Social Li(f|v)e Tracking

Telecooperation Lab

Technical Documentation of the App Fitnessrace

Team: Quebec

Team Members:

Saad Bin Sajjad (Matr. Num. : 2383916) on Front-End,

Sayed Irfan (Matr. Num. : 2627928) on Back-end Development

Darmstadt, 6th March 2018

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Telekooperation
Prof. Dr. Max Mühlhäuser

Index

1. Motivation

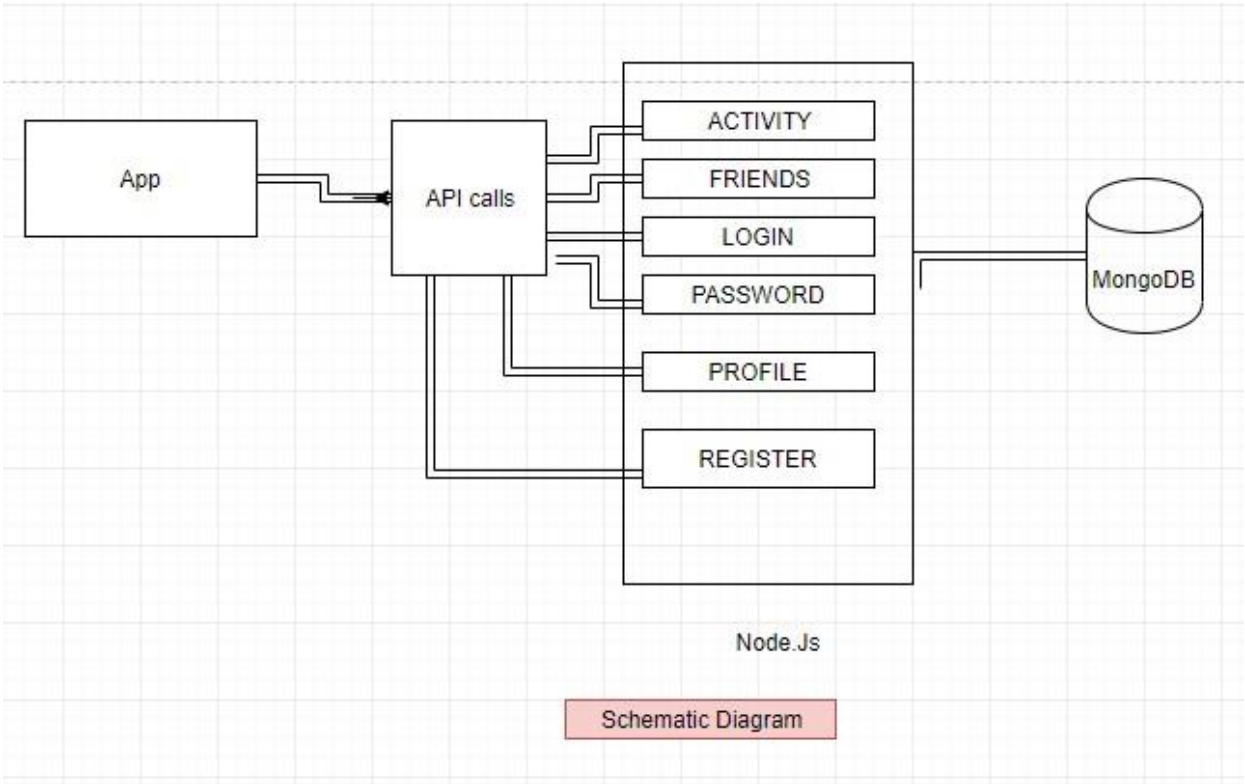
Fitness is one of the most desired ability of a human being. Although it is much expected but we do too little to acquire it due to our laziness or our busy schedule or simply just because it is boring. In this age of social networking we spend countless hours for socializing with our friends and share many activities that we do throughout the day. But most of the activities are just some photos, selfies, sharing random posts etc. But what if we could share our activities towards fitness which will be entertaining as well? That is the main thought behind development of the app Fitnessrace. It is the app that will help us get fitter by doing activities such as walking, running, biking and playing an augmented reality based game that will encourage people to be more active than their friends. At the end of the day you will be part of a race, race towards fitness which will make you healthy.

So the basic motivation is to gear up towards a healthy lifestyle by adapting with the ever increasing
Effects of technologies such as social media, smart phones , Augmented Reality etc. Besides, in the present age with pollution, lack of healthy foods and other unhealthy stuff, it is a crying need to have people involved more and more in fitness activites.And FitnessRace, our app provides a trendy way of doing it.

2. Architectural Overview and Design Decisions

The App is basically developed for android devices.
On the Front end we have used Google Activity Recognition API because it is very robust and we could report the activity based on the accuracy results. Same with Android Fused location API which reports the location very accurately. We used this API to trigger changes on the Google map as the user moves. Additionally, we sampled out results from Google Places API which we also used as the best Place Information service out. For showing the listed results of timeline,friend list or rewards, we used Recycler View along with Data Adapter to show it. All the network calls were done by using Retrofit2 with okhttp3 , which are one of many ways to make network calls. But these libraries were used because asynchrnous calls could be made and they have a well defined interface.

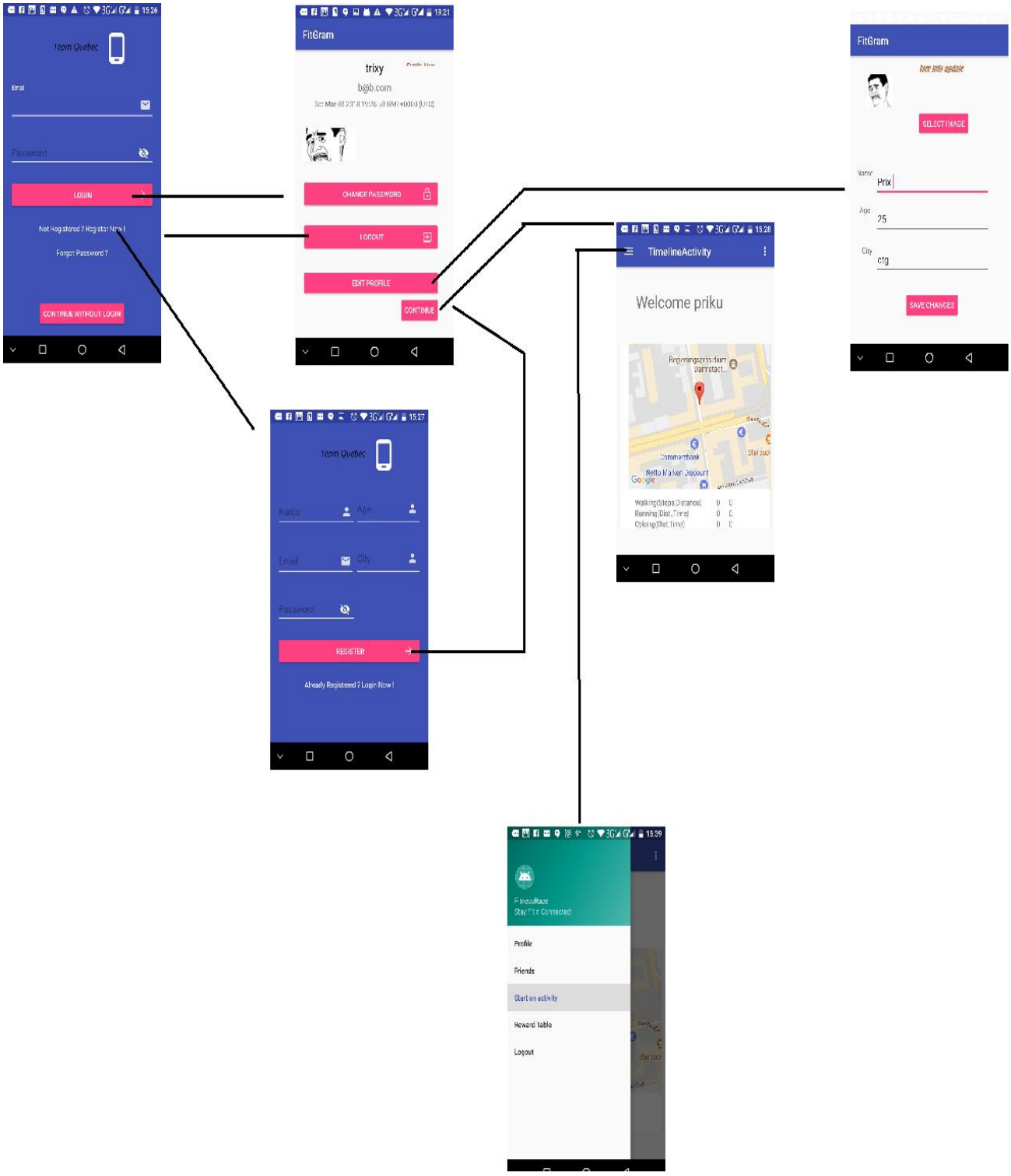
On the backend we have implemented the server using Node.js and NoSQL database MongoDB.
For resolving the app-calls several REST-APIs have been created. Sent data are stored on the collections of the schema created on MongoDB. The node server is hosted online on an online service called Heroku and the MongoDB is hosted on an online service called MLab. The REST-APIs which are implemented are POST api for user registration, authentication, password change, image uploading etc; GET api for user info, user-info for individual account, finding friend, showing friend list, activity sharing etc.; PUT for users achivements, rewards, making new friend, updating activity etc. ; DELETE for user account deregistration and previous data deletion etc.

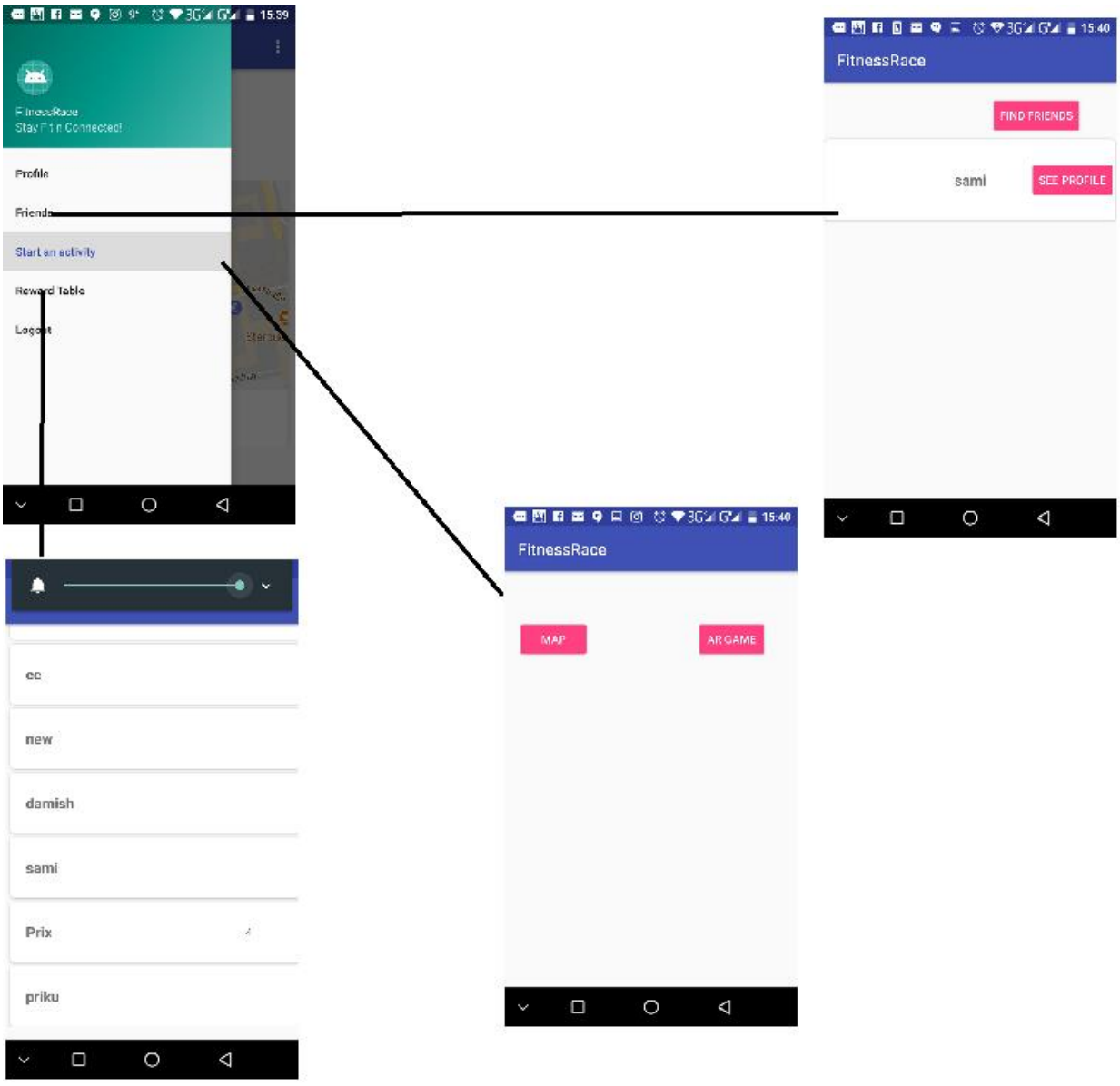


Schematic Diagram

3. Graphical Feature Explanation

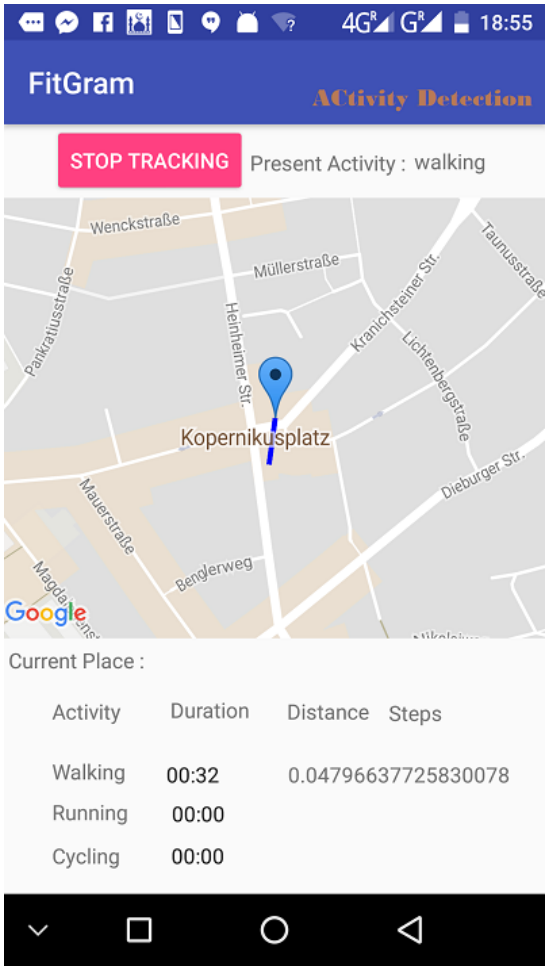
The wireframing is as below :

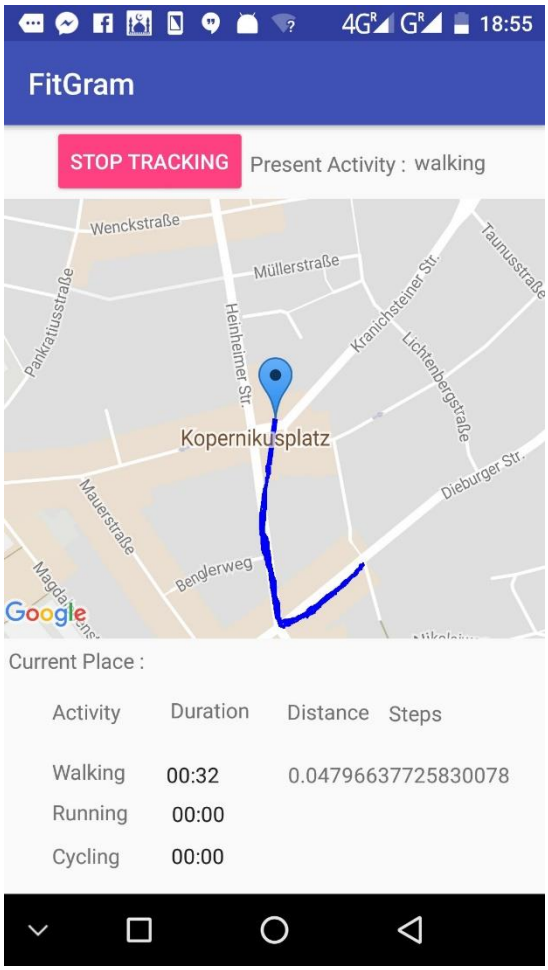




Here the connected screenshots indicates the sequence of the desired app screen flow.

The Main Activity screen (Map) where the users movement as well as location and activity can be tracked in real time. The Data is sent to the server when the user presses Stop Tracking button.(image below)

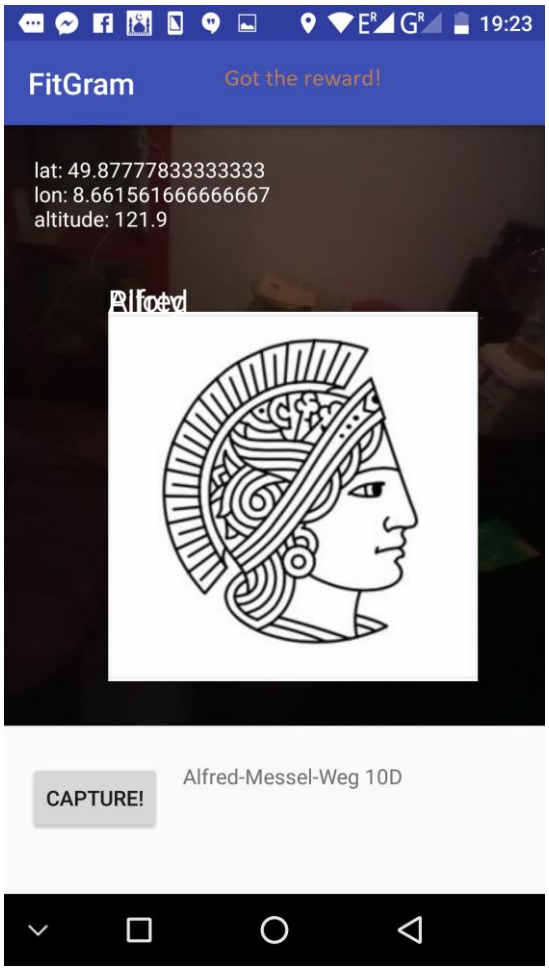




A path is drawn based on the locations covered by the activity, such as walking, running, biking.

A sample drawn path is shown on the image above.

The AR Game activity screen, when a Reward is found the capture Image button appears. The probable locations where the rewards can be found are shown to the user on the map.(image below)



Requirements and Instructions:

- Android phone with at least API Level 16 (Android 4.1).
- For phones which have Marshmallow or higher the Permissions of Location and Phone should be given by the user to avoid app crashes.
- Active Internet connection.
- Register on the App
- Keep in the pocket or wherever with you, it will start automatic detection of your activity.

Used Technologies on App side

- Google Activity Recognition API for detecting user activity
- Picasso Image Library for displaying Images
- Retrofit2 and okhttp3 for Network calls
- Android Sensor manager for detecting step count.
- Google maps and Places API
- Manual conversion of GPS coordinate system (Fused location API) to camera coordinate system for placing AR object in the camera view.

Used Technologies on Back end

- Node.Js for server
- Heroku online server
- MongoDb for NoSQL database
- MLab as the online database store

4. Features List

- **User Management**
- **User Profile**
- **User Timeline**
- **Physical Activity Detection**
- **Location Tracking**
- **Find Friends**
- **Friend List**
- **Activity Log**
- **Achievements and Rewards**
- **User data correction/update**
- **Automatic Activity Retrieve**
- **Implementation of Server in NodeJS**
- **Central database on MongoDB**
- **Implemented REST-API for the App-Calls**
- **Server end hosted online**
- **Bonus Feature- Augmented reality Game**