

Software Engineering  
14:332:452  
Spring 2013

## Fitness Friend



### Group 12 Members

Jose Villamor-Delgado  
Cody Goodman  
Jie Huang  
Kyle Raucci  
Florian Pranata

## Contents

|   |    |
|---|----|
| 1. Customer Statement of Requirements .....                 | 4  |
| 1.1 The Current Need for Fitness.....                       | 4  |
| 1.2 Health Data Analysis .....                              | 5  |
| 1.3 Making Fitness Fun.....                                 | 5  |
| 1.4 Making Fitness Affordable .....                         | 6  |
| 1.5 Ease of Use .....                                       | 6  |
| 2. Glossary of Terms.....                                   | 7  |
| 3. System Requirements Analysis.....                        | 7  |
| 3.1 Functional Requirements Table .....                     | 7  |
| 3.2 Non-Functional Requirements Table .....                 | 8  |
| 3.3 On Screen Appearance Requirements .....                 | 9  |
| 4. Functional Requirements .....                            | 10 |
| 4.1 Stakeholders.....                                       | 10 |
| 4.2 Actors and Goals.....                                   | 11 |
| 4.3 Use Cases  UC - 1: Visitor Registration (REQ -10) ..... | 11 |
| 4.4 Use Case Diagram.....                                   | 13 |
| 3.5 Use Case Traceability Matrix .....                      | 13 |
| 5. User Effort Estimation .....                             | 14 |
| 5.1 Use Case Effort Estimation.....                         | 14 |
| 5.2 Use Case Effort Classification .....                    | 15 |
| 6. Domain Analysis.....                                     | 15 |
| 6.1 Domain Concept Definition Table.....                    | 15 |
| 6.2 Association Definition Table .....                      | 15 |
| 6.3 Attribute Definition Table.....                         | 16 |
| 6.4 Domain Traceability Matrix .....                        | 16 |
| 6.5 Mathematical Models.....                                | 18 |
| 6.6 System Operation Contracts .....                        | 20 |
| 7. Interactive Diagrams .....                               | 21 |
| 8. Class Diagram and Interface Specification.....           | 25 |
| 8.1 Class Description Table .....                           | 25 |

|   |    |
|---|----|
| 8.2 Class Attribute Tables.....               | 25 |
| 8.3 Class Traceability Matrix.....            | 26 |
| 9. System Architecture and System Design..... | 27 |
| a. Architectural Styles .....                 | 27 |
| b. Identifying Subsystems.....                | 27 |
| 8.4 Class Diagram.....                        | 27 |
| c. Mapping Subsystems to Hardware .....       | 28 |
| d. Persistent Data Storage.....               | 28 |
| e. Network Protocol .....                     | 28 |
| f. Global Control Flow.....                   | 29 |
| g. Hardware Requirements.....                 | 29 |
| 10. Algorithms and Data Structures.....       | 29 |
| 10.1 Algorithms .....                         | 29 |
| 10.2 Data structures .....                    | 29 |
| 11. User Interface.....                       | 30 |
| 11.1 Website User Interface.....              | 30 |
| 11.2 Android App User Interface.....          | 33 |
| 12. Unit Test Cases .....                     | 35 |
| Test Coverage .....                           | 36 |
| Integration Testing.....                      | 36 |
| 13.1 History of Work .....                    | 37 |
| 13.2 Current Status.....                      | 37 |
| 13.3 Future Work .....                        | 37 |
| 14. References.....                           | 39 |
| Summary of Changes.....                       | 40 |
| Functional Requirements .....                 | 40 |
| Contribution Breakdown.....                   | 42 |

# 1. Customer Statement of Requirements

## 1.1 The Current Need for Fitness

There are many reasons for which people strive to improve their health and fitness level, unfortunately there are equally as many reasons for which these same people do not reach their goals. Regardless of the reason for setting health and fitness goals, everyone could benefit from a system that addresses the most common deterrents that people face while trying to improve their health by embarking on a fitness program or regimen. A system that minimizes these deterrents would allow an individual to remain more focused on achieving their health goals. Some common problems faced when committing to improving one's fitness are listed below.

- Self consciousness and insecurity
- Lack of health knowledge
- Lack of motivation
- Gym and training costs

There are currently two solutions commonly available to address these issues. The first being joining a gym and hiring a trainer. Although this addresses many of the listed issues by providing the knowledge and motivational skills of a trainer, this solution can often be costly and often requires the participant to maintain regular appointments with a trainer and make routine trips to the gym. The main alternative to this is to utilize at home training exercises such as P90X, Insanity or simply jogging and eating right on your own. This alternative is often more cost effective and allows an individual to train in the comfort and security of their own home or around their neighborhood. The biggest drawbacks to this alternative is that without professional knowledge it can be difficult for someone to monitor their own progress and make effective changes to their diet and workout regimen. Even with the use of health monitoring devices and diet trackers it can be difficult and inconvenient for an individual to compile and analyze their fitness data on their own. What would really help in making fitness goals more easily attainable would be a single product that could bring together all the benefits of both solutions without the drawbacks. Such a product should do all or most of the following.

- Make fitness fun
- Make fitness affordable
- Make fitness easy to track
- Make it easy to analyze tracked data
- Provide relevant information and fitness suggestions based on progress and tracked data
- Make important health data easily and readily available

## 1.2 Health Data Analysis

As stated previously there are an assortment of monitoring devices capable of providing a user with various health and fitness data such as the Motorola Motoactv (right). While technology has been rapidly improving, the capabilities of such devices are expanding and can monitor and record data regarding everything from heart rate, to hours slept, to the amount a person perspires. Despite this improvement in medical technology, most of these devices are design to provide nurses and doctors with data on patients under their direct care.

It is less common to have these devices directly provide the user with data on their own health, mainly because the user is unlikely to know the best course of action to take based upon such data. What would be very useful to health and fitness conscious consumers would be a system that in a sense cuts out the need for regular check ups by not only providing the user with pertinent data but providing professional analysis of long and short term data and making health and fitness suggestions based on a combination of recorded data and user inputs. For example the system may suggest, that a user whose breaths per minute sharply rises with their number of steps take, should exercise more often to increase their stamina. Another example would be combining a user provided symptom such as Chronic Fatigue Syndrome (CFS), with a long history of having less than 5 hours of sleep per night, to suggest that the person consider changing one's sleep schedule to ensure a greater amount of sleep per day.



## 1.3 Making Fitness Fun

One of the most common reasons people fail to reach their fitness goals is a lack of motivation due to the large amount of effort fitness often requires for small increments of progress. Often people lose their motivation because they do not immediately begin to see results and decide that the large amount of effort is not worth the incremental gains. Two things people often look for when embarking on a fitness regimen are a way to make fitness fun and a way to track small improvements in order to keep one's self motivated. One common way to do this is to work out in pairs or groups in order to observe each other's progress and to keep each other motivated, however this is sometimes difficult if friend's and partner's schedules do not perfectly line up. With today's heavy presence of social networking sites and mobile apps everyone uses sites such as Facebook and Twitter to tell their friends about everything from the meals they eat to the movies they go see. Being able to directly share fitness data and progress with friends on these social networking sites could potentially be a great motivator, allowing friends, family, coworkers, etc to post supporting comments to help maintain motivation. These social networking sites could also easily be used to turn fitness into a game amongst friends by providing a platform on which friends can compete with each other and brag or comment about their progress. Simple games could be created from data logged by various health monitoring devices such as pitting a group of friends against each other to see who can jog the longest distance in an hour, day, week, month, etc.

## 1.4 Making Fitness Affordable

One misconception that usually occurs is that people think that the only way to get fit is to exercise at the gym. Most of the time, this misconception turns people away from improving their fitness level, due to the monthly membership payment which is normally very costly, especially in the case of using a personal trainer. What most people often do not know is, maintaining the regularity along with a consistent diet is definitely enough to make running the most efficient way to improve fitness level. What we need is a system that can help people analyze their exercise routine and give useful feedbacks to help people reach their fitness goals, so that it can substitute the presence of a trainer. By providing such system, we believe we are able to offer significant benefits to general people, not only by just helping people get fit properly, but also cut down their expenses for such a simple goal of getting healthy. With just one health monitoring device and our application downloaded on their smartphone, we believe they can easily exercise and improve their health without even worrying about not progressing towards their fitness goal.

## 1.5 Ease of Use

When it comes to using a device, people will prefer a system that is simple and does not require too much of manual input. We are aware that a complicated system usually fails to appeal customers and provokes negative user experience, thus removing the software off the competition. Being easy to use is arguably one of the most important factors in making great software, regardless of how many functions the software offers, how sophisticated the system is, or even how beautiful the user interface looks. Obviously, our intention is to make software which is simple and easy to use without sacrificing any key functions, because we want our software to be used by not only the young generation, but everyone.

Our proposed system requires virtually no user manual input. We believe that this is very important to us, as we strive for great accessibility. As we said before, we want our software to be used by everyone, even physically disabled people. Given the fact that our software works in tandem with a health monitoring device, the user is only required to wear the device and set up their fitness goals, for example, their target weight. Our software will actively track, analyze, and provide instantaneous information about user's progress. It will also constantly give useful feedbacks via notification, making it easy for user to receive suggestion to their training regime.

## 2. Glossary of Terms

### **Diet Trackers**

A device that measures the amount of calorie intake and calories burned.

### **Heart Beat**

A number of heartbeats per unit of time, typically expressed as beats per minute(bpm).

### **Symptom**

Subjective evidence of disease or physical disturbance.

### **Chronic Fatigue Syndrome(CFS)**

severe, continued tiredness that is not relieved by rest and is not directly caused by other medical conditions.

### **Personal Trainer**

A fitness professional involved in exercise prescription and instruction. They motivate clients by setting goals and providing feedback and accountability to clients.

### **System**

A set of health monitoring device, application, and user interface working together as a group.

### **User Experience (UX or UE)**

User experience involves a person's emotions about using a particular product, system or service.

## 3. System Requirements Analysis

| 3.1 Functional Requirements Table |                 |   |
|-----------------------------------|-----------------|---|
| ID                                | Priority Weight | Requirement   |
| REQ - 1                           | 5               | System keeps a database for all users and their data  |
| REQ – 2a                          | 5               | System can receive input from health monitoring device  |
| REQ – 2b                          | 5               | System can receive manual input from user   |
| REQ - 3                           | 5               | System generates scores to rate a user's fitness activity   |
| REQ - 4                           | 4               | System can generate graphs to display progress in a time frame  |
| REQ - 5                           | 4               | System allows users to challenge each other, winner is determined by side-by-side comparison of scores earned in specified time frame |
| REQ - 6                           | 3               | System keeps track of user rankings based on fitness scores   |
| REQ - 7                           | 3               | System calculates user's percentile ranking based on fitness scores   |
| REQ - 8                           | 2               | Users can share their fitness scores/graphs to social networking sites  |
| REQ - 9                           | 1               | Users can login   |
| REQ - 10                          | 1               | Users can register an account   |

### 3.2 Non-Functional Requirements Table

| ID       | Priority Weight | Requirement  |
|----------|-----------------|--|
| REQ -11  | 5               | System must keep user data secure                                    |
| REQ - 12 | 4               | System must display data and analysis in clear, easy to read formats |
| REQ - 13 | 4               | System interface must be simple and easy to use                      |
| REQ – 14 | 4               | System must prevent dictionary attacks on user login                 |
| REQ - 15 | 3               | System should be self sufficient and require minimum maintenance     |



### 3.3 On Screen Appearance Requirements

This system consists of two working on-screen appearances that work in the same way but can be used on cross-platforms. The main on-screen appearance requirement is for the website which must abide by the rules and programming languages of html, css, javascript, and php. On the other platform, the android app must meet the requirements of the programming language of java.

**1. Home/Main Page** – gives the user the ability to create an account, view details about the webpage, log into their respective accounts

**2. Refresh** - allow user to refresh page if page is unresponsive or needs to load new information.

**3. Login**

**a. New user** - allows someone to register as a new user. Takes user to the Registration page.

**b. Current members** - allows users login to their data.

**c. Lost password/username** - sends request to system to verify user and create new password or retrieve username account.

**d. Old members** - those who deleted their account could retrieve and restore old accounts

**4. Registration** – Allows new users to register into the database and create an account.

a. Disclaimers, permissions and User Agreements

b. User information registration

**5. Performance tracking** – users' on screen output of their data and how well they are doing compared to the average user.

**6. Account Setting** - add/modify changes to account settings and/or information about user

**7. Help** - FAQ, send feedback, ask for help from administrators

**8. About App** - users can find more info (links) about health, training and tips, also information about how to use the app

## 4. Functional Requirements

### 4.1 Stakeholders

- Athletes
- Professional Sports Leagues
- Fitness Trainers
- Medical Practitioners
- Medical Device Companies

Each of the stakeholders listed above could have their respective fields significantly impacted by the development and implementation of our system. Ultimately our completed system should provide athletes with a way to not only track their workout and fitness progress but also to stay motivated. Athletes and various departments of professional or even amateur sports leagues could utilize our system to track the progress of the athletes participating in league to provide fitness statistics on the athletes. The challenge a friend, and ranking aspects of the system would allow coaches and team managers to make informed decisions on who they should play in what situation by providing them with a clear ranking system that can be used to rank their players in order of best fitness.

In a similar fashion, fitness trainers and medical practitioners can utilize this system as an efficient way of tracking their client or patients fitness activities without having to rely on the patient's word which is not always reliable.

Medical device companies would be directly impacted by our system in a business sense, if our system were to be widely used and implemented, demand for health/fitness monitoring devices would increase. The widespread use of a system such as ours would be of great interest to companies that design, make, sell, and promote the use of health monitoring devices.

## 4.2 Actors and Goals

| Actor   | Goals   |
|---|---|
| <b>User</b> – a registered user of the system | To improve their own fitness and health<br>To track their workout progress<br>To compare their workout stats with friends |
| <b>Database</b>                               | To store user input data<br>To retrieve user input data<br>To facilitate comparison and analysis of user data             |
| <b>Monitoring Device</b>                      | To track and record various aspects of a user's health<br>To allow easy efficient uploading of user's health data         |
| <b>Social Network</b>                         | To provide client with a platform to share data with friends  |
| <b>Visitor</b> – Unregistered User            | To explore system capabilities<br>To view software product information  |

## 4.3 Use Cases

### UC - 1: Visitor Registration (REQ -10)

**Initiating Actor:** Visitor

**Actor's Goal:** To create a client account

**Participating Actors:** Database

**Precondition:** Visitor provides a currently unused username/password combination

**Postcondition:** A new client account with the given username/password is created

**Flow of Events for Main Success Scenario:**

- 1 → User Navigates to registration page
- 2 ← User is prompted to provide username, password, email address
- 3 → User provides username, password, email address
- 4 ← System checks for username availability, and password/email address validity  
    \*\*If invalid password/email or unavailable username → return to 2
- 5 ← System creates and stores new user account to database

### UC – 2: User Login (REQ - 9)

- A user is able to login to their account by providing a valid username and password combination
- A user that provides an invalid username and password combination more than 5 times will be temporarily locked out of the system (Derived From REQ – 16)

### UC - 3: View Data and Ranking (REQ – 1, REQ – 6, REQ – 7, REQ – 3)

- User is able to view previously entered data
- User is able to view current ranking as compared to all other users in database

- User is able to view their percentile ranking compared to all other users in database

#### **UC - 4: Upload Data From Device (REQ – 2a)**

**Initiating Actor:** User

**Actor's Goal:** To store data from recent workout

**Participating Actors:** Database, Monitoring Device

**Precondition:** User is logged in and has data file from the monitoring device available.

**Postcondition:** Data from file is stored in user's account

**Flow of Events for Main Success Scenario:**

- 1 → User navigates to data upload page.
- 2 → User selects data file directory
- 3 → User clicks upload button
- 4 ← System receives data file upload
- 5 ← System stores data from file to database under user's account

#### **UC – 5: Manual Data Input (REQ – 2b)**

- Allows user to input data manually rather than uploading a file from a monitoring device

#### **UC – 6: Challenging a Friend (REQ – 5)**

**Initiating Actor:** User

**Actor's Goal:** To challenge another user to see who has earned a better score in a specified time frame

**Participating Actors:** User2, Database, Social Network

**Precondition:** Users are logged in, and have data uploaded for specified time frame

**Postcondition:** Users' scores are compared side by side and a winner is chosen

**Flow of Events for Main Success Scenario:**

- 1 → User navigates to challenge friend interface
- 2 → User enters username to be challenged
- 3 ← System notifies challenged user
- 4 → Challenged user accepts challenge
- 5 ← System shows users' data side by side
- 6 ← System calculates and displays scores
- 7 ← System chooses a winner based on scores

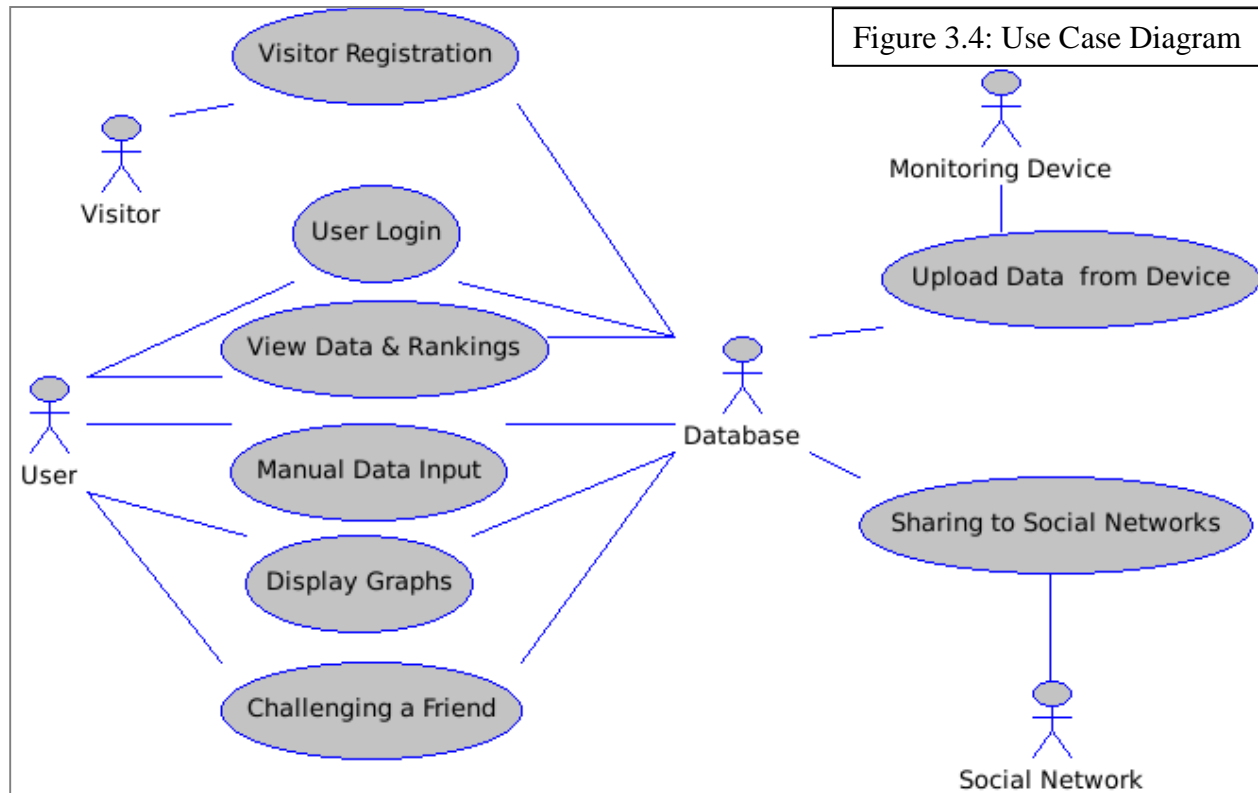
#### **UC – 7: Sharing to Social Networks (REQ – 4, REQ – 8)**

- Allow a client to share fitness data to a linked social networking site
- Allow a client to share generated graphs/tables to a linked social networking site

#### **UC – 8: Display Graphs (REQ – 4)**

- Allows user to view a graph of their workout progress within a specified timeframe

## 4.4 Use Case Diagram



| 3.5 Use Case Traceability Matrix |   |   |   |   |   |   |   |   |   |    |
|----------------------------------|---|---|---|---|---|---|---|---|---|----|
| REQ#                             | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| UC-1                             |   |   |   |   |   |   |   |   |   | X  |
| UC-2                             |   |   |   |   |   |   |   |   | X |    |
| UC-3                             | X |   | X |   |   | X | X |   |   |    |
| UC-4                             |   | X |   |   |   |   |   |   |   |    |
| UC-5                             |   | X |   |   |   |   |   |   |   |    |
| UC-6                             |   |   |   |   | X |   |   |   |   |    |
| UC-7                             |   |   |   | X |   |   |   | X |   |    |
| UC-8                             |   |   |   | X |   |   |   |   |   |    |

## 5. User Effort Estimation

### 5.1 Use Case Effort Estimation

#### UC – 2: User login

Assumption: User is starting at website homepage.

**Navigation:** Data entry fields are on homepage no navigation required

**Data Entry:**

- Click username field
- Enter username (roughly 4- 16 keystrokes)
- TAB → password field
- Enter password (roughly 4 – 16 keystrokes)
- Press “Enter” or “Return” key, or click login button

#### UC – 4: Upload Data from Device

Assumption: User has already performed login procedure

**Navigation:** From home page (4-10 clicks)

- Click “upload data”
- Browse to file location (roughly 1-6 clicks)
- Click “ok”
- Click “upload” or press enter key

#### UC – 6: Challenging a Friend

Assumption: User has already performed login procedure.

**Navigation:** From home page (1 click)

- Click “Challenge a Friend”

**Data Entry:** (1click, 4-16 keystrokes, 1 enter/click)

- Click “username” (to be challenged) field
- Enter friend’s username (roughly 4-16 keystrokes)
- Click “Send Challenge” or press enter key

## 5.2 Use Case Effort Classification

| Use Case | Description  | Category | Weight |
|----------|--|----------|--------|
| UC - 1   | One click, three data entry fields                 | Simple   | 5      |
| UC - 2   | One click, two data entry fields                   | Simple   | 5      |
| UC - 3   | One click  | Simple   | 5      |
| UC - 4   | 4 – 10 clicks, slightly complex interface          | Average  | 10     |
| UC - 5   | 2 clicks 5 data entry fields                       | Average  | 10     |
| UC - 6   | One click, one data entry field                    | Simple   | 5      |
| UC - 7   | 3-6 clicks, two data entry fields slightly complex | Average  | 10     |
| UC - 8   | 2 clicks   | Simple   | 5      |

## 6. Domain Analysis

### 6.1 Domain Concept Definition Table

| Responsibility  | Type | Concept                          | Use Case                            |
|---|------|----------------------------------|-------------------------------------|
| Create new user account                                   | D    | User Account Creator             | UC – 1 Visitor Registration         |
| Verify User login information is correct                  |      | Login Verifier                   | UC – 2 User Login                   |
| Display user's data and rankings                          | D    | User Account Viewer              | UC – 3 View Data and Ranking        |
| Upload data from health monitoring device                 | D    | Device Data Up-Loader            | UC – 4 Upload Data From Device      |
| Receive and store new data input from user                | D    | Account Data Manager             | UC – 5 Manual Data Input            |
| Conduct user challenges                                   | D    | Data Comparator                  | UC – 6 Challenging a Friend         |
| Post client data/graphs/tables to social networks         | D    | Social Network Sharing Interface | UC – 7 Sharing to Social Networks   |
| Generates and Displays graphs based on user inputted data | D    | Graph Generator                  | UC 8 – Display Graphs               |
| Keeps track of user rankings / leader boards              | D    | Rankings Monitor                 | UC – 3 User Views Data and Rankings |

### 6.2 Association Definition Table

| Concept Pair         | Association Description                             | Association Name  |
|----------------------|---|-------------------|
| User Account Creator | User Account Creator utilizes the login verifier to | User Registration |

|   |  |                   |
|---|--|-------------------|
| Login Verifier                                | check if given username/password combination is valid and currently available.                             |                   |
| Account Data Manager<br>Device Data Up-Loader | Data Analyzer and Account Data Manager both allow the system to receive input data from the user           | Data Input        |
| Graph Generator<br>User Account Viewer        | Both concepts allow the user to track their progress by viewing previously entered data in different forms | Progress Tracking |

### 6.3 Attribute Definition Table

| Concept  | Attributes                  | Attribute Description                                       |
|--|-----------------------------|---|
| Account Data Manager<br>Device Data Up-Loader<br>User Account Creator<br>User Account Viewer<br>Login Verifier | Data Access and Storage     | Read/Write from/to database                                 |
| Graph Generator<br>Rankings Monitor  | Data Analysis               | Performs operations on existing data                        |
| Social Network Sharing Interface   | Social Networking Interface | Facilitates interactions between system and social networks |
| Admin Account Manager  | Account Settings Interface  | Utilities for technical services                            |

### 6.4 Domain Traceability Matrix

| Use Case | Domain Concept       |                |                     |                      |                      |                 |                                  |                 |                  |
|----------|----------------------|----------------|---------------------|----------------------|----------------------|-----------------|----------------------------------|-----------------|------------------|
|          | User Account Creator | Login Verifier | User Account Viewer | Device Data Uploader | Account Data Manager | Data Comparator | Social Network Sharing Interface | Graph Generator | Rankings Monitor |
| UC:1     | X                    |                |                     |                      |                      |                 |                                  |                 |                  |
| UC:2     |                      | X              |                     |                      |                      |                 |                                  |                 |                  |
| UC:3     |                      |                | X                   |                      |                      |                 |                                  |                 | X                |
| UC:4     |                      |                |                     | X                    |                      |                 |                                  |                 |                  |
| UC:5     |                      |                |                     |                      | X                    |                 |                                  |                 |                  |
| UC:6     |                      |                |                     |                      |                      | X               |                                  |                 |                  |
| UC:7     |                      |                |                     |                      |                      |                 | X                                |                 |                  |
| UC:8     |                      |                |                     |                      |                      |                 |                                  | X               |                  |



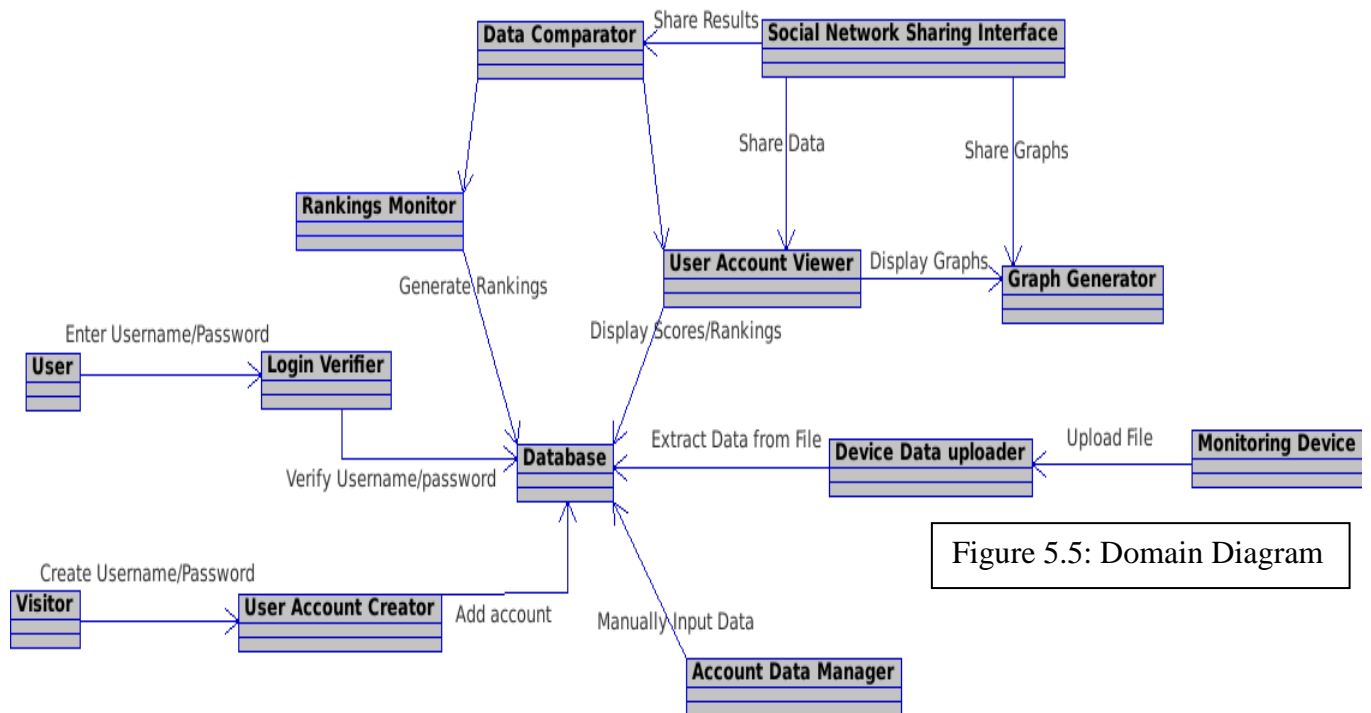


Figure 5.5: Domain Diagram

## 6.5 Mathematical Models

For manually inputting user data, calories burned can be calculated by using the following equations.

Equations for calculating calories burned from heart rate. <sup>1</sup>

Male:  $((-55.0969 + (0.6309 \times \text{HR}) + (0.1988 \times \text{W}) + (0.2017 \times \text{A}))/4.184) \times 60 \times \text{T}$

Female:  $((-20.4022 + (0.4472 \times \text{HR}) - (0.1263 \times \text{W}) + (0.074 \times \text{A}))/4.184) \times 60 \times \text{T}$

where

HR = Heart rate (in beats/minute)

W = Weight (in kilograms)

A = Age (in years)

T = Exercise duration time (in hours)

$$\text{Cardio points} = \left\lceil \frac{\text{calories burned}}{30} \right\rceil$$

The rationale for cardio points equation is a normalization of the score. Since exercise is often used by many to lose weight we decided to use the amount of calories burned for healthy weight loss as a baseline for our scoring system. Typically recommended healthy weight loss is 1-2 pounds per week. A one pound loss in weight requires a 1000 calorie/day deficit and since the average person takes in 2000 calories per day, 3000 calories must be burned to lose one pound per week. Our equation gives a score of 100 for 3000 calories burned per day.

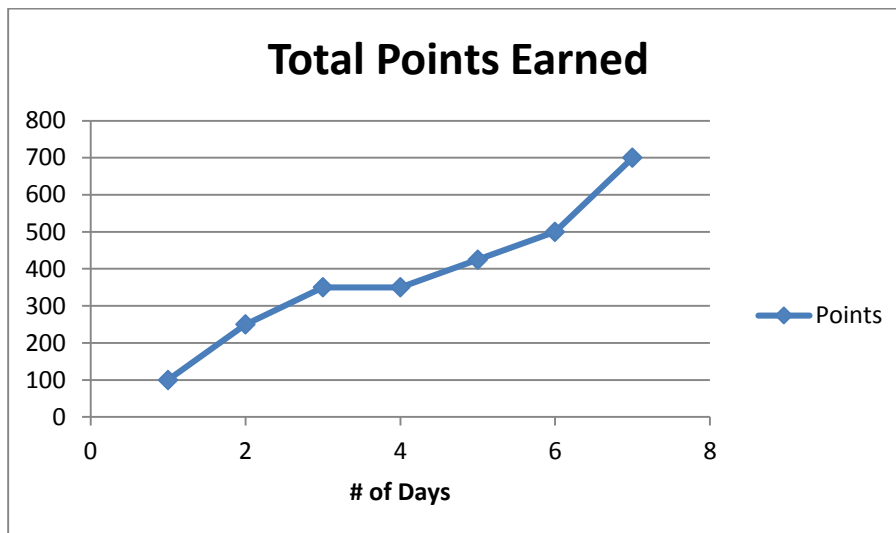
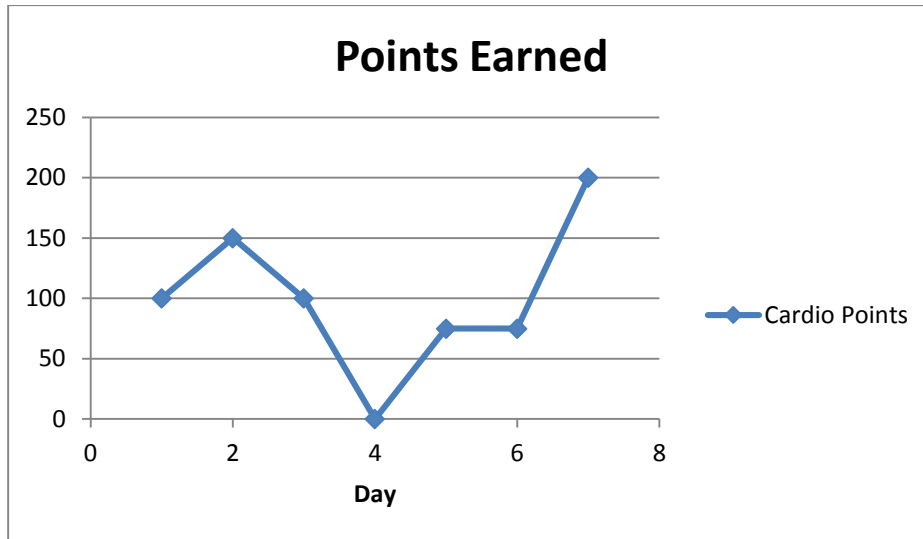
Monthly leader boards will be generated by summing the amount points earned within the month.

Lifetime leader boards will be maintained by taking an average of each users' monthly scores.

$$\text{Percentile Rank} = \frac{\text{Total number of users} - (\text{ranking} - 1)}{\text{Total number of users}} * 100\%$$

Graphical Representations will be displayed as Total Points Earned vs. Time and as Points Earned Per Day vs. Time

Example Plots:



## 6.6 System Operation Contracts

### User Account Creator (UC – 1)

#### Responsibilities

- Create an account for a new user (register a visitor)

#### Preconditions

- User chose a username that is currently not in use
- User chose a valid password
- User provided a valid email address
- User's email address is not currently linked to an existing account

#### Post conditions

- A new account with the user supplied information is created

### Device-Data Uploader (UC – 4)

#### Responsibilities

- Update user account with data from health monitoring device

#### Preconditions

- User has utilized a health monitoring device to generate health data
- User connected health monitoring device to their computer
- User can browse to health monitoring device's data file
- User is logged in

#### Post conditions

- User's account is updated on the database with the data from their health monitoring device's data file

### Data Comparator (UC – 6)

#### Responsibilities

- Display data of challenging and challenged user
- Display scores of challenging and challenged user

- Declare a winner of the challenge

#### Preconditions

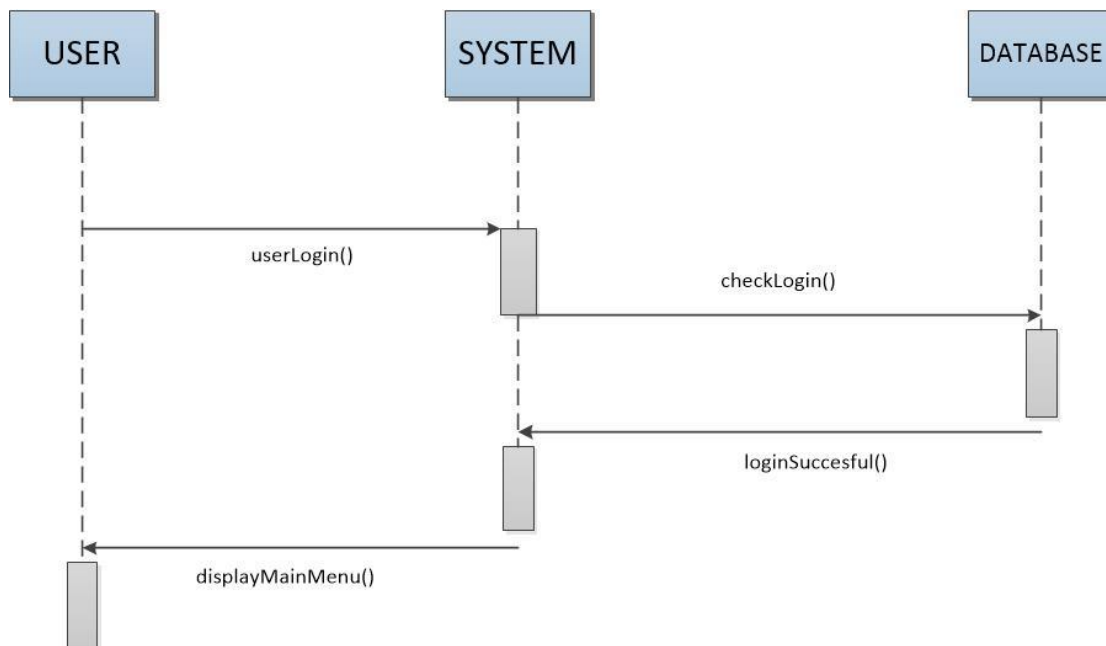
- Users must be logged in
- User (1) challenged user (2)
- User (2) accepted challenge

#### Post conditions

- Side by side comparison of data is displayed on both users' screens
- Side by side comparison of scores are displayed on both users' screens
- Winner and loser are declared on screen

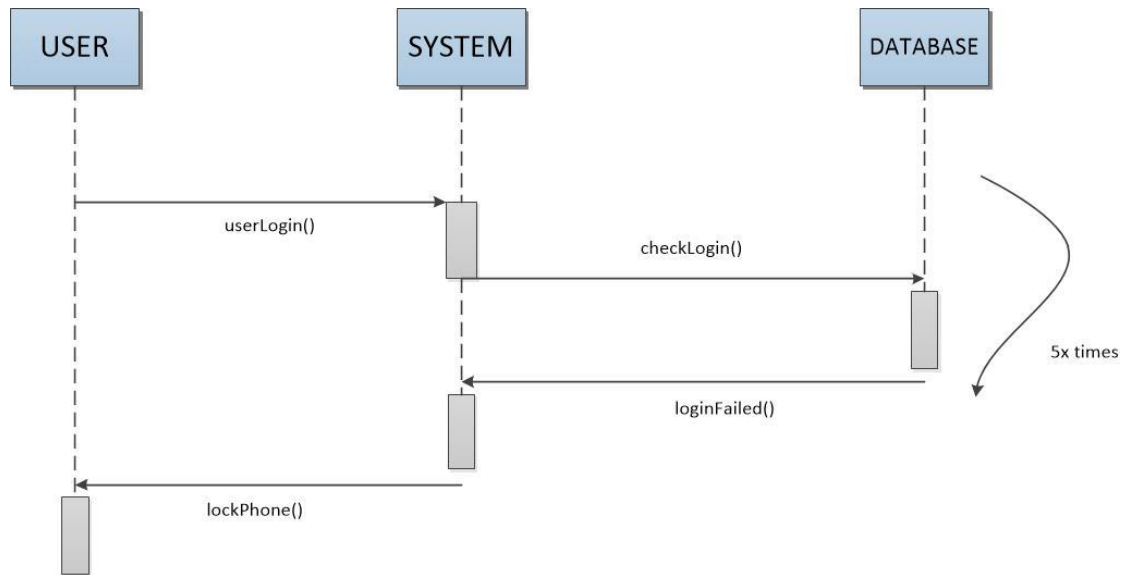
## 7. Interactive Diagrams

### UC – 2: User Login



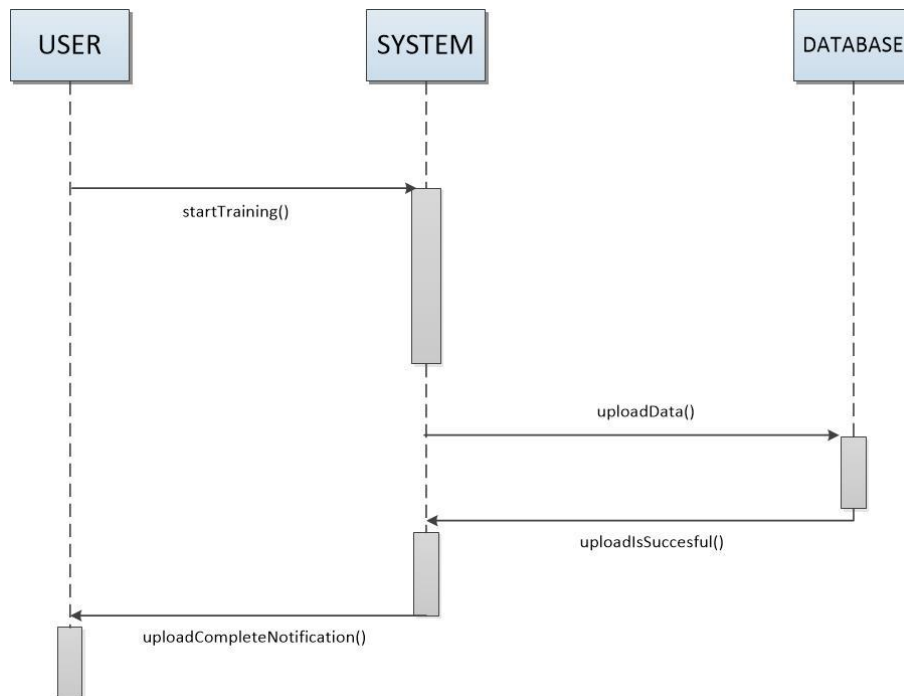
When users login, it must be checked with the database to make sure it is proper and is in the database. Then after it is successful, the user is returned to a main screen for the logged in user.

### Alternate Case



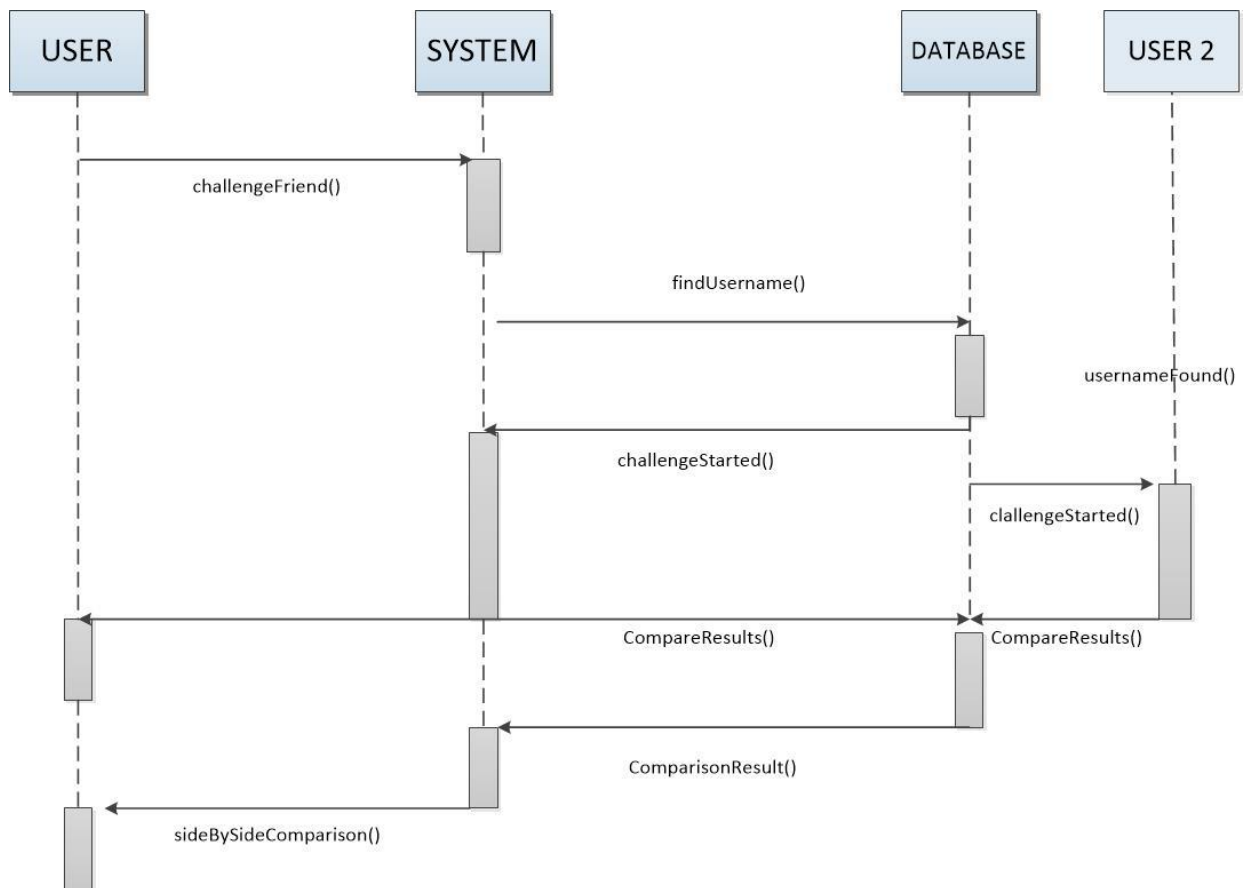
User is asked to log into the system and is then checked with the database for proper login information. The user is allowed 5 times to log in successfully, if not, the screen will return login failure and make user wait a few minutes to login.

### UC – 4: Upload Data from Device



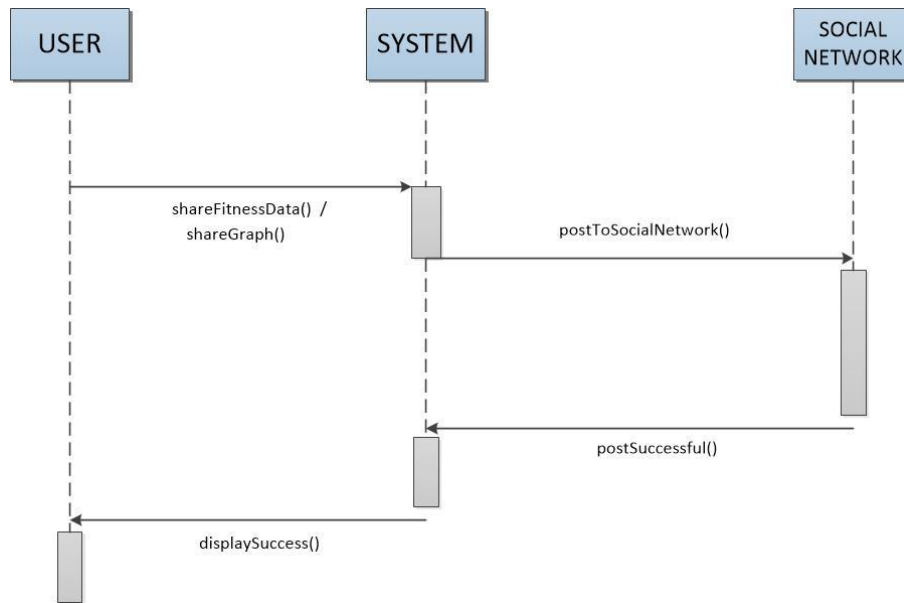
User sends data to the system, the system then uploads the data to database, the database then sends feedback to system then to clients.

## UC – 6: Challenging a Friend



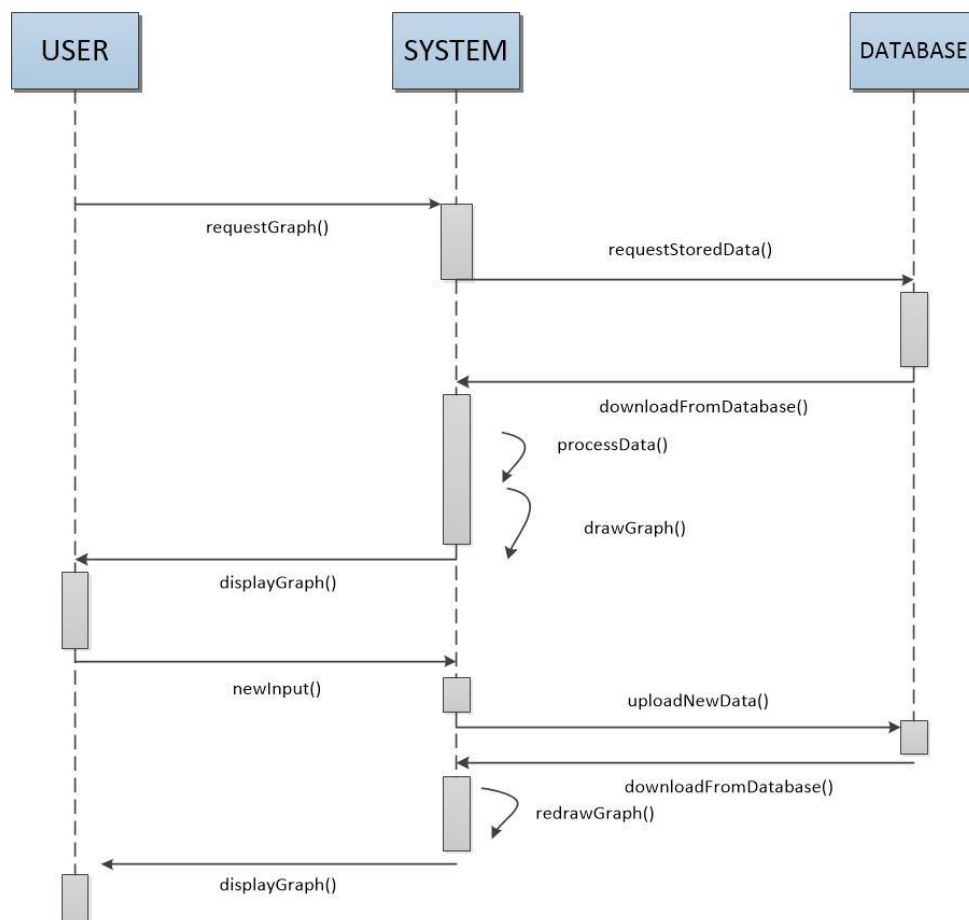
User challenges a friend / another user by typing his/her username, which is then sent to the database for username lookup. If the username does exist, challenge will be started and after some period of time, workout results from both users will be sent to server for comparison. Lastly, the comparison will be sent to the users to display the winner of the challenge.

## UC – 7: Sharing to Social Networks



Clients gives access for the system to share the information about the client’s workout and posts to social medias, when successful, the client will be notified it was successful.

#### UC – 8: Display Graphs (REQ – 4)





Clients request for their data in a graph form, the system then requests database to retrieve the data about the client and process the data and creates a graph. The graph is then displayed to the client side and if the client has new information on their fitness/workout, they can send new/updated information to system and then the system will recreate a new/updated graph and is then displayed to clients.

## 8. Class Diagram and Interface Specification

### 8.1 Class Description Table

| Class Name   | Description   |
|--------------|---|
| Account      | Holds all of user's data and account information such as username/password, and health/fitness data                                 |
| Analyzer     | Contains functions that perform operations on data and generates pertinent statistics and graphical representations                 |
| Database     | Contains a list of all accounts as well as functions for creating a backup and restoring user accounts                              |
| Data Display | Allows the user to view all their previously entered data and generated graphs as well as share them via email or social networking |
| Home Screen  | Provides user to links to the other interfaces as well as allowing them to input new health/fitness data.                           |
| Login Screen | Verifies user login information   |

### 8.2 Class Attribute Tables

| Class: Account        |   |
|-----------------------|---|
| Attribute / Data Type | Description   |
| Username              | String data type used for identifying user                      |
| Password              | String data type used to verify user identity                   |
| Fitness Data          | Various data types used to represent the user's fitness records |
| Email Address         | String data type to store user's email adress                   |

| Class: Analyzer       |  |
|-----------------------|--|
| Attribute / Data Type | Description  |
| analyzeData()         | Performs operations on raw user input to generate refined information for the user       |
| createGraphs()        | Generates graphs depicted the user's raw input and information gained from analyzeData() |

| Class: Database       |  |
|-----------------------|--|
| Attribute / Data Type | Description  |
| Account List          | List of all created instances of class:Account                       |
| createAccount()       | Initializes and stores a new account and adds it to the account list |
| storeData()           | Writes all entered user input data to the database                   |

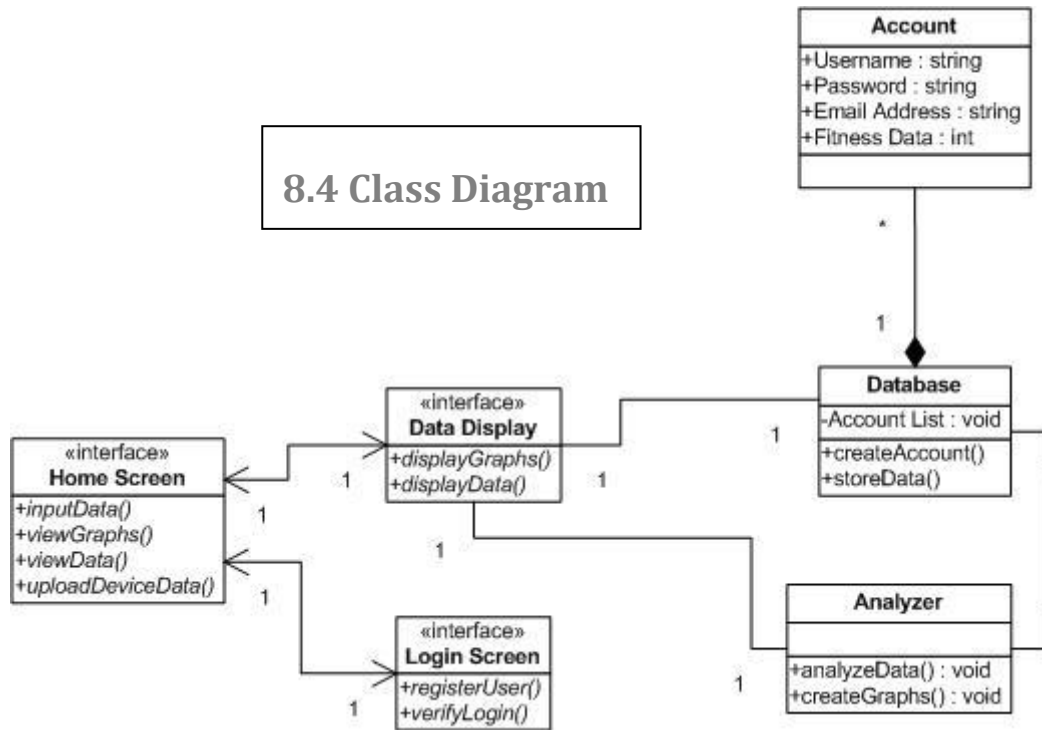
| Class: DataDisplay    |   |
|-----------------------|---|
| Attribute / Data Type | Description   |
| displayGraphs()       | Displays a user's stored/generated graphs to the screen in an organized fashion |
| displayData()         | Displays user's stored raw health and fitness data in an organized display      |

| Interface: Home Screen |   |
|------------------------|---|
| Attribute / Data Type  | Description   |
| inputData()            | Allow to input new raw health/fitness data and sends it to the database     |
| viewGraphs()           | Sends user to Data Display interface and calls the displayGraphs() function |
| viewData()             | Sends user to Data Display interface and calls the displayData() function   |
| uploadDeviceData()     | Reads input data from health monitoring device and sends to database        |

| Interface: Login Screen |  |
|-------------------------|--|
| Attribute / Data Type   | Description  |
| registerUser()          | Sends new user information to the database for creation of new user account    |
| verifyLogin()           | Checks that the username and password combination entered by the user is valid |

| 8.3 Class Traceability Matrix |            |                 |          |          |             |              |
|-------------------------------|------------|-----------------|----------|----------|-------------|--------------|
| Domain Concept                | Class Name |                 |          |          |             |              |
|                               | Account    | Account Setting | Analyzer | Database | Home Screen | Login Screen |
| Create Account                | X          |                 |          | X        | X           | X            |
| Manage Account                | X          | X               |          | X        | X           | X            |
| Insert Data                   | X          | X               |          | X        | X           |              |
| Manage Data                   | X          | X               |          | X        | X           |              |
| Create Graph                  |            |                 | X        | X        |             |              |
| Display Graph                 |            |                 | X        |          | X           |              |
| DisplayData                   |            |                 | X        |          | X           |              |

## 8.4 Class Diagram



## 9. System Architecture and System Design

### a. Architectural Styles

Our system uses the Client/Server style of architecture. This model provides our system with centralized data storage as well as allowing multiple user accounts access and provide data to the database. The client side of the system runs on the client's own machine such as a computer or mobile device. This side handles all the local procedures such as logging in and accessing the database. On the other hand, the server side of the system, which is a MYSQL server, handles the data storage in one place for data analysis and processing, whilst also provides user a secure place for data storage itself. The client/server system also reduces the workload of client machines by having the server do all the data processing. This system also makes it easy to share user workout score with other users, or even challenge another user to a workout competition. Updating the system algorithm, along with other software updates, is also made easier with the Client/Server style. However, a centralized system will generate high amounts of traffic for the server side of the system; thus we will need a reliable server that can handle the workload of the traffic.

### b. Identifying Subsystems

The system is split into a client side subsystem and a server side subsystem. The client side consists of user interface and data input, while the server side

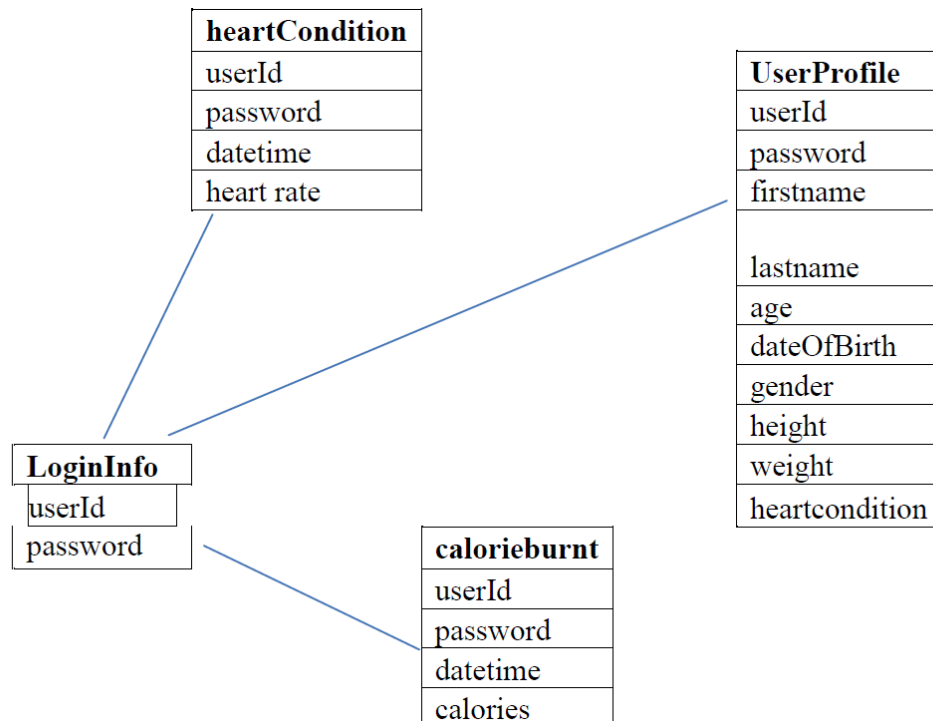
system contains the information and database of each client's data. The client side uses a web browser to access the client side system (html, php), which allows the user to create account, upload data of their progress, and view other users' scores. Same thing goes to the android app that serves as a companion app. It consists of user interface and data processing; it allows the user to create an account, log into their account, and view leaderboard of his/her friends. The server side system takes the data that comes from the client side system and inputs into a database in MySql and stores the data. The server side allows the user to add, manipulate or adjust their information.

### c. Mapping Subsystems to Hardware

There are two needed subsystems, a client side (computer/phone) and a server side. The server will store the database and information of clients while the client side computer will allow the user to access their account, upload their data, and retrieve information they want to look at. The client side could run on multiple computers/phones and the server side can only be accessed by authorized personals.

### d. Persistent Data Storage

The client's data must be stored in a database on a server in MySql. The user's information and health related information will be stored in an array in the database. The following shows the information the array will store of different clients.



### e. Network Protocol

Our system has a central server, which is hosted through Apache HTTP. Requests to access information in the databases, which is MySql, are done through php.

#### **f. Global Control Flow**

Biometric Health Monitoring is an event-driven system. What this means is that, after user creates his/her account, user will be able to do every function implemented in the software in random order. After setting up his/her exercise routine, user can do any implemented function at any given time, such as upload a new data from device, challenge another user, or share their exercise results to connected social network.

#### **g. Hardware Requirements**

The software will require the use of a monitoring device such as Motorola Actv watch. The mobile application will require a mobile device such as a phone or tablet that runs on the Android platform. The mobile application and web service will require an internet connection to access the social media aspect of the software.

## **10. Algorithms and Data Structures**

### **10.1 Algorithms**

The only algorithms implemented in our system's design are basic sorting algorithms used to sort users' scores to generate rankings.

### **10.2 Data structures**

The system utilizes the array data structure in order to store, and sort user data. The use of arrays allows for straight forward and efficient sorting used to generate relative user rankings based on the scores they receive from their workout logs.

# 11. User Interface

## 11.1 Website User Interface

The diagram illustrates a login interface for 'FITNESS FRIENDS'. It consists of the following elements:

- Header:** The text 'FITNESS FRIENDS' is centered at the top.
- Input Fields:** Two rectangular boxes are stacked vertically, labeled 'username box' and 'password box'.
- Login Button:** A wide, rounded rectangular button labeled 'Login Button' is positioned below the input fields.
- Registration Link:** The text 'Create new account' is centered below the login button.
- About Link:** The text 'About' is centered at the bottom of the interface.

Our user interface follows a generic pattern that most website uses for their login purposes, for the sake of simplicity. User types in their username and password in the respective boxes and clicks on login button afterwards. In case user wants to create a new account, he/she can just simply click on the “create new account” text which will bring the user to the registration screen.

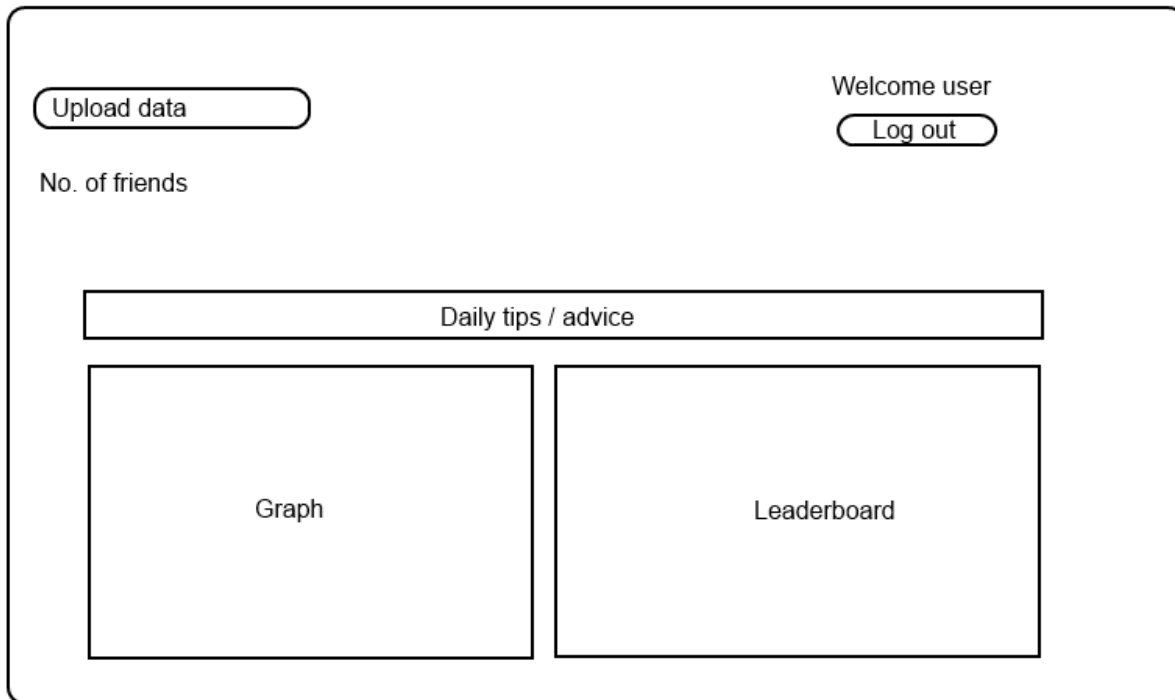
# FITNESS FRIENDS

Welcome! please fill in the information below.

|                 |
|-----------------|
| username box    |
| password box    |
| retype password |
| first name      |
| last name       |
| email address   |
| age             |
| weight (lbs)    |

create account

Filling in the information above and clicking on create account button afterwards will creates a new account with user's own personalized profile.



The main menu of the website consists of several elements. They are the following:

1. Upload data button  
By clicking this button, user will be able to upload their workout result from the monitoring device. The precondition of this is monitoring device is connected to the computer, or the workout data has been uploaded to computer.
2. Log out button  
Clicking this button will bring the display back to login screen.
3. Daily tips/advice  
This element shows a daily tips that will help user work on their workout routine. It displays a new advice every day.
4. Graph  
This element shows calories burnt on a single day over some period of time. This gives the user a clear, graphical representation of how hard she/he has been working out.
5. Leaderboard  
This element sorts scores from user and his/her friends and puts it onto a ranking.

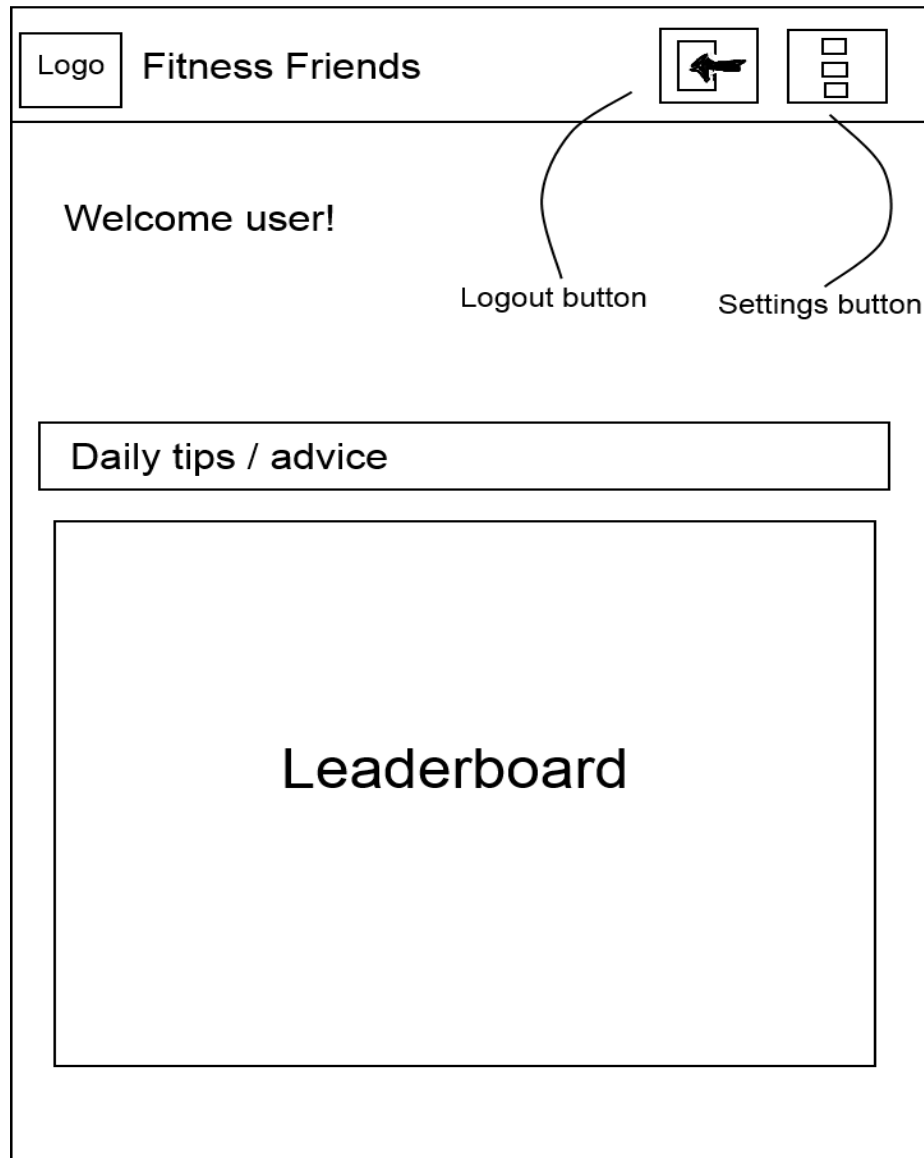


## 11.2 Android App User Interface

The diagram illustrates the user interface for the 'Fitness Friends' app. It features a central login form with the following elements:

- Title:** 'Fitness Friends' is displayed prominently at the top of the form area.
- Input Fields:** There are two input fields, one labeled 'Username box' and one labeled 'Password box', stacked vertically.
- Action Buttons:** A large 'Login' button is positioned below the password field.
- Link:** A text link 'Haven't created an account yet?' is located below the password field, likely leading to a registration screen.
- Footer:** An 'About' button is located at the bottom center of the main container.

For our android application, our user interface also strives for simplicity, by following the generic pattern provided by other applications on Google Play. Clicking on “Haven’t created an account yet?” will inflate a registration layout.



As the android application only serves as a companion app, user is unable to upload their workout data straight from the application. There are a few elements on the main menu screen, which are the following:

- Log out button  
This will log user out from the app and bring window back to the login screen.
- Settings button  
Clicking on this will allow user to edit and change their preferences/profile.
- Daily tips/advice  
This element shows tips/advice that will help user work on their workout and training regime. It shows a different advice on every different day.
- Leaderboard

This element sorts scores from user and his/her friends and puts it onto a ranking

## 12. Unit Test Cases

The purpose of the unit test cases is to test individual parts of the system's code. By breaking the overall code into smaller parts to test it is easier to more accurately and thoroughly test each part of the code, and if a problem is observed the smaller the tested parts are the easier it will be to find the source of the problem in the code.

### Test ID: TC1\_Register User

Assumption: User is at Login Screen.

| Input Requirement                  | Expected Output | Pass/Fail   | Comments  |
|------------------------------------|-----------------|---|---|
| Valid Registration Info            | Account Created | Pass if a new account with the user's info is created and saved to the database   |   |
| Incomplete User Entries            | Error Message   | Pass if error message is received and user is prompted to re-enter data           |   |
| Account information already in use | Error Message   | Pass if error message is received and user is alerted that account already exists | Can be triggered by either having a duplicate username, or provided email address is already linked to another username |

### Test ID: TC2\_Login

Assumption: User is at Login Screen.

| Input Requirement                     | Expected Output  | Pass/Fail   | Comments  |
|---------------------------------------|------------------|---|---|
| Valid Username/Password Combination   | Login Successful | Pass if user is directed to Home Screen   |   |
| Invalid Username/Password Combination | Login Failure    | Pass if user remains at login screen and receives invalid username/password message | Repeated failures should result in account being temporarily locked |

### Test ID: TC3\_Upload Data File

Assumption: User is at Home Screen.

| Input Requirement | Expected Output | Pass/Fail  | Comments   |
|-------------------|-----------------|--|--|
| Device Data File  | Data Updated    | Pass if user's workout data is correctly updated to reflect the data in the uploaded workout file. | Testing will have to be repeated with various types and quantities of data |

### Test ID: TC4\_Display Data

Assumption: User is at Account Settings Screen and has previously entered/uploaded data to the system.

| Input Requirement   | Expected Output                          | Pass/Fail   | Comments |
|---------------------|--|---|----------|
| Uploaded Data files | Graph of user's data for specified month | Pass if output graph correctly represents the user's data |          |

### Test ID: TC5\_Compare Data to Friend

Assumption: All users already have data uploaded to their accounts

| Input Requirement          | Expected Output                       | Pass/Fail   | Comments |
|----------------------------|---------------------------------------|---|----------|
| Valid username to search   | User is directed to comparison screen | Pass if comparison screen correctly displays each user's data and correctly indicates who has more points |          |
| Invalid username to search | Error message                         | Pass if error message indicating an invalid username is recieved  |          |

### Test ID: TC6\_Leader Boards

Assumption: All users already have data uploaded to their accounts

| Input Requirement               | Expected Output                        | Pass/Fail   | Comments |
|---------------------------------|--|---|----------|
| User navigates to leader boards | Leader boards ordered by users' scores | Users are correctly listed in order based on the amount of points their account has received. |          |

## Test Coverage

The test cases listed above cover the most important use cases and the use cases that are most likely to be problematic. Our main focus is to ensure that the data is uploaded and displayed properly and the leader boards and friend comparisons are operational.

## Integration Testing

We've decided to implement a "Big Bang" integration testing strategy. Due to the design of our system and out time constraints the Big Bang strategy officers the most effective form of integration testing for our system. This strategy consists of us putting together the entire system and running through each individual test case on the completed system to ensure that the system performs as expected.

## 13. History of Work, Current Status, and Future Work

### 13.1 History of Work

We created a table to show our project milestones along with its respective completion date.

|                                    |                           |
|------------------------------------|---------------------------|
| First report                       | February 22 <sup>nd</sup> |
| Second report                      | March 15 <sup>th</sup>    |
| Start of app & website development | March 22 <sup>nd</sup>    |
| Website launch                     | March 30 <sup>th</sup>    |
| Demo 1                             | April 2 <sup>nd</sup>     |
| First report revision              | April 29 <sup>th</sup>    |
| Full report                        | May 5 <sup>th</sup>       |

### 13.2 Current Status

Initially we wanted to create a health monitoring program that serves as a virtual assistant for users to work out properly, with an optional professional review if the user chooses to have such review. But after the feedback we received at first demo we found out that the product was not appealing enough so we decided to change the core idea of our project. We decided to make a program that not only serves as a health monitoring program, but also takes the workout results to one estimated score that represents how hard they have been working out.

We currently do not have nor plan to implement many functions to our product, taking everything into account. We are very undermanned and from the beginning of the project it has been virtually only three of us putting effort onto the project.

Right now what we have is a working client-server connection and data processing. User can log into/create their account, and upload their workout results from main menu. We are still trying to work on our scoring, leaderboard system, and also calories-burnt graph.

### 13.3 Future Work

There are a number of features that we would have liked to implement in our system's design but were unable due to our constraints on time and man power. Below is a list of the most significant features that we believe could add a great amount of functionality to our system.

**Personalized Tips:** Currently our system displays randomized “Daily Tips” regarding general workout and fitness knowledge. These daily tips could possibly be upgraded to be personalized based on an analysis of the user’s data in order to be more effective.

**Automatic Uploading:** A great feature to add would be to enable the user to instantly and wirelessly upload their fitness/workout data from their health monitoring device to our system’s database. This gives the user instant gratification for their workout by allowing them to immediately receive points for their workout and does not require them to remember to take the time to upload the data later. Two approaches to this feature would be to sync the monitoring device to the user’s smart phone, or if possible turn the user’s smart phone into a monitoring device by utilizing the phone’s gps to track their workouts (at least those that can be tracked via gps).

**Diet Journal:** There are already many dietary journal applications available but it could be a good idea to implement one alongside the fitness and workout data. This way, users can keep track of their diet and workout habits side by side and can allow the system to perform more detailed health analysis such as calorie intake vs. calories burned.

**Professional Review:** This is a feature that we originally wanted to implement but decided it would require too much time to implement and therefore was not included in the scope of our final demo. The professional review feature would allow for system users designated as “professionals” to receive permission from a user to view their dietary, health, and fitness data and make suggestions as to how they can improve their overall health. Such professionals could be a coach, dietician, nutritionist, doctor, personal trainer, etc.

**Friends/Groups Leader Boards:** If given more time we would have liked to create a feature where users can have a friends list, and/or be part of various groups of users and be able to compare their data directly to their list of friends or to the group and generate leader boards specific to that list or group.

**Strength Training and Other Workouts:** Our system is limited to mainly handling cardio type exercises that can be easily handled by boiling them down to the number of calories burned off by the workout. A more comprehensive system could be developed in order to handle other types of workouts such as strength training or weight lifting. This system would have to be more complex since there are many different workouts aimed at specific muscle groups, and when comparing users age and body type (weight/height) should factor into a score generated since users with more body mass should be able to perform weightlifting workouts with more weight.

## 14. References

- 2) **"Adult Obesity Facts."** *Cdc.gov*. Centers for Disease Control and Prevention, 13 Aug. 2012. Web. 08 Oct. 2012. <<http://www.cdc.gov/obesity/data/adult.html>>.
- 3) **"Google Code University: Android."** *Code.google.com*. Google, n.d. Web. 23 February 2013. <<http://code.google.com/edu/android/index.html>>.
- 4) **"Heart Diseases & Disorders."** *Hrsonline.org*. Heart Rhythm Society, n.d. Web. 23 Feb. 2013. <<http://www.hrsonline.org/Patient-Resources/Heart-Diseases-Disorders>>.
- 5) **"Heart Rate."** *Wikipedia*. Wikimedia Foundation, 10 Aug. 2012. Web. 23 Feb. 2013. <[http://en.wikipedia.org/wiki/Heart\\_rate](http://en.wikipedia.org/wiki/Heart_rate)>.
- 6) McGee, Marianne Kolbasuk. **"11 Telemedicine Tools Transforming Healthcare."** *Informationweek*. InformationWeek, 23 Mar. 2012. Web. 23 Feb. 2013. <<http://www.informationweek.com/healthcare/mobile-wireless/11-telemedicine-tools-transforminghealth/232602982>>.
- 7) **"Wearable Wireless Medical Devices to Top 100 Million Units Annually by 2016, ABI Research."** *Wearable Wireless Medical Devices to Top 100 Million Units Annually by 2016, ABI Research*. Business Wire, 17 Aug. 2011. Web. 08 Oct. 2012. <<http://www.businesswire.com/news/home/20110817006223/en/Wearable-Wireless-Medical-Devices-Top-100-Million>>.

### Math Equation Reference

<sup>1</sup> Calories burned from heart rate: <http://www.shapesense.com/fitness-exercise/calculators/heart-rate-based-calorie-burn-calculator.aspx>

Equations Derived by LR Keytel, JH Goedecke, TD Noakes, H Hiiloskorpi, R Laukkanen, L van der Merwe, and EV Lambert

# Summary of Changes

The purpose of this section is to clearly state the changes made in the project.

## Functional Requirements

The following Functional requirements have been removed in order to tighten the scope of the project. Our original project was too ambitious especially now that our group consists of only three members, some requirements were cut from the project in order to focus on the more significant ones.

|                  |          |   |
|------------------|----------|---|
| <b>REQ – 2a</b>  | <b>5</b> | <b>System can access, and compare user data</b>                         |
| <b>REQ – 2b</b>  | <b>5</b> | <b>System can retrieve, and display user data</b>                       |
| <b>REQ – 2c</b>  | <b>5</b> | <b>System can analyze and modify user data</b>                          |
| <b>REQ – 2d</b>  | <b>5</b> | <b>System can receive input and store new data from user</b>            |
| <b>REQ – 3a</b>  | 5        | System can calculate calories burned from user data                     |
| <b>REQ – 3b</b>  | 4        | System can generate graphs and tables from user data                    |
| <b>REQ – 3c</b>  | 4        | System can monitor calorie intake v.s. calories burned                  |
| <b>REQ - 7</b>   | 3        | User is allowed to change login information and settings                |
| <b>REQ - 8</b>   | 2        | Administrator can access user data and account information              |
| <b>REQ - 9</b>   | 2        | Administrator can delete or restrict user accounts                      |
| <b>REQ - 10</b>  | 2        | User can share data, graphs and tables to Social Networks               |
| <b>REQ – 11a</b> | 1        | User can grant data access to other users                               |
| <b>REQ – 11b</b> | 1        | User can request data review from users labeled as health professionals |

The above requirements were replaced with the following

|                 |          |   |
|-----------------|----------|---|
| <b>REQ – 2a</b> | <b>5</b> | <b>System can receive input from health monitoring device</b>   |
| <b>REQ – 2b</b> | <b>5</b> | <b>System can receive manual input from user</b>  |
| <b>REQ - 3</b>  | 5        | System generates scores to rate a user's fitness activity   |
| <b>REQ - 4</b>  | 4        | System can generate graphs to display progress in a time frame  |
| <b>REQ - 5</b>  | 4        | System allows users to challenge each other, winner is determined by side-by-side comparison of scores earned in specified time frame |
| <b>REQ - 6</b>  | 3        | System keeps track of user rankings based on fitness scores   |
| <b>REQ - 7</b>  | 3        | System calculates user's percentile ranking based on fitness scores   |

This change in functional requirements allows us to focus more on the competitive aspect of our system that is designed to keep users entertained and motivated to use the system. The focus of the update project (as can be seen by the functional requirements) is the generation of



scores/rankings based on a user's workouts and a challenge system that allows a user to go head to head with their friends.

## **Use Cases**

Our project's use cases have been updated to reflect the changes in the functional requirements. Use cases involving the old professional review concept, as well as some administrative use cases such as banning/restricting accounts etc. have been removed and replaced or updated by use cases such as **Challenging a Friend**.

## **User Interface**

The user interface section has been overhauled to reflect the changes we made and to more clearly demonstrate the user interface being implemented in our design.

## **Domain Analysis**

The rest of the report has been updated by adding/removing/editing concepts in order to reflect the changes made to our functional requirements and use cases.

## **Test Cases**

The following test designs were removed from the report because the functionality that they were designed to test is no longer included in the scope of the project.

**Test ID: TC7\_Database Backup and Restore**

**Test ID: TC4\_Social Networking (SN)**

The following test cases were changed/edited as indicated in order to more accurately test our systems functionality within the project's scope.

**TC3\_Input Data and Analysis + TC8\_MonitoringDevice → TC3\_Upload Data File**

These two cases were combined and edited into a single test case that tests the system's ability to receive a .csv data file.

# Contribution Breakdown

| Contribution Breakdown Report #3        |      |         |     |
|---|------|---------|-----|
| Task                                    | Cody | Florian | Jie |
| Summary of Changes                      | 100  | 0       | 0   |
| Customer Statement of Requirements      | 70   | 30      | 0   |
| Glossary of Terms                       | 0    | 100     | 0   |
| System Requirements                     | 50   | 30      | 20  |
| Functional Requirements Specification   | 100  | 0       | 0   |
| Effort Estimation                       | 100  | 0       | 0   |
| Domain Analysis                         | 100  | 0       | 0   |
| Interaction Diagrams                    | 0    | 100     | 0   |
| Class Diagram & Interface Specification | 100  | 0       | 0   |
| System Architecture & System Design     | 0    | 50      | 50  |
| Algorithms & Data Structures            | 100  | 0       | 0   |
| User Interface Design & Implementation  | 0    | 50      | 50  |
| Design of Tests                         | 90   | 10      | 0   |
| History of Work                         | 50   | 50      | 0   |
| References                              | 10   | 0       | 90  |
| Project Management                      | 50   | 0       | 50  |