

## Iteration #3 Plan

### 1. Objective

The purpose of this iteration is to implement the aggregation of activity data imported from a device to the activity tracker. At the end of this iteration, a user should be able to:

- Create an account
- Login in to an existing account
- Import activity data and visualize it
- Display aggregated data including:
  - Average Distance
  - Average Pace
  - Calories Burned
  - Average altitude gain
  - Average altitude loss

### 2. Use Case

For this week, we decided to implement the third use case - the display activity statistics. This use case naturally follows our progress done on the first two iterations in which we:

- Implemented a database which made possible the creation of a user's profile and validation of credentials
- Successfully imported activity data from a "device" and saved it in the database.

This iteration will take into consideration the format of the data provided by the client in order to manage it. For this part we made the assumption that all future data will follow the same format.

The user will be able to sort/filter the statistics displayed in two ways:

1. Over a set time (date) period
2. By setting start and end days

The user will thus be able to see the statistics related to their statistics through a filter. The statistics that will be displayed are:

- Average Distance run
- Average Pace
- Calories Burned
- Average Altitude gain
- Average Altitude loss

### **3. Display Activity Statistics**

Name: Display activity statistics

Description: The user would like to display activity data as well as statistics regarding this data so they can analyze their activities and gain insight on it. They would also like to be able to sort this data and display it for convenience.

Basic Flow:

1. Registered User signals intent to view activity data
2. System pulls the relevant data from a database
3. Statistics, such as pace are calculated from the data
4. System displays the data and statistics in a neat form

Alternate Flow:

- 2a Database does not contain any records of the activity
  1. System displays a message notifying the user
- 2b System can not connect to the database
  1. System displays an error message
- 4a. User selects a different metric to sort data
  1. System sorts data with respect to that metric
  2. System displays data in the sorted order
- 4b. User sets up a different length of lap (such as 1km, 2km etc)
  1. System recalculates statistics
  2. System displays data

Actors:

Registered User – has preexisting account and is logged in

Persistence Entity – receives the intent to display activity

Data Source – this entity contains records of the activity

Data Parser – this entity exchanges data from the database to the persistence entity

Preconditions:

The user must have an account and they must be logged in.

The persistence entity must exist

Postconditions:

System must have displayed the activity information with the option to set up laps and sort the data differently.