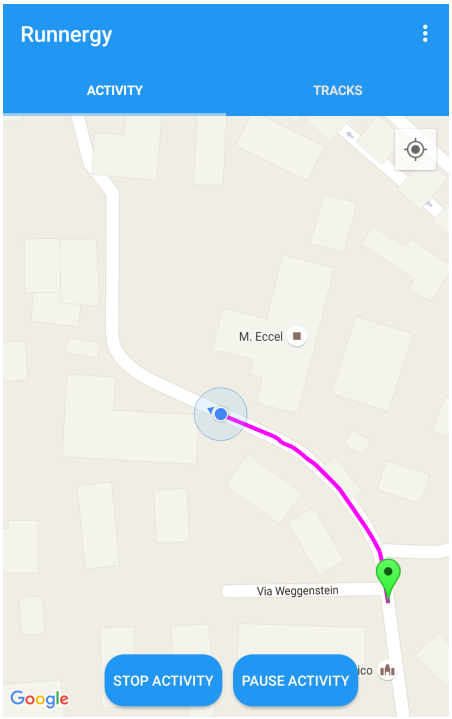

Figure 1: Running in Bolzano

1.2 Motivation

The reason why I developed this application is that I do a lot of cycling in my freetime. For me it is important to get better every time. When I get better it is the biggest success for me. For this reason I time myself when I do long distances. The longest distances I ride on my bike are up to 50 kilometres, usually in the

mountains. I make such long routes a few times a month. In the past, I only had the opportunity to figure out the total time difference at the end of doing a route multiple times, not the time difference during the activity. Important statistics like the distance I travelled and the average speed were also unknown.

Now I have Runnergy that tracks my activities on my phone. The app manages my activities and shows me their characteristics such as the average speed. The best comes when I do the same activity again: In a notification I can see how good or bad I am anytime. It is not even necessary to unlock the phone, a quick look to a notification is enough to see the difference. If I am slower than before, it can be a incentive to cycle faster.

2 Live tracking

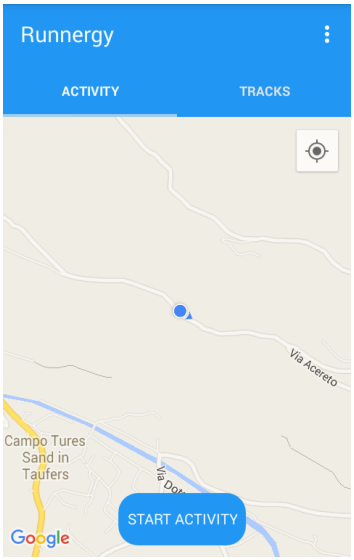

Figure 2: Main screen with current location

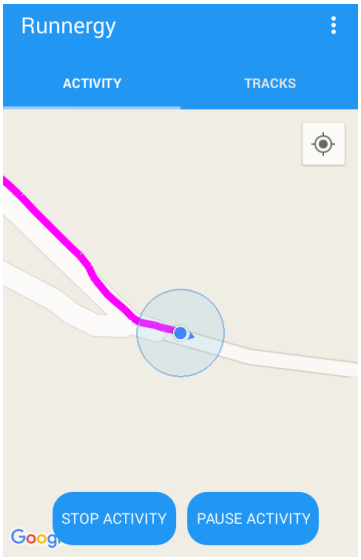

Figure 3: Live tracking of an activity

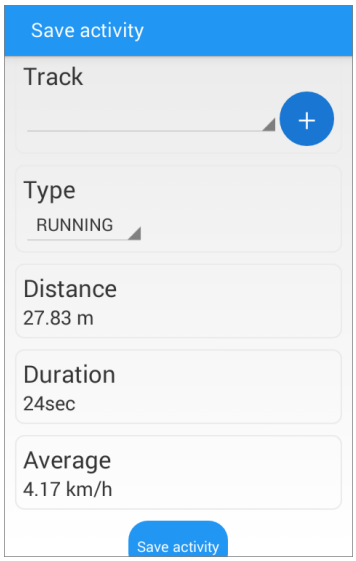

Figure 4: Save the activity

The main screen of Runnergy shows a map and if available, on the map there is a blue point which is your current location. Tap the button ”START ACTIVITY” and you can start an activity. The route will be drawn on the map. You can lock your phone, then the app runs in the background.

An activity can also be paused. A notification shows you the information if an activity is going on or not when your device is locked.

Tap the button ”STOP ACTIVITY” and your activity will be stopped. A new screen appears where you can see the distance, the duration and the average speed. You have to choose a type for the activity. The available types are:

- Running
- Walking
- Trekking
- Cycling
- Skiing
- Other (used when the type of your activity is not one of the above)

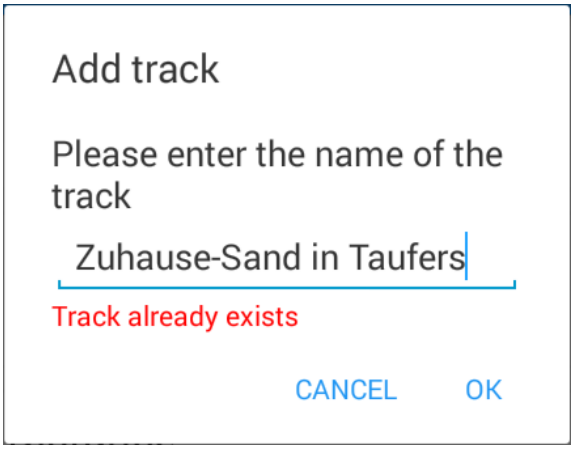

Figure 5: Insert a track

The last thing you have to do is choose or insert the track of this activity. After this you can save the activity.

3 Comparing two activities

If you want to challenge an activity, the first thing you have to do is choosing an activity. In the tab ”TRACKS” you can see the name of your tracks and how many activities each track has. They are sorted in descending order according to the number of activities in a track. This means that if you have two tracks, track ”A” has two activities and track ”B” has six activities, track ”B” is in the list before track ”A”.

If you tap on a track you can visit the activities of this track. They are grouped by the type of the activity. The groups are sorted in descending order according to the number of activities with this type. If you expand a group, you will see each activity of this group. The latest are up. Activities have one of the following backgrounds:

- **green with golden border and icon:** this activity is the best
- **yellow:** this activity is the closest to the average
- **red:** this activity is the worst
- **white:** this activity is none of the preceding

The unit of length and the date format can be changed in the settings.

After you choose an activity you will see the route on a map. Start and end point are marked. The green marker is the start and the red marker is the end point. If you start a comparison, first the app checks if you are on the start.

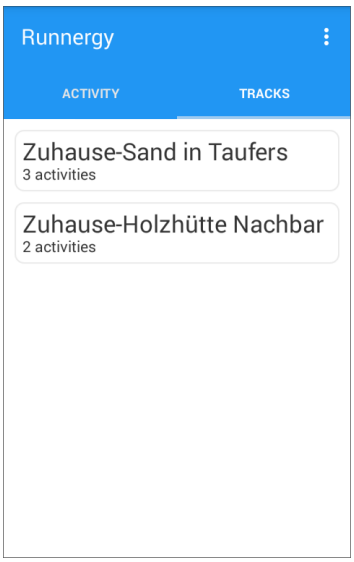

Figure 6: List of tracks, most popular are up

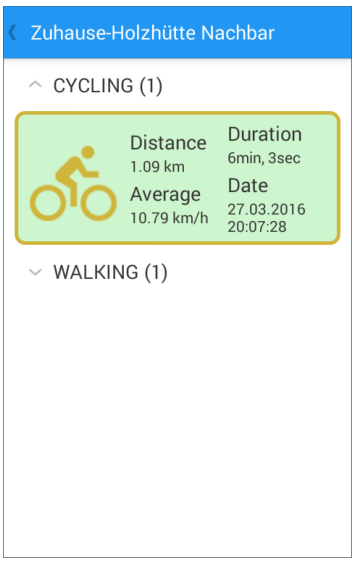

Figure 7: List of activities, divided into groups

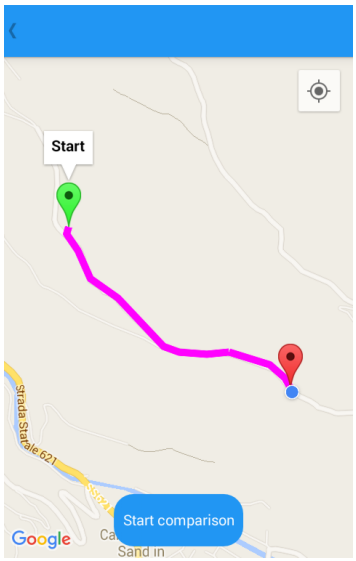

Figure 8: The route, shown on a map

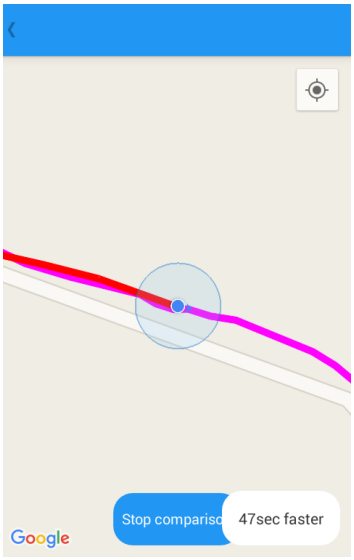

Figure 9: After comparison started you can see the difference

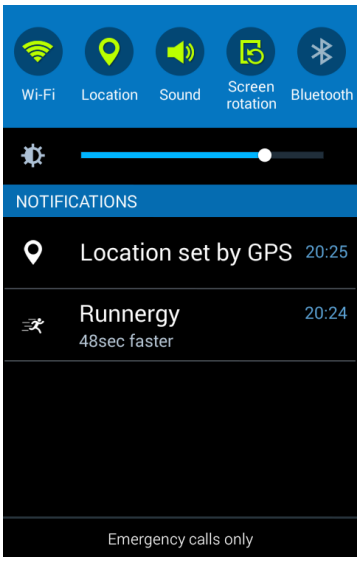

Figure 10: A quick look at the notification shows the difference

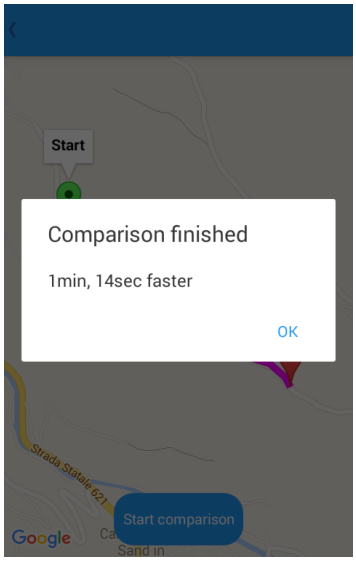

Figure 11: After you finished, you will see the final result in a dialog

Runnergy draws your current route over the existing route. You can see the difference of time between yourself and your virtual competitor in the lower right corner. If you switch to the home screen or you lock the device, you always have your time in a notification in view.

Stopping the activity has to follow that the final difference of time is displayed in a dialog.

4 Manage tracks and activities

It is very easy to manage your tracks and activities. It is possible to delete activities and tracks can be renamed or deleted. If you want to do that, you just have to perform a long tap on the appropriate item(s). In the CAB[1] you see the icons to change or delete the selected item(s).

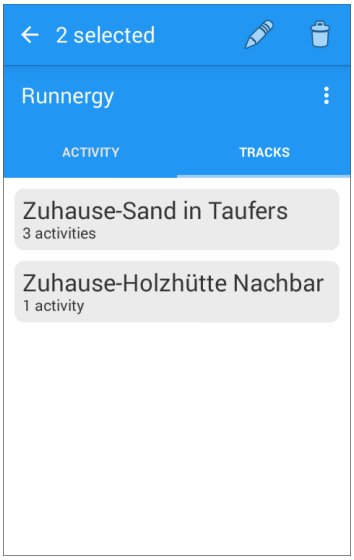

Figure 12: Two tracks are marked

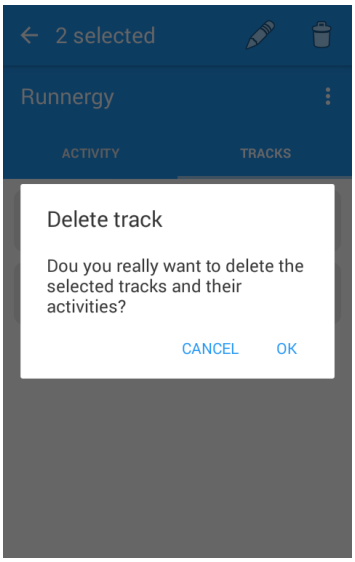

Figure 13: Delete them in one step

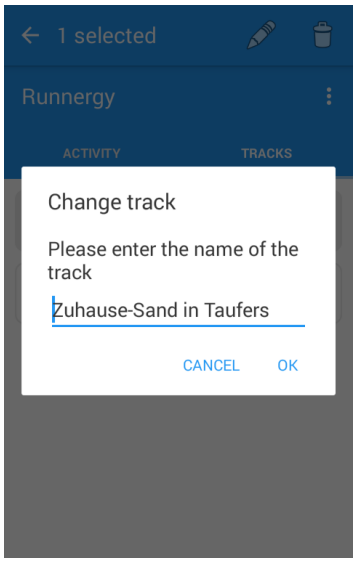

Figure 14: Change the name of a track

[1] A "Contextual Action Bar" is a temporary action bar, most typically used for tasks that involve acting on selected data or text

5 Software used

Java SE Development Kit 8	The Java Standard Edition Development Kit (JDK) is a development environment for building applications, applets and components using the Java programming language.[4] It was developed by Oracle. Runnergy is written entirely in Java.
Android SDK	In combination with the JDK the Android SDK allows to compile Java-based applications in the virtual machine of Android. The Android SDK also comes with some helpful tools like an emulator to test the application on a PC.
SQLite	Default preinstalled database management system on Android. One of the advantages is that SQLite databases created in Android are visible only in the application that created them.
Android Google Maps API v2	Provides several options for working with maps, for example draw a route on a map or calculate its distance.
Google Play Services	Provides high-quality, lower-powered location-based background services.
AppCompat v7	Brings Googles ”Material Design”, a design guideline, to devices running an older version of Android.
Design Support Library	Includes some important Material Design components. In this application it is used for switching between different views via tabs.[2]
Android Studio	The official Android IDE made by Google. It is based on IntelliJ.
GitHub	Hosts projects online.[3] GitHub uses the free software ”git” for file versioning.
Lockito	Lockito allows you to make your device follow a fake route, with total control over the speed and GPS signal accuracy. You can also simulate a static location. This was a must-have tool for me to test the app.
MiKTeX	MiKTeX is an all-in-one tool to write documents in LaTeX[4] and compile them as a PDF file. This documentation is written in LaTeX.

Table 1: Software used

[2]Used implementation: `https://github.com/google/iosched`
[3]Link to this project: `https://github.com/Feichta/Runnergy`
[4]For more details see `https://en.wikipedia.org/wiki/LaTeX`

6 FusedLocationProviderApi

6.1 What is the FusedLocationProviderApi?

The FusedLocationProviderApi is the latest location API made by Google. It is a powerful API that always gives you the latest and accurate location or location updates after an interval or on location change.

6.2 Advantages

This API has many advantages:

- It is easy to use for developers who want to interact with location information.
- Delivered coordinates are as accurate as possible
- Low usage of battery
- You do not need to think about best location provider, because it automatically chooses the best one suited for your hardware.
- Adjustable accuracy controls battery drain. The more accurate your location, the more power you are using. The four priorities are:
  - high = accurate within around 20 metres, about 7.25% per hour
  - balanced = accurate within around 40 metres, about 0.6% per hour
  - low = accurate within around 10 kilometres, battery drain is almost immeasurable
  - now power = passively listens for location updates from other clients and uses the best accuracy possible, additional battery drain is null.
- The minimum displacement between location updates can be set. If you do not move, the result is that these non-essential coordinates are not saved.
- If another client also requests locations, the FusedLocationProviderApi uses them instead of requesting new locations.
- Good documentation [5]

[5] See `https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi`

7 Database

7.1 Scheme

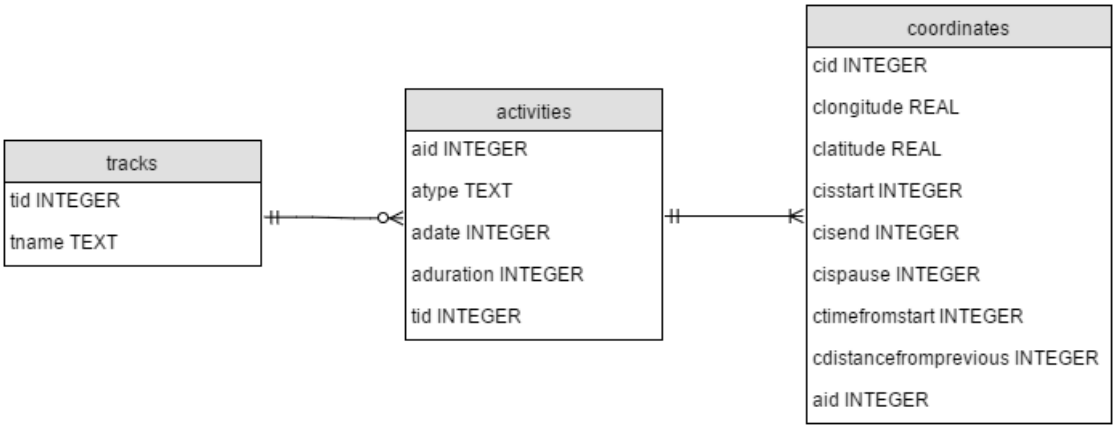

Figure 15: Schema of database

7.2 CREATE TABLE commands

```
CREATE TABLE tracks(
     tid INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
     tname TEXT NOT NULL UNIQUE);

CREATE TABLE activities(
     aid INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
     atype TEXT NOT NULL,
     adate INTEGER NOT NULL,
     aduration INTEGER NOT NULL,
     tid INTEGER NOT NULL,
     FOREIGN KEY (tid) REFERENCES tracks(tid)
          ON DELETE CASCADE ON UPDATE CASCADE);
```

Figure 16: Tables ”tracks” and ”activities”

```
CREATE TABLE coordinates(
     cid INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
     clongitude REAL NOT NULL,
     clatitude REAL NOT NULL,
     cisstart INTEGER NOT NULL,
     cisend INTEGER NOT NULL,
     cispause INTEGER NOT NULL,
     ctimefromstart INTEGER NOT NULL,
     cdistancefromprevious REAL NOT NULL,
     aid INTEGER NOT NULL,
     FOREIGN KEY (aid) REFERENCES activities(aid)
          ON DELETE CASCADE ON UPDATE CASCADE);
```

Figure 17: Table ”coordinates”

8 Activity diagram

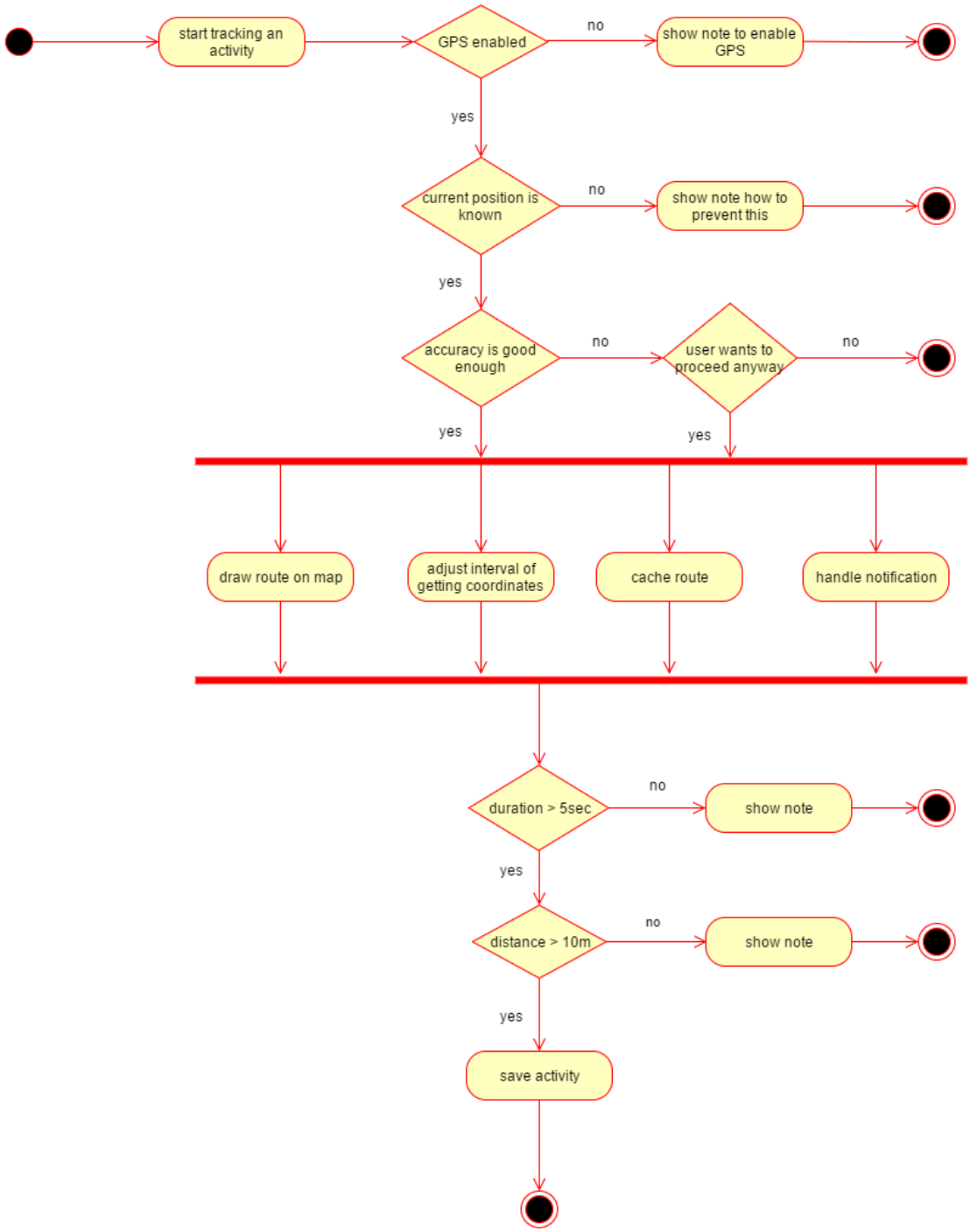

Figure 18: Activity diagram

9 Locations

9.1 Longitude and latitude

To describe a location on earth two values are important: longitude and latitude. Earth is subdivided into 180 degrees of latitude and 360 degrees of longitude. Degrees of longitude go through the North and South Pole, degrees of latitude run parallel to the equator. A location is the intersection of these two values.

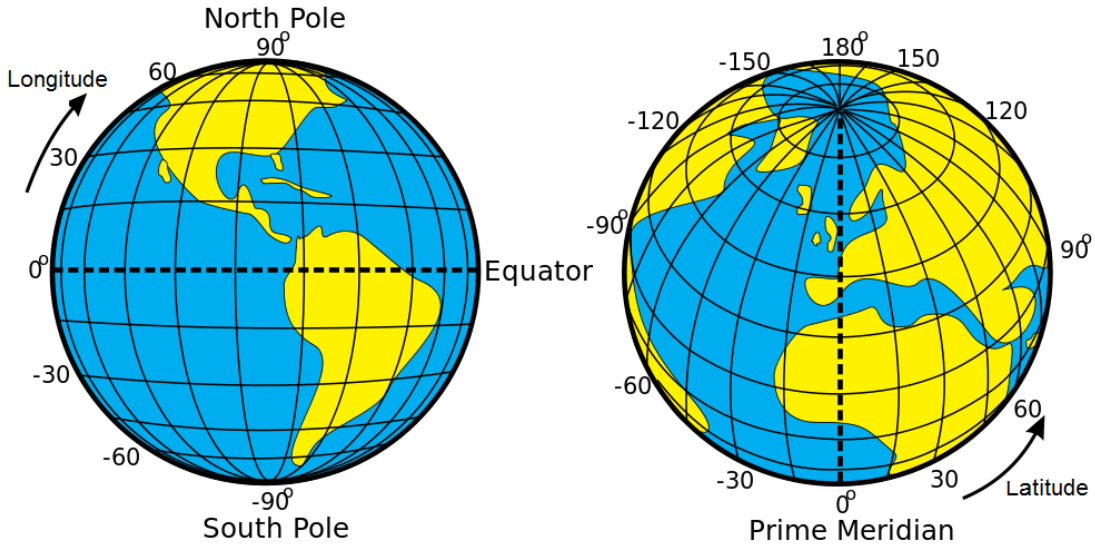

Figure 19: Map with longitudinal and latitudinal degrees[1]

For example, the ”Alexanderplatz” in Berlin has the longitude 13° 24’ 48” E and the latitude 52° 31’ 19” N. In addition measuring longitude and latitude in degrees, minutes and seconds, they can also be measured as decimal degrees.

$$Decimal\ degrees = degrees + \frac{minutes}{60} + \frac{seconds}{3600}$$

9.2 How does GPS work?

Devices need to know our current location in many everyday situations, think of the navigation system in a car. The technology that finds out where we are is called GPS. It stands for ”Global Positioning System.” This system is made by 24 satellites that are circumnavigating the earth in a height of about 20,000 kilometres. The number of satellites varies, because they have a limited lifetime and from time to time they are replaced.

These satellites constantly send radio signals containing the current time. This time and the time when the receiver receives the signal are needed to

calculate the distance to the satellite. To calculate the distance, the following formula is used:

$$s = c * (time\ receiver - time\ satellite)$$

Since c, the speed of light, is about 300,000 m/s, the time sent by the satellite must be accurate. To assure this, they use atomic clocks. They are so precise that the deviation is just one second in 300 million years.

With the calculated distance to the satellite, it can be guaranted that the receiver is within a specific radius of the satellite. But this is not enough to get the exact location as the figure shows.

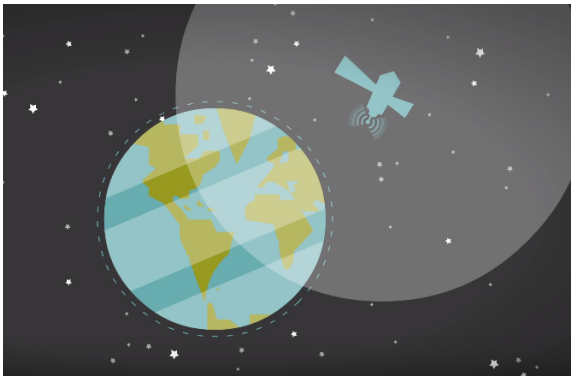

Figure 20: Current location is somewhere within this radius[5]

Try to get the distance from a second satellite and you get another overlapping sphere. Keep doing that and with just four measurements you can pin your location to one point: the intersection of the four circles. This will be your most likely location. Another intersection is in space, but this point will be ignored.

Figure 21: Your position is where the four radii intersect

Doing several requests getting the current location makes it possible to calculate the speed and the direction if you are moving.

To increase the accuracy, the formula to calculate the distance from the receiver to the satellite is extended with the displacement of the position of the satellite during sending the signal.

In all this there's a problem: Normally there is not enough space to put an atomic clock into the receiver. The weight of them is several kilos. This is why the quality of the GPS in phones is not always as good as we want it to be.

10 Expandability

10.1 Calorie counter

It would be great, if Runnergy can tell you how many calories you have burned. There are tables with approximate values of how much calories you burn in which sport. It would not be very difficult to extend Runnergy with this feature but the calculated value would never be exact because it is individual how much energy someone consumes. So this feature is irrelevant.

10.2 Ranking of the most active users

Another feature is a ranking of the most active users, based on the total distance they made. The distance of each user will be saved on a database on my own private server, the access to the database would handle a simple REST API in PHP.

The problem lies in the authentication of the user. How can a user on Android devices be identified? Authentication with username and password would do the job, but it is too much effort to implement a login screen, account registration, encrypt the transmission of passwords and so on. Another problem is that I want to keep the app as easy to use as possible. The integration of a (mandatory) user authentication in this manner would destroy this premise.

A method of authentication, where the user has to do nothing, is based on the email account of the user with whom he is logged in into Android. The problem here are devices running a distribution of Android which does not presuppose an account to activate Android.

Another solution is to authenticate the user by a device-id like IMEI[6]. This is a good solution but what if the user changes his/her phone? Nevertheless, this type is the best I have found so far, I will probably implement it in future.

[6] A number, usually unique, to identify GSM and UMTS mobile phones

11 A look back

11.1 My experience from this project

The greatest experience that I have gained from this project is that reading the documentation before writing the code is important to make a project successful.

11.2 Problems I faced

- **Fragmentation:** Android is an operating system for mobile devices, it is open source, intuitive and the diversity of devices is large.

  In my opinion Android has one big problem: To ensure such a huge diversity, software and hardware of Android devices often come from different manufacturers. Google is not the only company that builds mobile phones. Many manufacturers delay the delivery of updates, because they customize the operating system and every time Google releases a new version, they have to adjust and test the newly introduced features on their devices. Too many devices are only supported a few months or not at all. This makes it hard to develop an app that runs perfectly on every device.

  Apple, for example, solved this problem, because software and hardware comes from one manufacturer. A big part of their phones is running the latest operating system from Apple.

- **Accuracy of GPS:** As mentioned in paragraph 2.2 the accuracy of the GPS in phones is not the best because of their size. The results of the measurements in Runnergy are as good as the accuracy of the coordinates that the FusedLocationProviderApi provides. If the accuracy is bad, the results vary widely or are totally wrong. Hopefully, this technology improves in the future.

- **Native apps:** Using the device's native programming language has some advantages without opposition, for example you have the best possible device integration, but the main disadvantage is the development time. You have to learn the properties of each platform you want to run the app before you can start coding.

  However, there are mobile web applications too. Mobile web applications run in browser of the devices and operate across all platforms. They are distributed via the web and are not installed on the device itself.[3]

List of Figures

List of Tables

References

[1] Djexplo. Latitude and longitude of the earth. https://en.wikipedia.org/wiki/File:Latitude_and_Longitude_of_the_Earth.svg, 2011. [Online; accessed 24-March-2016].

[2] Freepik. Runer silhouette running fast free icon. http://www.flaticon.com/free-icon/runer-silhouette-running-fast_55240, 2016. [Online; accessed 24-March-2016].

[3] ltd michaels, ross & cole. Native mobile apps: The wrong choice for business? http://www.mrc-productivity.com/research/whitepapers/NativeAppsWrongChoice.pdf, 2016. [Online; accessed 24-March-2016].

[4] Oracle. Java se development kit 8. http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html, 2016. [Online; accessed 24-March-2016].

[5] Wilton L. Virgo. How does your smartphone know your location? https://www.youtube.com/watch?v=70cDSUI4XKE, 2015. [Online; accessed 24-March-2016].

Statutory declaration

I declare that I have developed and written this documentation completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked.
I agree that this documentation can be stocked in the schoollibrary for public use.

.....	.....
Location, Date	Signature