Report TrackFit
Project 2

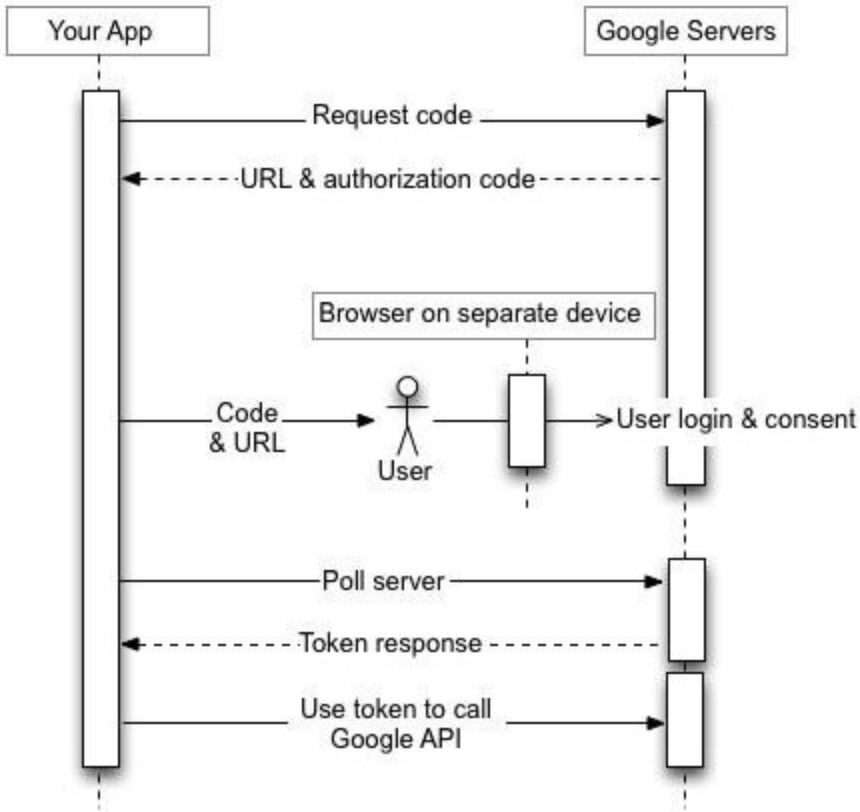
Introduction

The aim of this project was to offer google fit users the possibility to compare their activity data, more precisely their step data to the average of other users. Therefore we used the google fit API to ask each user for permission to get their google fit data. We then stored the data in a noSQL database named MongoDB. In the last visualization step we read the data from the database and computed a graph shown in html.
For programming we used Java 8

Methods:

Google fit API:

To get the user information of google fit we had to use the google fit API via OAuth 2.0. There we first had to perform a web service request to the google URL to get the authorization code. This code is then used for the user to log. In parallel our application polls the google URL at a specified interval. After the User gives us the permission The google server response gives us an access and refresh token. With the access token we can access the User data. The biggest problems were to understand the documentation and choosing the libraries.



Data model:
The idea of our model is that for increasing data there is no object that dynamically increases in size, which is also the reason why we used a noSQL database. To prevent this we based our data model on these three Collections:

- Steps
- Date
- User

The User Collection stores each User as one object. Each object contains informations like name, e-mail, individual id…… The Date Collection contains each full day as one object. Each object contains informations like StartTimeMillis, an individual id number, the average steps and a counter for the number of Users in the database. The Steps Collection contains the most objects and although the number of objects increases much faster than in the other Collections, the size of the objects is still fixed. Each object contains the UserID it belongs to, the DateID and the number of steps.
The biggest problems while creating and implementing the database and model were to understand and find all commands on how the database works. Additionally it was difficult to find the commands for java.

Visualization:
The output from the database contains the steps taken by the user and the average steps taken by all users using the apps. We visualize this data with the help of javascript and HTML. We are using a special javascript library called D3.js.It allows us to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document. For example, one can use D3 to generate an HTML table from an array of numbers. Or, use the same data to create an interactive SVG bar chart with smooth transitions and interaction. With minimal overhead, D3 is extremely fast, supporting large datasets and dynamic behaviors for interaction and animation. D3's functional style allows code reuse through a diverse collection of official and community-developed modules.

Therefore using D3, We have created a stacked line chart to represent the user data compared to the other user's data, with users data represented in blue and average users steps are represented with red line. The X axis indicates the date and month on when the users steps were measured, While the Y axis represents the number of steps taken by the user and the average steps taken by other users.

Further improvement of our program should include other fitness data available on google fit, like cycling or running data. A more complex improvement would be to also include fitness data of other platforms like Withings or Fibit.