ID3 Update for Software Requirements

1. Introduction

1.1 Purpose

Commercial wearable devices have emerged as promising tools for measuring physical activity both in laboratory and real-world settings. However, researchers have faced challenges extracting these data in an efficient, scalable way. In this paper, we describe the Built Environment and Active Populations (BEAP) Engine, a web application/tool we developed to collect, process and analyze physical activity data exported from Apple Watch and Fitbit devices(As a value added) the tool also predicts past physical activity (from metrics such as energy expenditure and distance walked) using machine learning methods, and returns the file in a more user-friendly format

1.2 Intended Audience

The app is intended for researchers focusing on physical activity measurement to predict past physical activity.

1.3 Scope

Explain the scope of the software. What are the main goals and objectives? (In order of priority)

1. Deploying the web application in a functional preliminary state

2. Extensively document all involved repositories to make project more readable and open for any future work

3. Provide team feedback on difficulties and challenges during development process

4. Create coverage in Java and React components for a potential addition of Garmin done by Professor Fuller

5. Any kind of speed (performance) improvements are helpful because it was slow.

2. System Features and Requirements

2.1 Functional Requirements

1. Upload Fitbit, and Apple watch data by the user

2. Improving overall user experience especially when displaying data

3. Logging in for users

4. Logging out for users

5. User access to Process/Predict Fitbit and Apple watch data using SVM, Random Forest and Decision Tree

Nice to have Features:

1. Upload Garmin data

2. Process/Predict Garmin data

2.2 External Interface Requirements

1. A login functionality

2. A logout functionality

3. Predict functionality

    a. Start processing prediction of activity based on chosen device (Fitbit or Apple Watch) for the file that is uploaded.

4. Selecting Data to be Processed

    a. This is in form of a radio button where user can choose which data they want to process

5. Process data functionality

    a. Process data to ready for prediction

6. Upload file functionality

    a. This is in form of a drop zone where a user can upload as many files as they want to ready for processing.

7. Navbar for navigating user functionalities

8. Progress bar functionality

    a. A bar showing how much of the process for uploading or the prediction is done

9. A sign up functionality

10. Data Deletion functionality

11. Ability to choose what device's files will be manipulated

    a. This is in form of a radio button where user can choose which smart watch data they are uploading to the system

12. Choosing the format in which the data would be displayed

    a. This is in form of a radio button where user can choose the format they want to view their data or process their data

13. Download the manipulated data

    a. A button where they can download the processed data or predicted data

2.3 Nonfunctional Requirements

1. The app should be able to handle large amounts of data files in a performant way.

   - Data file(s) coming from user can be one or multiple files

   - Max Total File upload will be 500 MB

   - If it is more than 500 MB that user will get a warning to split it in two uploads

   - Stakeholder expected that there will not be no more than 10 people uploading files to the system at the same time, but we will try to go for a higher maximum on the number of people

2. Security

   - Employ the TLS security protocol built into HTTPS requests

   - Making use of the database crypto module to ensure safety of data stored inside of database

3. Processing the data faster/ more efficient or a feedback that the server is busy for accepting any process

- Minimize unnecessary database operations if possible