# 

# 

# 

# 

# 

# 

# CSC 431

# 

# **UM gym machines and waiting times**

# 

# 

# **Software Requirements Specification (SRS)**

**Team 10**

| Ross Stewart | Developer |
| --- | --- |
| James Johnson | Developer |

# **Version History**

| **Version** | **Date** | **Author(s)** | **Change Comments** |
| --- | --- | --- | --- |
| **1.0** | 2/23/23 | Ross Stewart,  James Johnson | First Draft |
| **1.1** | 5/5/23 | Ross Stewart,  James Johnson | Final Draft |
|  |  |  |  |
|  |  |  |  |

# 

# 

# 

# 

# 

# 

# 

# 

# **Table of Contents**

1. System Requirements

1.1 Functional Requirements

1.1.1 Sign Up

1.1.2 Log In

1.1.3 Create User Profile

1.1.4 View Home Page

1.1.5 View Exercises By Muscle Group

1.1.6 View Individual Exercise

1.1.7 View Favorite Exercises

1.1.8 View Gym Crowd Level

1.1.9 Log Out

1.2 Non-Functional Requirements

1.2.1 Efficient Exercise Sorting

1.2.2 Useful Notifications

1.2.3 User Reported Data Analytics

1.2.4 Network Security

1.2.5 User Data Privacy/Security

1.2.6 Efficient API Interactions

2. System Constraints

2.1 Tool Constraints

2.1.1 Mobile Application Framework Constraint

2.1.2 Amazon Web Services

2.2 Language Constraints

2.2.1 React Framework

2.2.2 Database Interactions Language

2.3 Platform Constraints

2.3.1 Mobile Application Platforms

2.4 Hardware Constraints

2.4.1 Mobile Devices

2.5 Network Constraints

2.5.1 Internet Connection

2.6 Deployment Constraints

2.6.1 Deployment to Respective App Stores

2.7 Transition & Support Constraints

2.7.1 End of Project

2.8 Budget & Schedule Constraints

2.8.1 Semester Timeline

3. Requirements Modeling

3.1 User and App

4. Evolutionary Requirements

4.1 Functional Requirements

4.1.1 View App Analytics

4.1.2 Update Exercise Data

4.2 Non-Functional Requirements

4.2.1 Maintain Accurate Data

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **Table of Figures**

3. Requirements Modeling

3.1.1 User and App

### 

### 

### 

### 

### 

### 

### 

### **1.** **System Requirements**

#### **1.1 Functional Requirements**

#### 1.1.1 Sign up

| Title | Signup |
| --- | --- |
| Description | Create user account to store user data for the app |
| Priority | 3 |
| Precondition(s) | The user must have an email, phone number, or Google account. |
| Basic Flow | * User opens application * User is prompted to sign up or log in * First time user click sign up * User is redirected to third party to enter information for account |
| Postconditions(s) | User is prompted to save login info and the user has an account in the system, the user profile is then created. |
| Use Case Diagram | 3.1 |

##### 

##### 

##### 

##### 

##### 1.1.2 Log In

| Title | User logs into account |
| --- | --- |
| Description | The user enters login info to access application |
| Priority | 3 |
| Precondition(s) | The user must have a working account |
| Basic Flow | * User opens application * User is prompted to sign up or log in * User clicks log in * User is redirected to third party to enter login information |
| Postconditions(s) | The user is directed to the home page. |
| Use Case Diagram | 3.1 |

##### 1.1.3 Create User Profile

| Title | Create user profile |
| --- | --- |
| Description | The user makes an account through the sign up |
| Priority | 3 |
| Precondition(s) | The user signs up |
| Basic Flow | * The user signs up * An account is made for the user * They are initialized to have no favorite exercises |
| Postconditions(s) | Redirects user to login page |
| Use Case Diagram | 3.1 |

##### 

##### 1.1.4 View Home Page

| Title | View home page |
| --- | --- |
| Description | The central page of the app, the user can view all of the functionalities of the home page |
| Priority | 0 |
| Precondition(s) | The user needs to log in |
| Basic Flow | * The user logs in and is directed to the home page * The user can view exercises by muscle group, view gym crowdedness, and view favorite exercises |
| Postconditions(s) | The user is able to use the different options at the home page |
| Use Case Diagram | 3.1 |

##### 1.1.5 View Exercises By Muscle Group

| Title | View exercises by muscle group |
| --- | --- |
| Description | The user is can select one of various muscle groups, and can view an alphabetically sorted list of exercises that target the selected muscle group |
| Priority | 0 |
| Precondition(s) | The user is on the home page |
| Basic Flow | * User clicks on target muscle group * The user is presented with a selection of exercises for that muscle group * The user can click on an exercise to view it in detail |
| Postconditions(s) | The user can view various exercises for each muscle group, and can view each individual exercise individually |
| Use Case Diagram | 3.1 |

##### 1.1.6 View Individual Exercises

| Title | View individual exercise |
| --- | --- |
| Description | The user is can select an exercise inside of a muscle group and is presented with information on that exercise as well as the option to add the exercise to favorites |
| Priority | 0 |
| Precondition(s) | User views exercises by a muscle group  OR User views exercises by favorites |
| Basic Flow | * User selects an exercise * User can see the exercise name, description/instructions, the expected wait time, as well as a link to an instructional video * Expected wait time will be calculated using gym crowdedness and the exercise popularity constant * User can also favorite the exercise |
| Postconditions(s) | User can read exercise info as well as can be redirected to an instructional video on the internet, and can have an exercise favorited |
| Use Case Diagram | 3.1 |

##### 

##### 

##### 

##### 

##### 1.1.7 View Favorite Exercises

| Title | View exercises by favorites |
| --- | --- |
| Description | User can view an alphabetically-sorted list of their favorite exercises |
| Priority | 1 |
| Precondition(s) | User is on the home page |
| Basic Flow | * User selects “View Favorite Exercises” * A list of their favorite exercises is presented to them, and the user can select exercises individually * If the user has no favorite exercises, instead of a list it will display “You have no favorite exercises” * If the user unfavorites an exercise, it will be removed from the list and the new list will be displayed instead upon return to the page |
| Postconditions(s) | The user can view various exercises for each muscle group, and can view each individual exercise individually |
| Use Case Diagram | 3.1 |

##### 

##### 

##### 

##### 

##### 

##### 1.1.8 View Gym Crowd Level

| Title | View gym crowdedness |
| --- | --- |
| Description | The user can view the crowdedness of the gym |
| Priority | 2 |
| Precondition(s) | User is on the home page |
| Basic Flow | * User is on the home page * The gym crowdedness is loaded upon app launch and is displayed on the home page * Gym crowdedness is taken from the gym counter in the gym |
| Postconditions(s) | User can view the gym crowdedness |
| Use Case Diagram | 3.1 |

##### 1.1.9 Log Out

| Title | Logging out |
| --- | --- |
| Description | The user logs out of their account. |
| Priority | 3 |
| Precondition(s) | The user is logged in |
| Basic Flow | * The user clicks the settings icon * The user clicks the logout button in the window * The user will also be logged out if the application is closed |
| Postconditions(s) | The user is logged out of their account and is brought to the and is prompted to sign up or log in. |
| Use Case Diagram | 3.1 |

**1.2 Non-Functional Requirements**

#### 1.2.1 Efficient Exercise Sorting

| Title | Efficient sorting of exercises |
| --- | --- |
| Description | Efficient sorting when the exercises are sorted alphabetically |
| Priority | 3 |
| Applicable FR(s) | 1.1.5, 1.1.7 |

1.2.2 Useful Notifications

| Title | Push notifications for the user |
| --- | --- |
| Description | The user will get a notification from the app when the gym is not crowded, or less crowded than usual |
| Priority | 4 |
| Applicable FR(s) | 1.1.8 |

##### 1.2.3 User Reported Data Analytics

| Title | User reported data analytics |
| --- | --- |
| Description | The users will report the popularity of each exercise, so the application can calculate the popularity constant of each exercise |
| Priority | 4 |
| Applicable FR(s) | 1.1.6 |

##### 

##### 1.2.4 Network Security

| Title | Network security |
| --- | --- |
| Description | The network connection for the application will be secure and protected |
| Priority | 2 |
| Applicable FR(s) | 1.1.1 - 1.1.9 |

##### 1.2.5 User Data Privacy / Security

| Title | Privacy / Security |
| --- | --- |
| Description | User data and login info will be stored protected through encryption |
| Priority | 2 |
| Applicable FR(s) | 1.1.1, 1.1.2, 1.1.3 |

##### 1.2.6 Efficient API Interactions

| Title | Efficient API interactions |
| --- | --- |
| Description | API interaction (third-party login) should not slow the application down by more than a second |
| Priority | 2 |
| Applicable FR(s) | 1.1.1, 1.1.2, 1.1.3 |

### 

### **2.** **System Constraints**

### **2.1** **Tool Constraints**

### 2.1.1 Mobile Application Framework Constraint

| Title | React |
| --- | --- |
| Description | React will be used to deploy both the backend and the UI |
| Priority | 0 |

##### 2.1.2 Amazon Web Services

| Title | AWS Cloud |
| --- | --- |
| Description | To maintain scalability and secure storage AWS will be essential |
| Priority | 0 |

#### 

#### **2.2** **Language Constraints**

##### 2.2.1 React Framework

| Title | React |
| --- | --- |
| Description | A Javascript framework used to develop apps for several different mobile platforms. This will allow for efficient development and deployment on multiple platforms |
| Priority | 0 |

2.2.2 Database Interactions Language

| Title | SQL |
| --- | --- |
| Description | To store and interact with user data and our own processes, a language is needed to communicate with relational databases we construct |
| Priority | 0 |

#### **2.3** **Platform Constraints**

#### 2.3.1 Mobile Application Platforms

| Title | Android and IOS. |
| --- | --- |
| Description | React will help us develop the app cross platform, and we will release the app on several common platforms. |
| Priority | 0 |

#### 

#### **2.4** **Hardware Constraints**

#### 2.4.1 Mobile Devices

| Title | Mobile Phones |
| --- | --- |
| Description | Hardware features of popular mobile devices will limit what can and cannot be done by our app. |
| Priority | 0 |

**2.5** **Network Constraints**

#### 2.5.1 Internet Connection

| Title | Internet |
| --- | --- |
| Description | The app is centered around the internet, without being able to update the data live, the app is at half functionality. |
| Priority | 0 |

#### 

#### 

#### 

#### **2.6** **Deployment Constraints**

#### 2.6.1 Deployment to Respective App Stores

| Title | App Store Deployment |
| --- | --- |
| Description | Every app store and platform will need to separately approve our app for release. Rules and regulations for each must be considered |
| Priority | 4 |

#### **2.7** **Transition & Support Constraints**

2.7.1 End of Project

| Title | Project Ends |
| --- | --- |
| Description | When the class and project ends this semester another group will have to take over this project. |
| Priority | 4 |

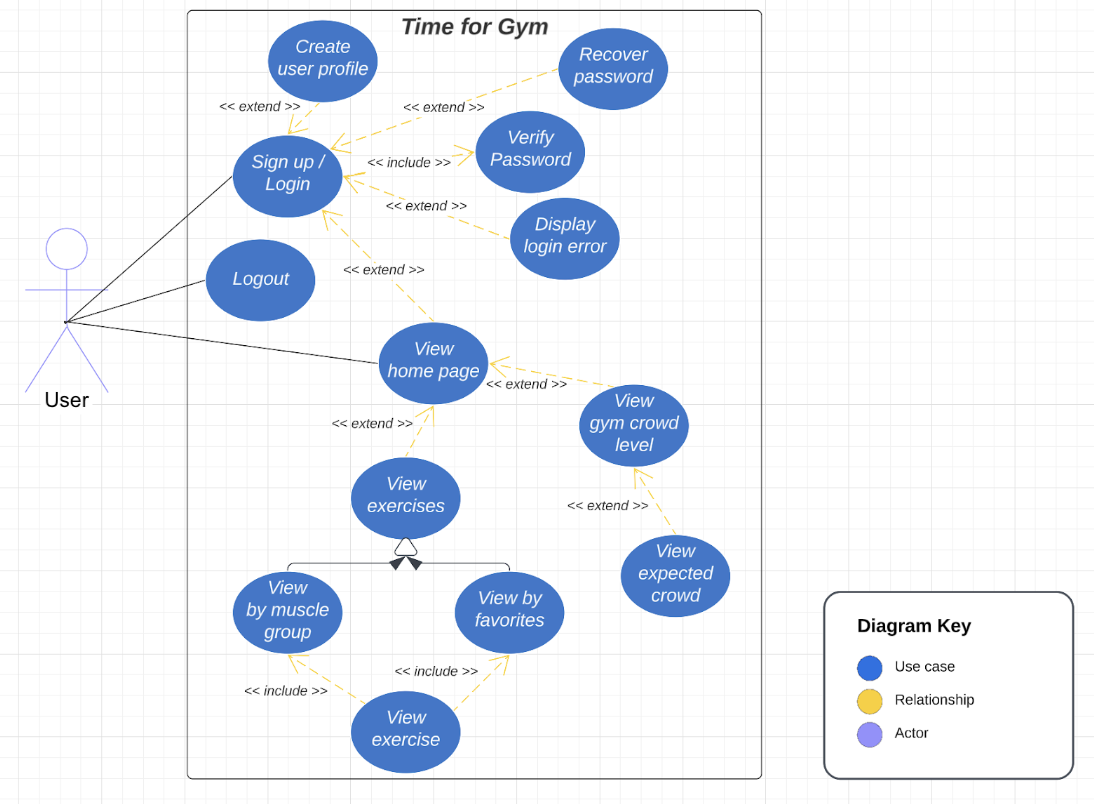
**2.8** **Budget & Schedule Constraints**

#### 2.8.1 Semester Timeline

| Title | End of Semester |
| --- | --- |
| Description | When the semester ends so must this project, due to the lack of budget |
| Priority | 4 |

### **3.** **Requirements Modeling**

### 3.1 User and App



### 

### 

### **4.** **Evolutionary Requirements**

### **4.1** **Functional Requirements**

##### 4.1.1 View App Analytics

| Title | Analytics |
| --- | --- |
| Description | To increase the efficiency of the app and usability of the app, data will be collected on how the app is used. Over time trends will emerge to allow us to update the app to better serve the users. |
| Priority | 4 |
| Precondition(s) | Data needs to be collected |
| Postconditions(s) | We can analyze the data to increase the accuracy of the app |

4.1.2 Update Exercise Data

| Title | Update Data |
| --- | --- |
| Description | To keep this app modern and updated in the world of fitness we need to stay updated with the newest machines and exercises. |
| Priority | 4 |
| Precondition(s) | New exercises must be discovered/learned before they can be added to the app. |
| Postconditions(s) | When discovered the new exercises are added to their respective group in the app. |

#### 

#### **4.2** **Non-Functional Requirements**

#### 4.2.1 Maintain Accurate Data

| Title | Accuracy of the Data Used |
| --- | --- |
| Description | This relates to user privacy and security, as well as the app’s performance. If our data is inaccurate the functionality of the app will be impacted. If the data is not accurate users are not inclined to use the app, especially if their data is at risk |
| Priority | 4 |
| Applicable FR(s) | View App Analytics |