

Assignment 1


Team Number: 22

Team Members

| Name | Student Nr. | Email |
|----------------------|-------------|-----------------------------|
| Martynas Rimkevičius | 2708302 | m.rimkevicius@student.vu.nl |
| Erik Vunsh | 2696857 | e.vuns@student.vu.nl |
| Baher Wahbi | 2707520 | b.wahbi@student.vu.nl |
| Abhisaar Bhatnagar | 2694178 | a2.bhatnagar@student.vu.nl |

Introduction

Author(s): Abhisaar Bhatnagar

BEAM™ is a desktop-based personalized, sporting application which allows its users to analyze their  physical activity (ranging over many sports) all by simply uploading GPX format files at the click of a button. This application is inspired by the likes of [Strava](#), [MapMyRide](#), and [GPXSee](#). The user chooses a sport, uploads a GPX file, and is met with unique metrics and graphs for each corresponding type of activity.

This application is designed with a user interface targeted for easy maneuverability and access by all of its users, mainly comprising of:

- **Active User:** A user who makes use of the app's basic functionality to track its basic metrics for the default sporting activities (i.e. running, walking).
- **Athlete:** A user who makes use of the app for advanced functionality to track basic and additional metrics of any sporting activity and set timely goals.
- **Coach/Trainer:** A user who makes use of the app for analysis of specific users to track all their metrics, activity history, and graphs.

The main features **all users** can interact with comprise of:

- ***Basic Functionality:*** Extract data from GPX files and output metrics visually and numerically
- ***Additional Functionality:*** Account system, timely goal creation, activity history, fun facts per data analyzation, calorie descriptor.

The **main modules** of the application consist of:

- **GUI:** app window which allows input/output interactions
- **Sports Data:** extracts the activity data and analyzes the metrics according to the assigned sport.
- **Data Representation:** translates and formats sports data to graphic/visual representations
- **User Accounts:** retrieves and stores user data (height, weight, age... etc) and allows access to activity history

The final application will be presented as an interactive fitness application based on the aforementioned main modules where the user interacts with the application and its respective functionalities. The way the application will operate is as illustrated:

- On the first ever app launch the user is prompted to input their personal data to be used in later metric calculations.
- The user interface will allow the user to navigate to the respective areas of the app they require i.e. GPX input, activity history, goal creation.
- The application will display the corresponding accessed screen.
- The user can input a GPX file of their activity for which the application will parse, display and store all its data.
- Each GPX File input is taken as a separate activity which will then be saved and can be accessed for future references.

Features


Author(s): Martyn Rimkevičius, Baher Wahbi, Erik Vunš, Abhisar Bhatnagar.

| ID | Short name | Description | Champion |
|----|--------------------|--|----------|
| F1 | GUI | Graphical User Interface to interact with all the features below. | Baher |
| F2 | Base Metrics | Analyze GPX file for base metrics: <ul style="list-style-type: none">• Total Distance• Total Time• Average Speed• Calories Burnt | Martynas |
| F3 | Additional Metrics | Analyzes GPX files for additional metrics expanded from base metrics depending on the sport selected. Such as: <ul style="list-style-type: none">• Cycling<ul style="list-style-type: none">○ Speed○ Power | Martynas |

| | | | |
|------|---------------------------|---|----------|
| | | <ul style="list-style-type: none"> Swimming <ul style="list-style-type: none"> Pace per 100m Stroke Length Running <ul style="list-style-type: none"> Pace Elevation Cadence | |
| F4 | Map GPS Coordinates | Display coordinates that are in the GPX file on a Google Maps environment. | Abhi |
| F5 | Graphs | Display sports metrics change over time on a graph so that a user could see their performance during the activity. | Martynas |
| F6b | User Goals | Custom user goals which are updated with each GPX entry. For example, a user sets a 100km running goal for their week. | Erik |
| F7b | Activity History | A saved history of all activities which can be accessed and reviewed by the user. For example, visualizing past activity data. | Baher |
| F8b | Account System | Users are prompted to input their information: <ul style="list-style-type: none"> Weight Height Gender Name Age | Baher |
| F9b | Fun Facts | A fun fact displayed to the user per GPX file input. For example, "You ran 47% of a marathon today". | Abhi |
| F10b | Calories Burnt "Per Meal" | Users can choose to visualize their calories burnt as a specific food. For example, "On today's run you have burnt 14 bananas". | Erik |

Quality requirements

Author(s): Martynas Rimkevičius, Baher Wahbi, Erik Vunš.

| ID | Short name | Quality attribute | Description |
|-----|------------------------------------|---|---|
| QR1 | Consistent Navigation | Usability  | By moving through different screens the user will end up at the screen that is expected. For example, when going back, the screen will change to the prior one. |
| QR2 | Extendable Sport List | Maintainability | It is easy to add new sports (with their respective metrics) to the architecture. |
| QR3 | Comprehensible Data Representation | Usability | Data (text, map and graphs) is structured and outputted in a clean and understandable manner. |
| QR4 | User Data Requirement | Availability | Unable to input or access GPX files without creating an account and filling user data. |
| QR5 | User Data Validation | Reliability | User data is in a reasonable range, such as age: 16-110, weight: 30-200kg and height: 100-230cm. |
| QR6 | User Data Safekeeping | Security | User data is encrypted with base64 encoding on input. |
| QR7 | Compatible Metrics | Availability | Each sport will only compute their own respective metrics. |

Java libraries

Author(s): Martynas Rimkevičius

[JavaFX](#) - used to create a GUI for the application.

[GMapsFX](#) - used to display Google Maps in the application without directly interacting with the Google Maps API.

[Graphhopper](#) - used to easily display GPX coordinates on a map.

[SonarLint](#) - used to ensure that all code adheres to the same style and practice.

[Gpx-parser](#) - used to extract data to a Java object and access metrics easily.

[GSON](#) - used to work with JSON files so that they could easily be converted to an object.

[Dex](#) - used to draw graphs visualizing activity data.



Time Logs

[illegible]