  October 2019

TopFit GEOTRACKER

![A drawing of a person

Description automatically generated]()

DEVELOPED BY GROUP 11

Nadezhda Mokhireva 101149416

Parisa Khataei 101111190

Vladyslav Bordiug 100629242

Mahdi Esmaeelpour 101124172

Contents

[PROJECT DESCRIPTION 2](#_Toc21708031)

[PROPOSAL 2](#_Toc21708032)

[BACK END 2](#_Toc21708033)

[STORYBOARD 2](#_Toc21708034)

[MOCKUPS 2](#_Toc21708035)

# PROJECT DESCRIPTION

## PROPOSAL

## BACK END

The APIs in this project are responsible for making secure transaction between the Servers and Clients. Java will be used to for back-end coding and as it was mentioned before, APIs will help connect to the database and decouple the back end.

Storing user data securely is the top-level priority in the development. Not only it provides positive user experience but also enables developers to manipulate the data and later release an improved version of the application with additional functionality.

TopFit is going use three separate database tables for Authentication, Users Information and User Statistics. Database operations will be performed with MySql for two reasons. The first is that it is open source, the second is that all group members are familiar with it.

# STORYBOARD



# MOCKUPS

## USER INTERFACE

Upon launching the application, the user is presented with the login screen (figure 2). If the user does not have an account yet, they have an option to press the *Register* button. They will be redirected to the registration screen (Figure 1).

|  |  |
| --- | --- |
| Figure . Registration screen | Figure . Login screen  See figures 3 and 5  See figure 4 |

If the user enters incorrect email and / or password, they stay on the login screen and a message appears, indicating that the authorization failed (Figure 3).

|  |  |
| --- | --- |
| *Figure 3. Login Failure* | A screenshot of a cell phone  Description automatically generated  *Figure 4.Password reset* |

It sometimes happens that the user is registered but cannot remember the password. In such a case, they can restore it (Figure 4).

Upon successful login, the user will be directed to the main page of the application, a.k.a. *New Activity* page (Figure 5). To access the side menu (Figure 6), the logged in user can press the *Menu* icon in the top left corner from any screen. Pressing the *Logout button* willtake the user to the Login page.

|  |  |
| --- | --- |
| *Figure 5. New Activity* | *Figure 6. Side menu* |

At the *New Activity* page, the user can press *START* button to activate the geo tracking. The activity will be recorded until the *STOP* button is pressed.

Picking the *My Profile* option from the side menu will redirect the user to the following screen (Figure 7):

|  |  |
| --- | --- |
| *Figure 7. Profile* | *Figure 8. Activity History* |

The user can edit their height and weight measurements and see their total distance walked. Saving / Cancelling the changes will take them back to the main page.

The *Activity History* screen allows the user to view the recorded routes. Clicking on the date will display the route that was recorded on that day. The user also has an option to clear activity history by pressing the *CLEAR* button. This can be also done from *Settings* page (Figures 9 and 10).

|  |  |
| --- | --- |
| *Figure 9. Settings* | *Figure 10. Confirmation Dialog* |

From this page, the user can modify their information: name, photo and password. To persist the changes, they have to press *Save* button.

The user also has the options to delete their account or clear the activity history. In this case, a confirmation dialog will appear, warning the user. To complete the action, they have to press *OK*.

## FUNCTIONALITY

**Login:** All users will be able to login with their username(email) and password, on “Submit” button click in the form, authVerify() function will be called. It will check if the username and password combination matches the one in the database.

**Register:** If a user does not have an account, they will be asked to register for one. The registerUser() function will verify that all mandatory fields are filled out with the proper values, and none are left empty.

**Forgot password:** To recover a forgotten password, a user will be asked to input their email associated with their account. The forgotPassword() function will check if the provided email exists in the database, then the user will be emailed instructions on how to create a new password.

**Change password:** In case a user wants to change their password, they will be able to do it. The user must provide the current password in order to create a new password. The changePassword() password will then check if the entered current password matches the one stored in the database and if the new password meets the requirements.

**Deactivate account:** If a user no longer wants to use the app, they might have their account deactivated. The deactivateAccount() function will ask the user to provide the reason for deactivating their account (for feedback purposes) and their password to verify that it’s the user themselves not somebody else.

**Start recording:** When a user presses the button while they are walking or running, the system starts to record the way that the user is going through, time that user spend for running and other function will record.

**Save record:** The saveRecord() function writes a record to the database table. The record is essentially a set of coordinates that describe the user’s route.

**Stop recording:** The “Stop” button allows the user to finish recording an activity. When the user wants to stop the recording their activity, the stopRecording() method will be invoked to stop recording the coordinates.

**Delete activity:** It might so happen that a user has some activities recorded that they don’t want to be associated with their account. To handle that, there is the deleteActivity() method that allows users to delete unused or old activities to make some space for new ones. This method actually deletes a database record, so there’s no way to restore the information.

**Clear activity history:** Many records associated with one user stored in the database. Each entry consists of the time that the activity took (for example, the user was walking for 43 min 21 sec), the distance and the average speed. These entries make up the activity history. With the clearHistory() method, the user is able to delete all the records for their account.