

# *myGymClub Project Report*



Image Source: <https://health.clevelandclinic.org/wp-content/uploads/sites/3/2013/09/inexpensiveExercise-1277759983-770x533-1.jpg>

*Prepared by*

*Luke Repta, Mike Apreza, Dongxin Zhang, Tao Wang*

**CS 440  
at the  
University of Illinois Chicago**

**April 2022**

## Table of Contents

	List of Figures	7
	List of Tables	8
I	Project Description	9
1	Project Overview	9
2	The Purpose of the Project	9
2a	The User Business or Background of the Project Effort	9
2b	Goals of the Project	10
2c	Measurement	10
3	The Scope of the Work	10
3a	The Current Situation	10
3b	The Context of the Work	11
3c	Work Partitioning	12
3d	Competing Products	15
4	The Scope of the Product	15
4a	Scenario Diagram(s)	16
4b	Product Scenario List	18
4c	Individual Product Scenarios	19
5	Stakeholders	22
5a	The Client	22
5b	The Customer	22
5c	Hands-On Users of the Product	22
5d	Maintenance Users and Service Technicians	24
5e	Other Stakeholders	24
5f	User Participation	24
5g	Priorities Assigned to Users	24
6	Mandated Constraints	25
6a	Solution Constraints	25
6b	Implementation Environment of the Current System	25
6c	Partner or Collaborative Applications	25
6d	Off-the-Shelf Software	26
6e	Anticipated Workplace Environment	26
6f	Schedule Constraints	26
6g	Budget Constraints	26
7	Naming Conventions and Definitions	26

7a	Definitions of Key Terms	26
7b	UML and Other Notation Used in This Document	27
7c	Data Dictionary for Any Included Models	27
8	Relevant Facts and Assumptions	27
8a	Facts	27
8b	Assumptions	27
II	Requirements	28
9	Product Use Cases	28
9a	Use Case Diagrams	28
9b	Product Use Case List	29
9c	Individual Product Use Cases	30
10	Functional Requirements	48
11	Data Requirements	53
12	Performance Requirements	55
12a	Speed and Latency Requirements	55
12b	Precision or Accuracy Requirements	56
12c	Capacity Requirements	56
13	Dependability Requirements	57
13a	Reliability Requirements	57
13b	Availability Requirements	57
13c	Robustness or Fault-Tolerance Requirements	58
13d	Safety-Critical Requirements	58
14	Maintainability and Supportability Requirements	59
14a	Maintenance Requirements	59
14b	Supportability Requirements	59
14c	Adaptability Requirements	59
14d	Scalability or Extensibility Requirements	60
14e	Longevity Requirements	60
15	Security Requirements	60
15a	Access Requirements	60
15b	Integrity Requirements	61
15c	Privacy Requirements	61
15d	Audit Requirements	62
15e	Immunity Requirements	62
16	Usability and Humanity Requirements	62

16a	Ease of Use Requirements	62
16b	Personalization and Internationalization Requirements	63
16c	Learning Requirements	63
16d	Understandability and Politeness Requirements	63
16e	Accessibility Requirements	64
16f	User Documentation Requirements	64
16g	Training Requirements	64
17	Look and Feel Requirements	65
17a	Appearance Requirements	65
17b	Style Requirements	65
18	Operational and Environmental Requirements	65
18a	Expected Physical Environment	65
18b	Requirements for Interfacing with Adjacent Systems	66
18c	Productization Requirements	66
18d	Release Requirements	67
19	Cultural and Political Requirements	67
19a	Cultural Requirements	67
19b	Political Requirements	68
20	Legal Requirements	68
20a	Compliance Requirements	68
20b	Standards Requirements	68
21	Requirements Acceptance Tests	70
21a	Requirements – Test Correspondence Summary	70
21b	Acceptance Test Descriptions	71
III	Design	75
22	Design Goals	75
23	Current System Design	76
24	Proposed System Design	76
24a	Initial System Analysis and Class Identification	76
24b	Dynamic Modeling of Use-Cases	78
24c	Proposed System Architecture	80
24d	Initial Subsystem Decomposition	80
25	Additional Design Considerations	81
25a	Hardware / Software Mapping	81
25b	Persistent Data Management	82

25c	Access Control and Security	82
25d	Global Software Control	82
25e	Boundary Conditions	82
25f	User Interface	83
25g	Application of Design Patterns	83
26	Final System Design	84
27	Object Design	85
27a	Admin Package	85
27b	Server Package	86
27c	Workout Package	87
27d	Member Package	87
27e	SocialMedia Package	88
IV	Project Issues	88
28	Open Issues	88
29	Off-the-Shelf Solutions	88
29a	Ready-Made Products	88
29b	Reusable Components	89
29c	Products That Can Be Copied	89
30	New Problems	89
30a	Effects on the Current Environment	89
30b	Effects on the Installed Systems	89
30c	Potential User Problems	89
30d	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	89
30e	Follow-Up Problems	90
31	Migration to the New Product	90
31a	Requirements for Migration to the New Product	90
31b	Data That Has to Be Modified or Translated for the New System	90
32	Risks	90
33	Costs	90
34	Waiting Room	90
35	Ideas for Solutions	91
36	Project Retrospective	91

V	Glossary	92
VI	References / Bibliography	93
VII	Index	94

## List of Figures

Figure 1 - Context of the Work	12
Figure 2 - Joining the Gym Scenario Diagram	16
Figure 3 - Working Out Scenario Diagram	17
Figure 4 - User Edits Profile Scenario Diagram	18
Figure 5 - Gym-Employee-Included Use Case Diagram	28
Figure 6 - User-Focused Use Case Diagram	29
Figure 7 - Initial Overall Class Diagram	77
Figure 8 - Account Creation Sequence Diagram	78
Figure 9 - Set Goal Sequence Diagram	78
Figure 10 - Delete Goal Sequence Diagram	79
Figure 11 - Login Sequence Diagram	79
Figure 12 - Checking In Sequence Diagram	80
Figure 13 - Initial Class Diagram, Broken into Subsystems	81
Figure 14 - Hardware / Software Map	81
Figure 15 - UI Mockup	83
Figure 16 - Final System Design Diagram	84
Figure 17 - Admin Package	85
Figure 18 - Server Package	86
Figure 19 - Workout Package	87
Figure 20 - Member Package	87
Figure 21 - SocialMedia Package	88

## **List of Tables**

<i>Table 1 - Business Event List Table</i>	<i>12-14</i>
<i>Table 2 - Product Scenario Table</i>	<i>18-19</i>
<i>Table 3 - Acceptance Tests Correspondence</i>	<i>70</i>



# **I Project Description**

## **1 Project Overview**

myGymClub is a mobile application designed for member-only gymnasiums to motivate and retain their customers through a rewarding and engaging exercise augmentation app. Many members rapidly sign up for a gym only to have trouble trying to find motivation to work out. Therefore, in the app, there is a level system. Users could get experience to level up by signing in everyday, finishing exercise tasks, uploading exercise teaching videos, setting up a goal of body fat percentage, reaching those goals, and so on. There will be an achievement system where users will earn medals after completing a certain set of exercises. To make the level and achievement meaningful, there is also a social system that enables users to login and a rewards system. With the social aspect, users are able to see their friends' achievements and levels, so that they can have a stronger motivation to catch up with their friends whose level is higher. The exact rewards would be up to each gym to decide, but common items would be coupons/discounts for in-gym purchases like food, drinks, clothing. A coin system is another idea as well. Members earn eCoins after completing exercise(s) and could be used to exchange some goods from local gyms or in the eShop.

## **2 The Purpose of the Project**

The purpose of this project is both to help people who have issues with finding motivation to work out and help gyms to maintain a recurring revenue stream. After a while of using this app, it will be easier for people to stick to working out, have fun, and have a great feeling of accomplishment. With these in mind, people are more likely to stay with their respective gym as a member. Through enhanced integration of the application with the gym equipment, the user would be able to have a more satisfying exercise experience, and would facilitate their continued membership at the gym.

### **The User Business or Background of the Project Effort**

The current user business model involves people coming to gyms and signing up for memberships, where they workout with Gym provided equipment. Despite offering special sign up bonuses, and offerings like complementary one-on-one training sessions, Gym memberships continue to decline precipitously as the year goes on.

The project is intended to make the app users happy by letting them have an entertaining, consistent, and rewarding work out experience while also seeing the improvement in their physical and mental state. Alongside this, gymnasiums that use this software would be able to better maintain/increase the number of members, and as a result have a larger, more consistent revenue stream. Gyms would use our software by giving all members access to the application.

The matter of decreasing gym memberships is of extreme importance for these establishments, because having a more effective way of retaining and gaining members directly increases their revenue. Despite Gyms already having many special bonuses and promotions to try to attract new members, memberships continue to fall, which is why our product would be so valuable for these Gyms.

### **Goals of the Project**

The goal of this project is to help member-only fitness centers improve their membership retention rate, and as a result, increase the revenue generated. Many gyms see large upswings in membership around times like new years, but they do not retain these new customers, resulting in lost possible revenue. The goal of the project is to directly reduce the number of membership cancellations and increase the number of members that a Gym has at the end of the year.

### **Measurement**

To know when our goals are met, we will monitor the number of members left in gyms, graphing the data and look for the long term trend. We will measure the number of memberships and revenue before and after implementing the software, and we can also measure the average length of membership before and after. We will deem the project to have met its goals if we are able to achieve the following metrics in the first few years of operation: increasing average membership length by at least 1-2 months; increasing membership numbers by at least 10-20%, and achieving at least a 30-40% revenue increase for the Gyms using the application.

## **3 The Scope of the Work**

The scope of the work of this project is to teach people to maintain healthy habits, teach them about consistency, and push them to become more fit as a result of increased motivation for exercise. All of these objectives will help fitness centers retain current members and increase membership levels.

### **The Current Situation**

Currently, gyms have memberships that allow their clients to visit the location and work out as often as they like each month they pay for a membership. This model presents a problem because many people find it difficult to stay motivated and continue exercising consistently, and as a result find they aren't using their membership enough to warrant continuing to pay for it, and therefore cancel it. Many gyms also offer group classes to make it more exciting for the customers, but gyms are not retaining members as much as possible. Personal trainers are also offered sometimes, but that sometimes cost extra money outside the membership, meaning many are not willing to pay extra. Some more popular fitness centers have mobile applications that let people keep track of their progress and see correctly demonstrated exercises, but this is still not enough to keep people motivated and keep

them as members. We are seeking to alleviate this problem by providing our application as a more effective way of retaining members that benefits both the gyms and their clients.

### **The Context of the Work**

The work incorporates all aspects of consistently going to the gymnasium. To do this the work must interact with four external entities, the user (possibly a gym member), a gym employee, the server, and smart gym equipment. When the user becomes a member, the gym employee gives that member a unique ID, which is saved in the server. Once the user registers with the unique ID, the server can connect the two. When the user signs in, according to the membership status shared by the gym employee, the server will allow or reject the user. If the user is valid, then they can request their statistics from the smart gym equipment, which will be sent to the user before being sent to the server in order to be saved and make any calculations if applicable. Every time the user edits their profile, the server will save the updated data. The gym employee will be able to share gym-specific goals and events with the user. If the user ever logs out and logs back into the app, the server can load that data. Every time the user participates in any activity, their status will be sent to the server to determine if they are eligible for a reward according to goals set by the user and gym employee(s). When requested, the server can also share the user's friends' achievements and level.

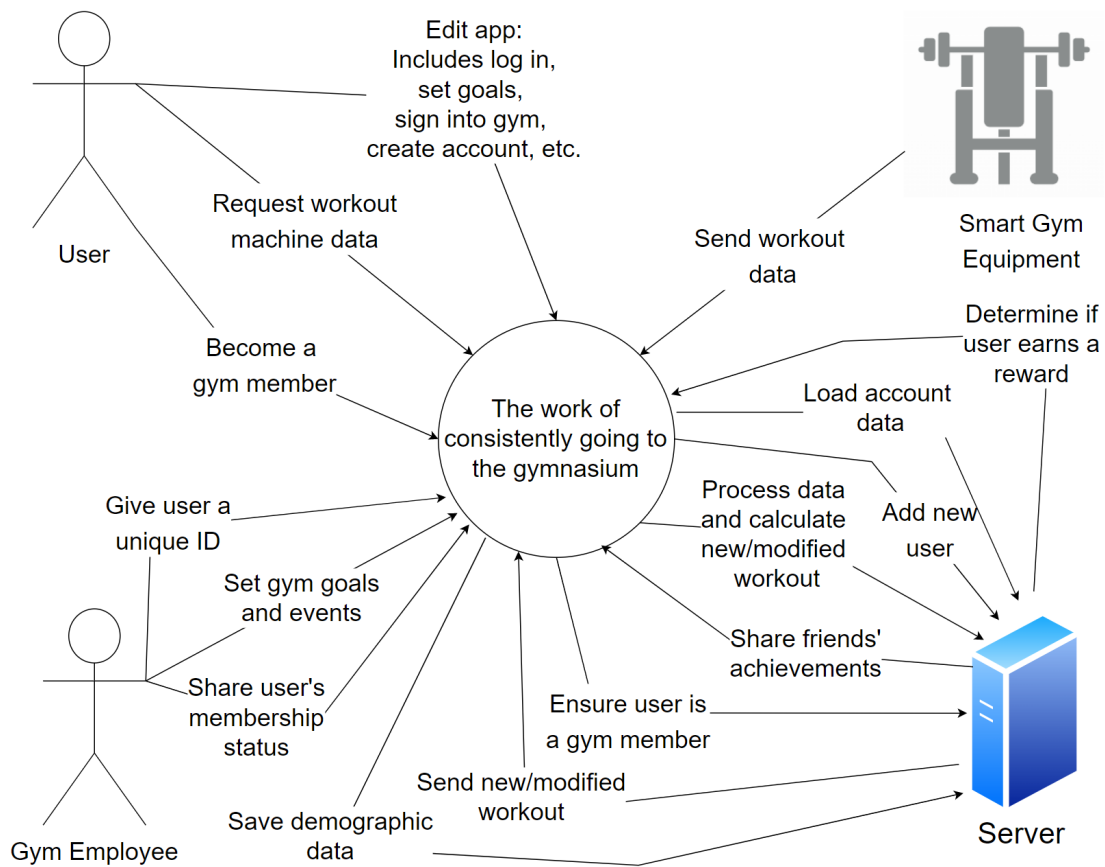


Figure 1 – Context of the Work

## Work Partitioning

### Business Event List

User buys Gym Membership	User buys a Gym Membership (input)  An account is created and given to the user (output)	When the user goes to a participating Gym and buys a membership, the Gym staff will send a request to the server to create an account, which will be given to the user.
--------------------------	--	---

User Logs in to App	<p>User enters username and password, sends request to server(input)</p> <p>Server acknowledges and allows login (output)</p>	<p>When a user attempts to log in to the app, it sends a request to the server with the entered username and password. If the server finds it to be a valid login, it sends back an acknowledgement to the user and allows the login.</p>
App queries Server for New/Modified Workouts	<p>User/App requests New Workouts (input)</p> <p>Server sends new workouts to user (output)</p>	<p>When the user/application requests new workouts, the server will process their account information, and send personalized workouts to the user</p>
User sends workout Information to Server	<p>User Workout Information (input)</p> <p>Information Received Acknowledgement (output)</p>	<p>When the user sends workout information to the server from their device, the server will process the data and add it to the user's account in the database, and send an Information received acknowledgment to the user's app.</p>
User's Gym Membership Expires	<p>User's Gym Membership Expiration Message (input)</p> <p>Membership Expiration Message to User (output)</p>	<p>If a user's membership with a Gym expires, the server will mark their account as no longer being a member, preventing access to the app. The user will receive a notice of their membership expiration, and of losing access to the application.</p>
Server is close to Storage	Server Storage Capacity	If the server is near to

capacity	Warning (input)  Acquire more storage capacity (output)	reaching its storage capacity, the server technician will receive a warning. The server technician should then request that more storage capacity be added
Server is at Bandwidth Capacity	Server Bandwidth Warning (input)  Throttle Connections + Message Users (output)	If the server is at bandwidth capacity, it will receive a warning. When the server receives this warning and is receiving more bandwidth than it can handle, it will throttle the existing connections and prompt users of possible slowdowns due to heavy load.
Server Crashes	Error Message to Server (input)  Error Message to Users (output)	If the server should crash, the server technician will set up an error message for users attempting to use the server indicating that the server is temporarily down.
App is Updated	App Update (input)  Update Data for Users (output)	When the application is updated, the developers will send the app update to the server. The server will then send this update data for all users, which will update the app on their device

*Table 1 - Business Event List*

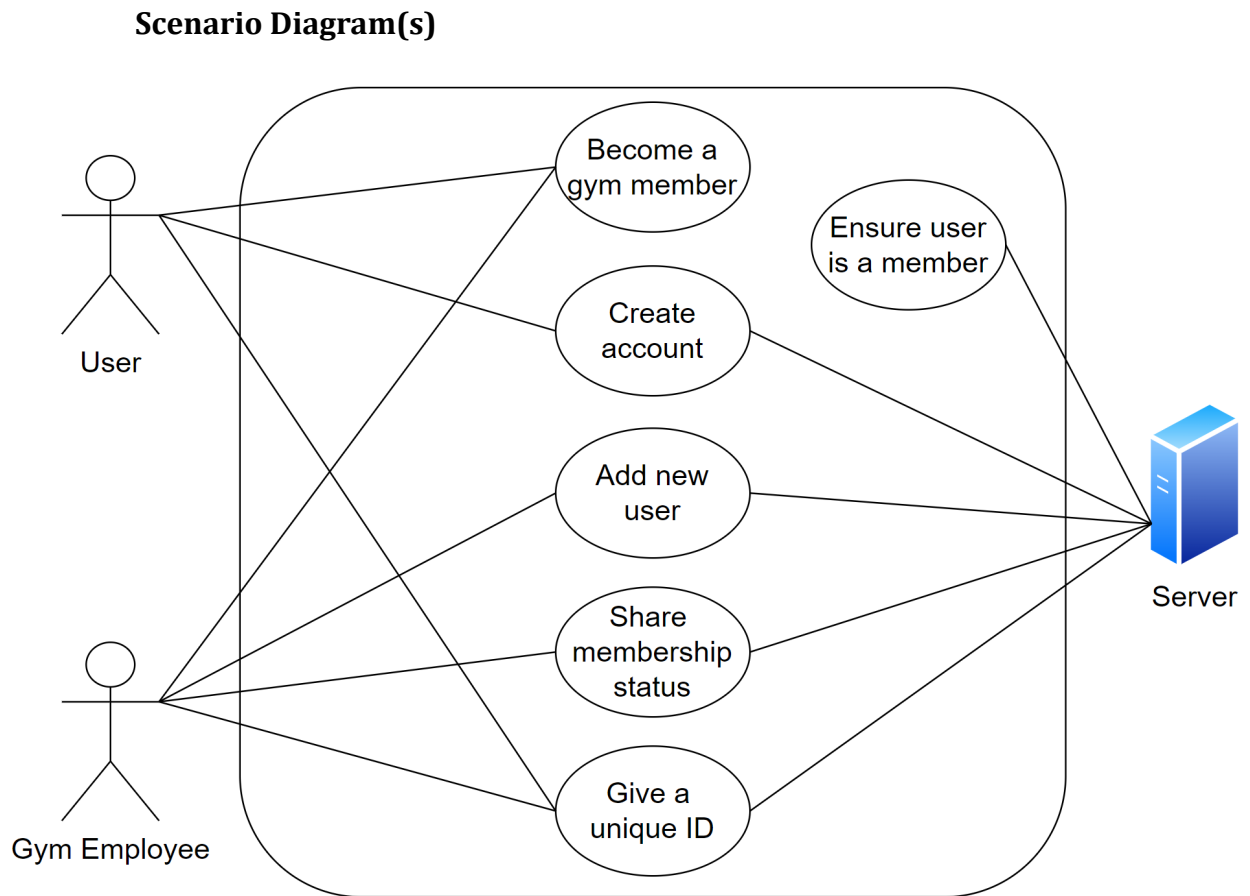
## **Competing Products**

Some gymnasiums offer a free fitness consultation to give new members a workout plan. One-on-one training is also offered. Some fitness centers offer it for free to members. Our app is not meant to replace professional help. Most people would not consult a trainer every week or so, which is what is recommended. However, based on current abilities and workout routine, a safe, more challenging workout can be made with the help of appropriate sensors if possible. This is especially beneficial to small gymnasiums, which need their members to stay more urgently.

Even less gyms have mobile applications. Currently, most help clients keep track of their progress and offer video tutorials on how to do certain exercises. In spite of these features, gyms still have a great loss in terms of their membership numbers and an even greater loss of potential revenue. Our app offers a competitive feature where the user can see their friends' achievements and levels. This can be motivating for competitive people. For reward-seeking people, they can earn rewards from accomplishing gym-specific goals. Members will pursue these feelings, and soon enough, they will have developed a healthy habit.

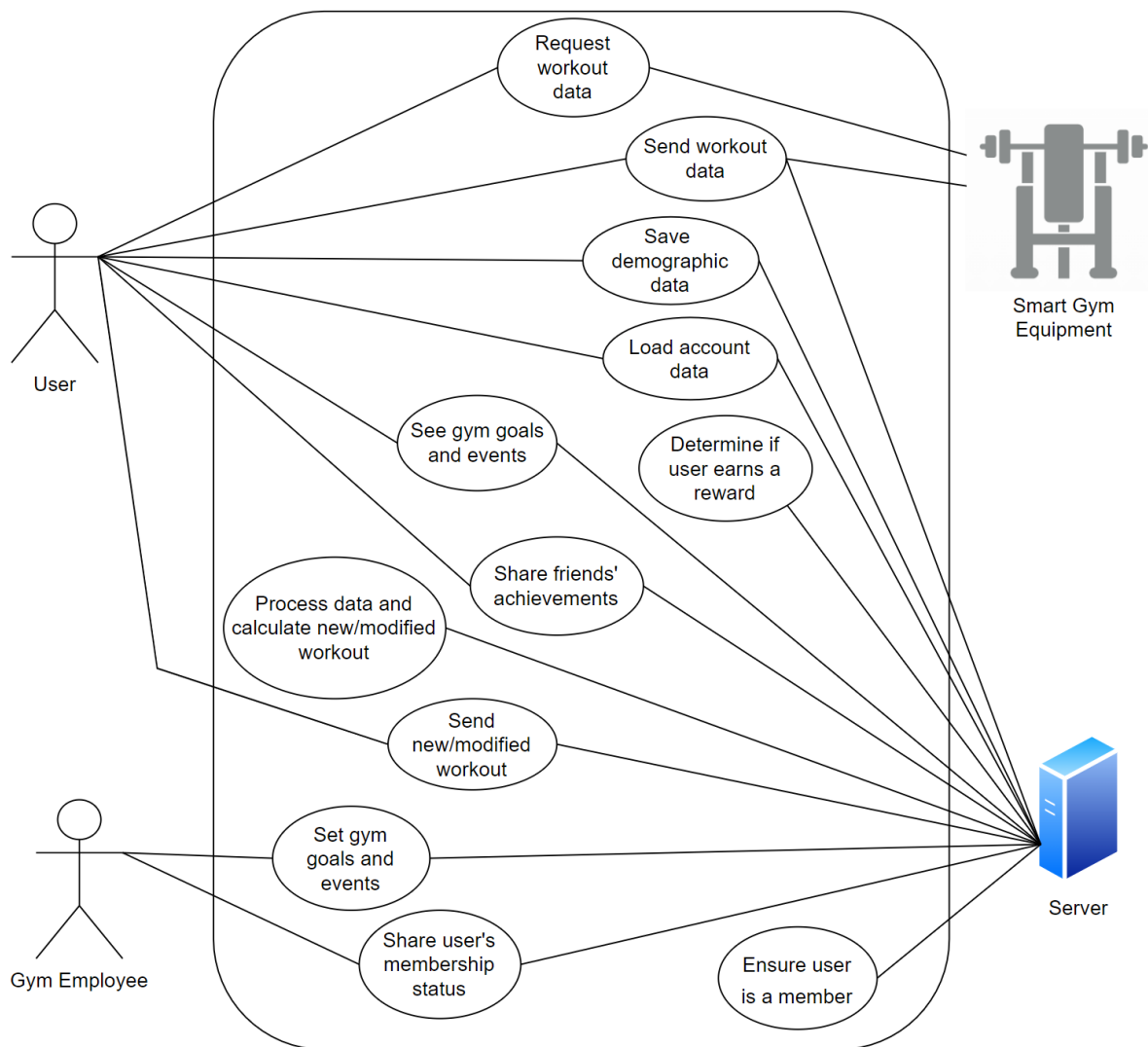
## **4 The Scope of the Product**

This product would be composed of the following key features: recording and storing activity/workout statistics from user workouts that can be viewed by said user at any time; using data from real time sensors on in-gym smart equipment to perform calculations such as calories burned (as a function of distance traveled, heart rate elevation, steps taken, height, weight, etc.); presenting users with personalized workout regiments and challenges based on their unique fitness level and individual requirements; have unique daily/weekly log-in rewards and customized challenges based on the user's individual fitness level; allow for social interaction in app by means of user profiles, friends, and events.

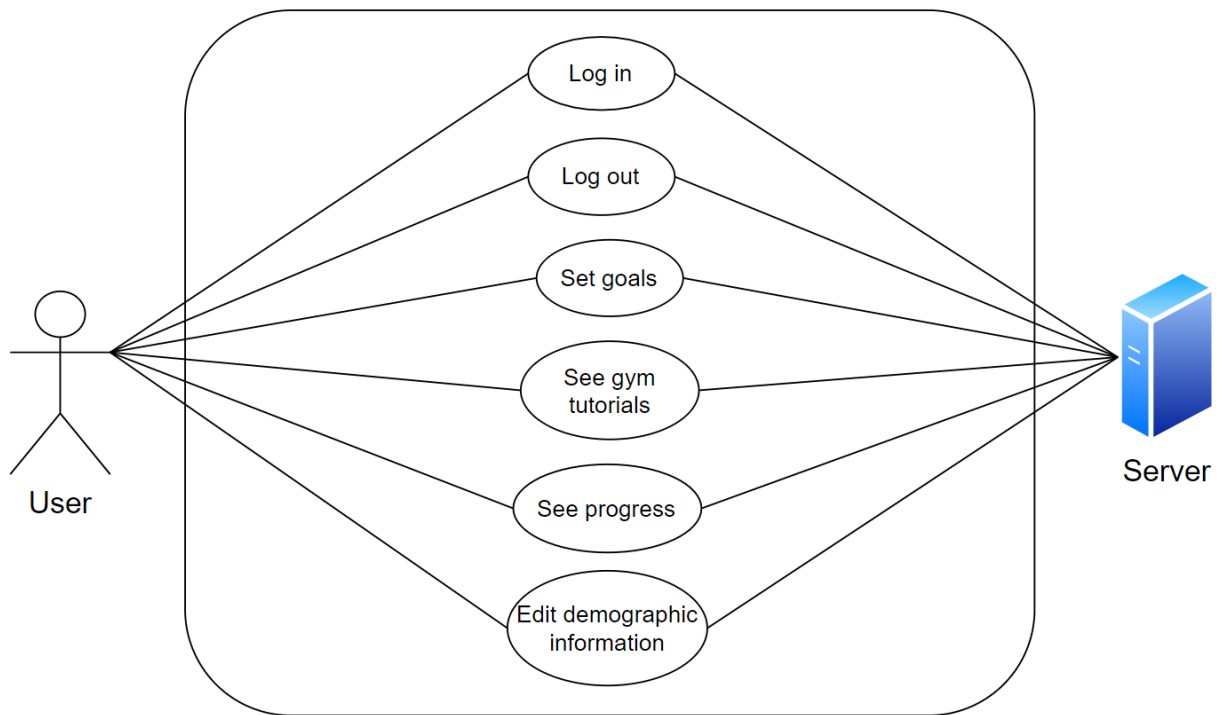


*Figure 2 - Joining the Gym Scenario Diagram*





*Figure 3 - Working Out Scenario Diagram*



*Figure 4 - User Edits Profile Scenario Diagram*

### Product Scenario List

Name	External Actor(s)
Log in	User, Server
Log out	User, Server
Set Goals	User, Server
See Gym Tutorials	User, Server
See Progress	User, Server
Edit Demographic Information	User, Server
Become a Gym Member	User, Gym Employee
Create Account	User, Server

Add New User	Gym Employee, Server
Share Membership Status	Gym Employee, Server
Give a Unique ID	Gym Employee, User, Server
Ensure User is a Member	Server
Request Workout Data	User, Smart Gym Equipment
Send Workout Data	User, Smart Gym Equipment, Server
Save Demographic Data	User, Server
Load Account Data	User, Server
Determine if User Earns a Reward	Server
See Gym Goals and Events	User, Server
Share Friends' Achievements	User, Server
Set Gym Goals and Events	Gym Employee, Server
Process Data and Calculate New/Modified Workout	Server
Send New/Modified Workout	User, Server

*Table 2 - Product Scenario List*

### **Individual Product Scenarios**

**Log In Scenario:** When a user first opens the app, they will be prompted to log into the app using their username and password. If this is their first time opening the app, they will be prompted to create an account/password as per the “Create Account Scenario”. When the user logs in, the username/password will be sent to the server to be hashed. The server will never store plain text passwords.

**Log out Scenario:** The user can log out of the application by clicking the ‘logout’ button. This will send a request to the server for the user’s username/password account to be logged out. When a user is logged out, the server will not constantly log and update their activity levels.

**Set Goals Scenario:** The user can set personal goals through the app by going to the “Set and View Goals” section, where they can choose from a variety of exercise goals, and input exact numbers. When the user creates a goal, it is sent to the server to be logged on their account.

See Gym Tutorials: The user can open the help menu where they will have the option to see tutorials for their specific Gym. This will send a request to the server, which will check in its database for the location the user is a member of, and send back the appropriate tutorial for that specific Gym location. The user will be able to read/watch videos here, depending on what the specific Gym supplies, and the more general tutorials provided for all Gyms.

See Progress Scenario: When the user presses the “See Progress” button, a request will be sent to the server to retrieve the current progress statistics for the user. This will retrieve things like miles run, weight lifted, heart rate data, etc. Once the user’s device receives this information, it will display it in a user-friendly interface, with charts and graphs showing the user’s progress over time.

Edit Demographic Information Scenario: The user will be able to select the “Edit Demographic information” option from a menu, where they will be able to change their demographic information such as age, weight, gender, etc. When the user clicks confirm, this information will be sent to the server, which will update the user’s records.

Become A Gym Member Scenario: When a user desires to become a member of a Gym that uses our application, they will be enrolled as a member by one of the Gym staff members, and they will receive instructions on how to install and use our application. The staff member will begin the “Add New user” scenario, and the user will be guided through the process of creating an account via the “Create Account” scenario.

Create Account Scenario: When the user first installs and opens the application, they will be prompted to create an account by entering a username and password (alongside personal information like phone number, email address, etc). This will send a “create account” request to the server, which will hash the username/password, check if such an account already exists in the database, if so prompt the user to use a different username, otherwise it will create a new account for the user in the database. Plain text passwords will never be stored.

Add New user Scenario: When a user first becomes a member of a Gym that uses our application, a Gym staff member will select the “Add New User” button on their employee version of the application, which will send a request to the server to begin the account creation process, leading to the “Create Account” Scenario.

Share Membership Status Scenario: When a user becomes a member of the Gym and creates an account, or when a user chooses to terminate their membership with the Gym, a Gym staff member will send a request to the server to have the user’s membership status changed.

Give a Unique ID Scenario: During the “Create Account” Scenario, after the user has created their account, the Gym staff member will send a request to the server to

generate a new unique ID. The server will then add this newly generated ID to the user's account, so that they can be uniquely identified other than through their username and password.

Ensure User is a Member Scenario: When the server receives a login request and it has validated that it is the correct username/password for an account, it will then check if the user is a member of a Gym using our application. If they are, the server will continue the login process normally, otherwise, the server will send back that the user is not a member, and that they can only access the application if they are a member of a supported Gym.

Request Workout Data Scenario: The user may manually, or automatically through their smart device interfacing with the smart gym equipment, retrieve the data of a workout from the smart Gym equipment's sensors (Like distance traveled, heart rate, steps taken for a smart treadmill) and send it the user's device.

Send Workout Data Scenario: When the user completes a workout, the "Request Workout Data" Scenario will be initiated between the user and the Smart Gym Equipment. When this finishes, the user's device will send a request to the server to add a workout. The server will receive this request, and add this workout to the users account in the database

Save Account Data Scenario: The user can select the "Save" button from the user's demographics menu, which will send the user's account information to the server, which will update the user's account in the database.

Load Account Data Scenario: When the user initiates a scenario like "Log-In" scenario or switches screens, the user will send a request to the server for the user's account data. The server will then retrieve the user's information from the database, and send it back to the user's device.

Determine if a User Earns a Reward Scenario: The server will perform a check of a user's account information to see if they have reached a sufficient threshold to complete their goals. If the threshold has been reached, the server will mark that the user is to earn the specified reward, which will be added to the user's account in the database.

See Gym Goals and Events Scenario: The user will select the option to See Gym Goals and Events, which will send a request to the server to access the goals and events for the Gym that the user is a member of. The server will then retrieve this data, and send it back to the user's device, which will display it in a user-friendly interface.

Share Friend's Achievements Scenario: The user will select the social tab and choose the Friend's Achievements option, which will send a request to the server to retrieve the achievements of every user added as a friend of the user. The server will then send

this information back to the user's device, which will display it in a user-friendly interface.

**Set Gym Goals and Events Scenario:** A Gym staff member will select the option to "Set Gym Goals and Events" from their staff version of the application, where they will choose what are to be the Goals and Events that all members of their Gym will be able to see. Once they confirm their choices, a request will be sent to the server containing this information, the server will then update its database with these goals and events, so that when users view the gym goals and events via the "See Gym Goals and Events" scenario, they will see the updated information.

**Process Data and Calculate New/Modified Workout Scenario:** The server will periodically or at set times (like after the user uploads some workout data) process the workout and fitness information of a user, and perform calculations to see if the user's workouts/challenges should be adjusted. Based on the results of these calculations, the server will update the user's workout/challenges in the database.

**Send New/Modified Workout Scenario:** Periodically or at set times (like user logging in, or requesting for new workouts) the server will send new/modified workouts to the user. The user's device will then display this information in a user-friendly interface.

## **5 Stakeholders**

### **The Client**

This product is to be commissioned by member-only gymnasiums such as L.A. Fitness or Planet Fitness as a means of enhancing the user experience at their gymnasiums while expanding as well as better retaining their clientele. The fitness center that commissions us will be the primary client, and depending on the negotiations with said client, we may be able to also license out the software to other gyms. The clients are to provide appropriate sensors and integration with the physical equipment in their gym in order for our product to be utilized most effectively.

### **The Customer**

The product will not have external customers, rather the users will be the members of the client gym(s) that employ our product, which will be an amalgamation of people of all ages and backgrounds. The software will not be 'bought' by these members, instead, they will receive access to the software so long as they are members of the client gym(s) that bought the product. Depending on the contract negotiated with the initial client gym that commissioned the product, we may be also able to license out the software to other gymnasiums..

### **Hands-On Users of the Product**

1.) User: Gym Members

i.) Role: Members of the Gym that go to the location to use their machines/equipment to exercise. Will use product to enhance and augment their exercise

ii.) Subject Matter Experience: Members will require no background in exercise/physical activity, product will be useful and fun for people of all levels of fitness. They will be rated as novices.

iii.) Technological Experience: Members will require no background in software beyond knowing how to use basic apps on their smart device. They will be rated as novices.

iv.) Other Characteristics:

a.) Age: All ages

b.) Gender: All

c.) Motivation: Using this app to keep them motivated and having fun exercising.

## 2.) User: Gym Staff

i.) Role: Gym Staff will facilitate social interaction through interactive on-location events for app users, and also provide rudimentary app support through explaining how the product works for members.

ii.) Subject Matter Experience: Gym Staff will be generally more knowledgeable about exercise and physical activity than the average member. They will be rated as journeymen.

iii.) Technological Experience: As part of the hiring/training process, Gym staff will be instructed in how to use the app so they can teach members/create events. They will be rated as journeymen.

iv.) Other characteristics:

a.) Age: Generally younger than 40

b.) Gender: All

c.) Motivation: Facilitate the efficient and quality use of the product by the gym members

## **Maintenance Users and Service Technicians**

Depending on contract negotiations with the client, either us, the client themselves, or a third-party will install, maintain, update, and/or service the product as needed.

## **Other Stakeholders**

Gymnasium staff will have to receive training on how the product functions, and how to create events for gym members to participate in. This will increase their responsibilities as an employee.

Client fitness center(s) may need to hire technical support to resolve issues with the product on-site, or perhaps contact us to help work out server/software problems that occur during the normal operation. We may also need to be in contact with them when we issue updates, to make sure the Gym is ready.

Management departments should filter out exercise tutorials that are not effective and harmful. Every tutorial should be fully tested before being uploaded.

## **User Participation**

Users could help test the app for any issues or bugs. They can also help test if a user can go to another gym of both the same company and different company and use the product. Gymnasium staff could help test how easy it is for a gym to maintain the product. Users could leave comments to reflect how effective the suggested exercises are and how long they are motivated.

## **Priorities Assigned to Users**

- Key users: The core users attracted by this product should be people who are just starting to work out, people who have a history of signing up and canceling gym memberships, and those who have low motivation when it comes to consistently going to the gym. This can help determine if the application is able to create motivation.
- Secondary users: Secondary targets are those who already have a fitness habit but who have not reached a particularly high fitness level. They have some room for improvement.
- Unimportant users: People who do not need to be considered are those who have no interest in joining any fitness center, or those who have a membership but do not go to the Gym (i.e. absentee members).



## **6 Mandated Constraints**

### **Solution Constraints**

#### **Constraint 1:**

Description: The product is a mobile application for smartphones.

Rationale: It is uncommon for people to have a website open while working out. Since the users (gym members) need to be sent data from smart gym equipment, a smartphone is required.

Fit Criterion: The product will consider the most popular types of smartphones and their versions in order to determine what can and cannot be done in the app.

#### **Constraint 2:**

Description: The product will be marketed mainly toward people who have a history of signing up and canceling gym memberships.

Rationale: This is a large group that can generate a lot of a gym's revenue.

Fit Criterion: The product will be designed for this group primarily.

#### **Constraint 3:**

Description: The product will be usable for at least five years.

Rationale: Most people that go to the gym are young adults and adults. When they become consistent with their routine, they will stick with it for a few years.

Fit Criterion: The product will consider people who have been at least a few years at the gym besides new members.

### **Implementation Environment of the Current System**

The environment for the app is based on Java as this is the popular language to implement a mobile application. SQL databases will be used to store data. The only smartphones that could support this application are Android phones. The iOS version should be released later once we get success on the Android version.

### **Partner or Collaborative Applications**

This application should be able to write pdf documents so that users are able to print a hard copy of their work-out records. It should also be able to send messages to technical managers informing them that the app is malfunctioning. The app would also be connected to embedded systems of gym machines to retrieve data.

## **Off-the-Shelf Software**

The off the shelf software needed for this application primarily consists of the equipment/sensors of the smart equipment that is to be present in the gyms to interface with the app. The application will receive information from the sensors on the smart gym equipment, and use it as part of its normal operations.

## **Anticipated Workplace Environment**

A solid Internet connection is expected in the gyms as the vast majority of Gym's either provide free wifi, or wifi for members. Due to the physical nature of the activities recorded by the sensors, the sensors need to not be so fragile as to be harmed by their normal operation (i.e. users lifting a heavy weight should not break the force sensor, etc.). While Gyms are not necessarily quiet, repeated noises like loud sounds for completing an achievement in-app should be avoided to provide maximum comfort for users and staff.

## **Schedule Constraints**

A working prototype will be desired as soon as possible by the Gyms hiring us, and they would likely have a few key deadline requirements. In the year that the application is scheduled for release, it should be completed before important holidays such as New Year's or Christmas, which is when the greatest number of people join gyms. Another important time for the application to be completed is the beginning of summer in June because a lot of people want to become fit for vacations, the pool, the beach, and so on.

## **Budget Constraints**

The project may not be able to get all possible kinds of sensors immediately, so we will have to develop the application with both our budget and the budget of the Gym in mind (they might have only a limited amount of capital to buy smart gym equipment, so we must primary focus on making the app work with those ). If the budget allows, some of the parts in the app may be developed to support other uncommon machines after the product is released. Smart Gym Equipment is expensive, and so while we will not be the ones providing the funding for such purchases, we may have to balance our development budget with the smart equipment budget so as to provide the best application for our client Gyms.

## **7 Naming Conventions and Definitions**

### **Definitions of Key Terms**

- Gym/Gymnasium/Fitness Center - A club, building, or large room, that usually contains special equipment, where people go to physically exercise (also known as working out) and/or participate in athletics.

- Member-Only Gym(s)/Gymnasium(s)/Fitness Center(s) - A club, building, or large room, that usually contains special equipment, where people go to physically exercise (also known as working out) and/or participate in athletics as long as they are a member of the group.
- Smart Gym Equipment - Special electronic machines that are used during physical exercise to enhance a person's physical state. These machines calculate information about the exercise the person is doing such as how many calories were burned, how much time they have been on the machine, and so on.
- Experience - A unit of measurement that quantifies the user's experience and progression throughout their time at the gymnasium. Also known as exp or XP.
- Level - A position on an imaginary numeric scale that quantifies a user's experience. It is also a positive whole number.
- Achievement - A task set by the gym member or employee(s) and is done successfully by the user.
- Reward - A gift given to a gym member in recognition for their effort, achievement, or time.
- Event - A planned social occasion set by someone.

### **UML and Other Notation Used in This Document**

This document follows the version 3.0 OMG UML standard. Any exceptions are noted where used.

### **Data Dictionary for Any Included Models**

- A SQL database is a structured set of data in a computer that is created and accessible by using a domain-specific language: Structured Query Language (SQL).
- The SQL database contains at least the user's unique ID, the user's age (valid values: 0-99), weight in pounds, name (text), membership status (0 for "not a members" and 1 for "is a member"), gym location (if there is more than one type of gym and/or location), the user's level and experience (values to be determined), contact information (text), list of friends, the user's current workout(s), the user's workout history, and the gym's goals and events. Not all the information will be in one table.

## **8 Relevant Facts and Assumptions**

### **Facts**

To use the mobile app, people cannot directly download it from the internet and instead need to find it at their smart devices mobile application store where they can

download the app. The user will have to become a member at a participating gym first before being given a unique identification number to access the app and create an account.

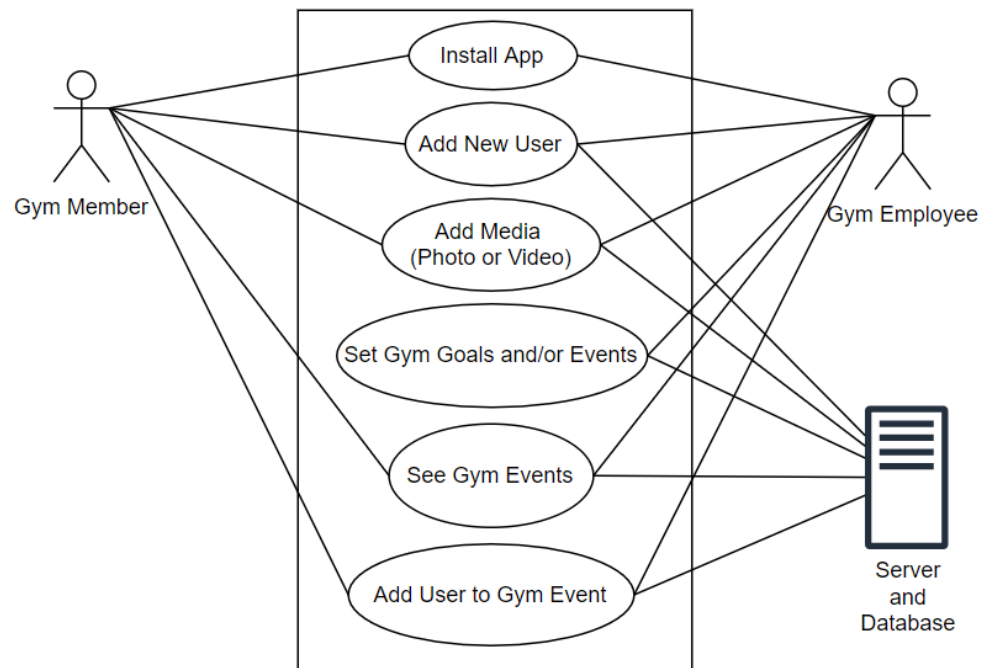
### Assumptions

We assume that users run it on Android and have sufficient space for downloading the mobile app. Users will be guided by staff on how to use each functionality of the app. We expect that the Gyms will hire their own technical support staff to interact with the application and provide support if necessary, rather than having us present on location to provide technical assistance.

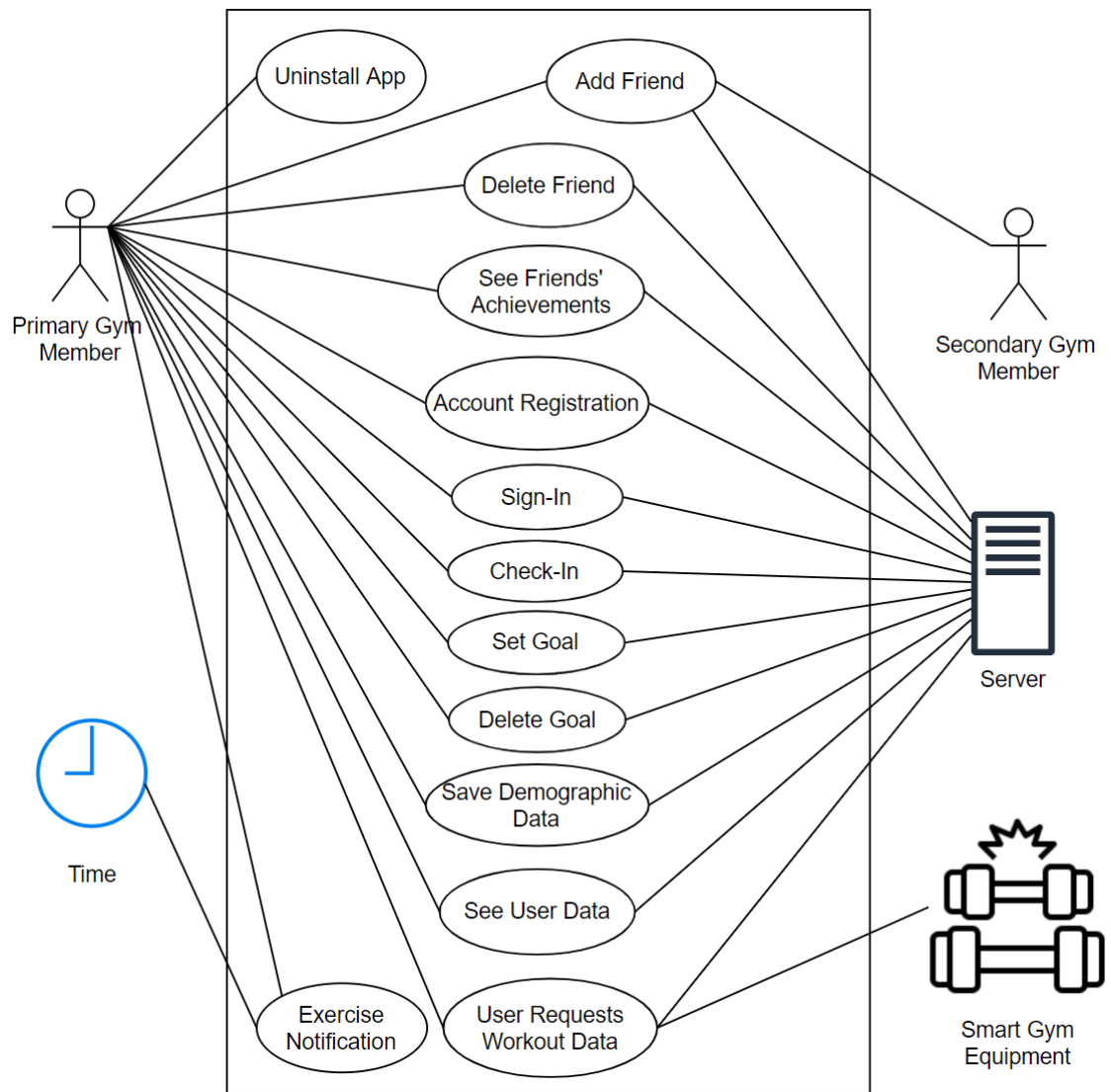
## II Requirements

### 1 Product Use Cases

#### 1a Use Case Diagrams



*Figure 5 - Gym-Employee-Included Use Case Diagram*



*Figure 6 - User-Focused Use Case Diagram*

## 1b Product Use Case List

Not Applicable

## 1c Individual Product Use Cases

<p>Use case ID: 1                      Name: Install App</p> <p>pre-conditions: The server and database for the app has been set up and up to date; The gym employees have been taught how to download the app and guide users; The users should have a valid smartphone.</p> <p>post-conditions: Users should be able to open the app on their smartphone without any errors or crashing.</p> <p>Initiated by: User</p> <p>Triggering Event: The individual joins the gym as a member, and the gym employee asks the person to download the app, to which they accept.</p> <p>Additional Actors: Gym Employee</p>
<p>Sequence of Events:</p> <ol style="list-style-type: none"><li>1. The individual becomes a gym member.</li><li>2. The gym employee asks the person if they want to download the app.</li><li>3. The individual agrees before finding and downloading the application, perhaps with the guidance of the gym employee.</li><li>4. The gym employee instructs the user to open the app.</li><li>5. The user clicks on the app icon.</li><li>6. The application loads and shows a welcome screen.</li></ol>
<p>Alternatives: The user decides to download the app on their smartphone without a gym worker being present.</p> <p>Exceptions:</p> <p>If a user does not have a smartphone, they will not be able to download the myGymClub app.</p> <p>A smartphone with insufficient RAM or not enough storage will not be able to open the app.</p>

Use case ID: 2

Name: Add a New User

pre-conditions: The user has successfully become a gym member.

post-conditions: The user will be able to register for an account and access their data.

Initiated by: User

Triggering Event: The user is applying for a gym membership.

Additional Actors: Gym Employee

Sequence of Events:

1. The user is in the process of applying for a gym membership.
2. The gym employee approves of the registration in the system.
3. Server adds a row for the user in the database giving them a unique ID and a boolean value to indicate that the user is a member.
4. The gym employee shares the unique ID with the user.

Alternatives: If the user is registering online with no gym employee present, the application will handle the call to the server. Once registered, the app will share the unique ID with the user via email or directly on the screen.

Exceptions: If the server cannot be reached, or cannot add a row in the database due to there being not enough space, or any other kind of server error, an error message will appear stating that there is a problem with accessing the database or server, respectively.

Use case ID: 3

Name: Save Demographic Data

pre-conditions: User is a current gym member, has downloaded the app, and has an account. Also, the user's smartphone has a stable wifi connection.

post-conditions: The user's data is up-to-date on the server.

Initiated by: User

Triggering Event: User clicks on the "Save" button after editing some field(s) in the demographics tab of the app.

Additional Actors: N/A

Sequence of Events:

1. The user clicks on the "Save" button after editing some demographic field(s) in the app.
2. The app disables the "Save" button.
3. The data is sent to the server.
4. The server modifies the appropriate table(s) and returns a successful message to the app.
5. The app enables the "Save" button.

Alternatives: N/A

Exceptions:

If the user does not have a sufficient wifi connection, the modified data will be saved locally and updated on the server once they have a stronger wifi connection.

If the server encountered an error in saving the data, it would return an error message saying that something went wrong on the server side.



Use case ID: 4

Name: Account Registration

pre-conditions: The user is a valid gym member and has their unique ID available. The app is also downloaded on their smartphone. The user has a stable wifi connection.

post-conditions: The user has an account and is able to use the app.

Initiated by: User

Triggering Event: The user clicks on the “Create Account” button.

Additional Actors: N/A

Sequence of Events:

1. The user clicks on the “Create Account” button.
2. App pulls up the form with the information the user has to fill in including the unique ID field.
3. The user fills out all the required information and any optional information they want to fill out.
4. The user clicks the “Submit” button.
5. App sends the information to the server.
6. Server adds the information to the appropriate database table(s).
7. Server sends a success message to the app.
8. The app changes the visual to the account profile scene with the user's information.

Alternatives: N/A

Exceptions:

If the ID entered by the user is not valid, a message will be sent to the app that the ID is not valid, which is also shown to the user.

If the server or app has any error, the user will be shown a message that there is an error with the server or application, and is prompted to please try again later.

Use case ID: 5

Name: Add Media (Photo or Video)

pre-conditions: The individual is a trusted user with no account violations. If the media is a video, it must be below the length and size requirements. The person filled out the required fields for the media's information, and also has a stable wifi connection.

post-conditions: The individual is able to share media with their friends, or the current gym members are able to see gym-made exercise tutorial(s) uploaded by an employee.

Initiated by: User or Gym Employee

Triggering Event: Person clicks on the "Add Media" button.

Additional Actors: N/A

Sequence of Events:

1. The user clicks on the "Add Media" button.
2. The information is sent to the server to validate.
3. The server accepts the video/photo and saves it in the appropriate place of the appropriate people to be able to see.
4. Server sends a success message for the user to see.

Alternatives: N/A

Exceptions:

If the video/photo is inappropriate, the server will alert a gym employee and not save the video/photo and information with it.

The server encountered an error, and will send an error message to the user.

Use case ID: 6

Name: Check-In

pre-conditions: The user is a gym member that has the app downloaded, account created, and signed into their account. The user's smartphone has a stable wifi connection.

post-conditions: The user has more experience points/has received their check-in reward.

Initiated by: User

Triggering Event: User logs into their account or opens the app (already logged in).

Additional Actors: N/A

Sequence of Events:

1. The user logs into their account or opens the app if they're already logged in.
2. Server makes sure that the user is inside the gym.
3. App shows the increase in the user's level/disperses their check-in reward.
4. App notifies the server of the check-in and sends the amount of experience/reward to be added.
5. Server adds experience points to the user's overall level and determines if the user has reached a goal. If so, a message is sent to give the user a reward, else it sends a success message.
6. The app switches to the workout-ready scene.

Alternatives: If the user signs in with no wifi, then the experience added will be stored locally until the user connects to wifi.

Exceptions: If the user is not in the gymnasium, they receive no experience points.

Use case ID: 7

Name: Set Gym Goals and/or Events

pre-conditions: The software has been set up for the gym and the gym employees have been trained to use it.

post-conditions: The gymnasium has gym-specific goals and/or events sharable to gym members.

Initiated by: Gym Employee

Triggering Event: The employee has clicked on the button on their device to add a gym goal or event.

Additional Actors: N/A

Sequence of Events:

1. The employee has clicked on a button on their device to add a gym goal or event.
2. The software switches to the appropriate screen.
3. The employee fills out the required information and any optional information.
4. The employee clicks on the appropriate button to submit the information.
5. Server saves the information and sends a success message back to the employee.
6. Server sends notification to gym members who have their notifications on.

Alternatives: N/A

Exceptions: If the server is experiencing an error, an error message is displayed to the gym employee.

Use case ID: 8

Name: Exercise Notification

pre-conditions: User has set up their chosen interval of time for a workout reminder.

post-conditions: User receives a notification to go to the gym.

Initiated by: Time

Triggering Event: User did not go to the gym by the scheduled time.

Additional Actors: N/A

Sequence of Events:

1. User did not go to the gym by the scheduled time.
2. The app sends the chosen kind of notification to the user.
3. User sees the notification.

Alternatives: N/A

Exceptions: The user has disabled the notification system.

Use case ID: 9

Name: Set Goal

pre-conditions: The user has an account and is a current gym member. The user has a stable wifi connection.

post-conditions: The user will be able to see and access their set goal.

Initiated by: User

Triggering Event: The user has tapped on the “Add Goal” button.

Additional Actors: N/A

Sequence of Events:

1. The user has tapped on the “Add Goal” button.
2. App switches to the appropriate screen.
3. User fills out the required information and any optional information.
4. User clicks on the “Add” button to submit the information.
5. Server saves the information and sends a success message back to the application.
6. App switches to the user’s goals screen.

Alternatives: If the user has no wifi, the goal will be saved locally and updated on the server once they are connected to wifi.

Exceptions: If the server is experiencing an error, an error message is displayed to the user.

Use case ID: 10

Name: Delete Goal

pre-conditions: The user has an account and is a current gym member. The user has a stable wifi connection. The user has at least one goal.

post-conditions: The user will not see their deleted goal.

Initiated by: User

Triggering Event: The user has tapped on the “Delete Goal” button.

Additional Actors: N/A

Sequence of Events:

1. The user has tapped on the “Delete Goal” button.
2. App shows a pop-up and asks the user if they are sure they want to delete the goal.
3. User taps on the “Yes” button.
4. Server deletes the goal from the database and sends a success message back to the user.
5. App updates goal screen.

Alternatives: If the user does not have a stable wifi connection, the goal will be removed locally and updated on the server once they are connected to wifi.

Exceptions: If the server is experiencing an error, an error message is displayed to the user.

Use case ID: 11

Name: User Requests Workout Data

pre-conditions: The user is a valid gym member and is in the fitness center by a smart gym machine. The user’s smartphone has a strong enough wifi connection.

post-conditions: The user can see their workout results. If applicable, the user will be able to see their new/modified workout, if they reached any goal(s), and if they earned any reward(s).

Initiated by: User

Triggering Event: The user has clicked/pressed/tapped the appropriate button on the smart machine to request workout data.

Additional Actors: Smart Gym Equipment

Sequence of Events:

1. The user has clicked/pressed/tapped the appropriate button on the smart machine to request workout data.
2. The app and smart machine link up in some way such as bluetooth, QR code scanning, entering a temporary number, etc.
3. The smart machine sends the workout data for the user to see and to the server to process.
4. Server saves the workout session for the user.
5. Server determines if the user needs a new/modified workout. If it does, it calculates it and sends it back to the user to see. The workout is saved for the user.
6. Server adds the goal of reaching the new/modified workout for a specific length of time to the user's list of goals.
7. Server determines if the user reached any goal(s) and earned any reward from the workout data. If so, the app notifies the user.

Alternatives: N/A

Exceptions: If the server or smart machine encounters an error, the person will be alerted that there was an error in the appropriate system.

Use case ID: 12

Name: Sign-In

pre-conditions: The user is a current gym member and has an account on the app. The smart device the user is using has a stable wifi connection.

post-conditions: The user will be able to access their data.

Initiated by: User

Triggering Event: The user has entered their log-in information and clicked on the "Sign In" button.

Additional Actors: N/A



Sequence of Events:

1. The user has entered their log-in information and clicked on the “Sign In” button.
2. The request is sent to the server.
3. The server verifies the username and password..
4. The server verifies if the user is a current gym member.
5. The server sends a success message to the app.
6. The app changes to the home screen.

Alternatives: If the user has no wifi, they can still sign in locally but will not have access to every feature of the app. One of the pieces of information that will be saved locally is the membership expiration date, which will be used to verify the user’s membership when offline.

Exceptions:

If the server sees that the username or passport is not valid, it sends a message to the user stating that at least one of the fields is incorrect.

If the server sees that the user is not a current gym member, it sends a message to the user stating that the user does not have a valid gym membership.

If the server encounters any other error, it returns an error message.

Use case ID: 13

Name: Add Friend

pre-conditions: The user is a current gym member and has an account on the app. The other, different user has registered on the app. Both users' devices have a stable wifi connection.

post-conditions: The users will be able to see their friend's achievements, shared media, and level at any time.

Initiated by: Primary User

Triggering Event: The user has entered another person's username in the search tab to add a friend.

Additional Actors: Different (Secondary) User

Sequence of Events:

1. Primary user has entered another person's username in the search tab to add a friend.
2. App sends the request to the server with each character input.
3. Server compiles a list of gym members with a similar username and returns it.
4. App shows the list of people to the user.
5. User taps the correct person and taps on the "Send Friend Request" button.
6. Server sends the friend request to the secondary user.
7. The secondary user accepts on their own time.

Alternatives:

If no results are found after searching for people, the app shows no results, and the primary user has to search again.

If the secondary user is not a current gym member, the friend request will still be in their inbox.

Exceptions:

If the user has no wifi, they cannot search for people.

If the server encounters an error, an error message is returned to the primary user.

Use case ID: 14

Name: Delete Friend

pre-conditions: The user is a current gym member and has an account on the app. Their device has a strong enough wifi connection. The user has at least one friend.

post-conditions: The user will not see the other person's achievements, level, and media.

Initiated by: User

Triggering Event: The user has clicked on the "Delete Friend" button.

Additional Actors: N/A

Sequence of Events:

1. User has clicked on the "Delete Friend" button.
2. App shows a pop-up message asking if they are sure they want to delete that friend.
3. User taps on "Yes".
4. Server updates the user's friends list.

Alternatives: If the user taps on the "Cancel" instead of the "Yes" button, nothing in the server will change.

Exceptions:

If the user does not have at least one friend, they cannot access the “Delete Friend” button.

If the user does not have a stable wifi connection, they cannot delete any friend(s).

If the server encounters an error, the user will receive a message to try again later.

Use case ID: 15

Name: See Friends’ Achievements

pre-conditions: The user is a current gym member and has an account on the app. The user has a stable wifi connection.

post-conditions: The user will be able see their friends’ achievements, level, and media.

Initiated by: User

Triggering Event: The user has tapped on the “Friends” button.

Additional Actors: N/A

Sequence of Events:

1. User taps on the “Friends” button.
2. App sends a request to the server to see the user's friends' shared data.
3. Server sends the information to the app.
4. App changes the scene to the “Friends” homepage and displays the friends data.

Alternatives: N/A

Exceptions:

If the user does not have a stable wifi connection, they cannot see the “Friends” homepage and will see a message that tells them to try connecting to wifi to see their friends.

If the server encounters an error, a message will be sent and shown to the user that says to try again later.

Use case ID: 16

Name: See Gym Events

pre-conditions: The software has been set up for the gym; The gym employee has been trained to use it; The user or employee has a wifi connection; The user has an account.

post-conditions: The person will see all gym events.

Initiated by: User or Gym Employee

Triggering Event: The person clicked/tapped on the “Events” button.

Additional Actors: N/A

Sequence of Events:

1. The user clicked/tapped on the “Events” button.
2. App switches to the appropriate screen.
3. Server retrieves all the gym events and returns them.
4. Software displays the gym events.

Alternatives: If the person has no wifi, they cannot see the gym events.

Exceptions: If the server is experiencing an error, an error message is displayed to the person.

Use case ID: 17

Name: Add User to Gym Event

pre-conditions: The application has been set up for the gym; The gym employee has been trained to use it; The employee has a wifi connection; The user has an account.

post-conditions: The user will have the gym event added to their workout record.

Initiated by: Gym Employee

Triggering Event: The gym employee clicked/tapped on the “Events” button.

Additional Actors: User

Sequence of Events:

1. The gym employee clicked/tapped on the “Events” button.
2. Software switches to the appropriate screen.
3. Server retrieves all the gym events and returns them.
4. Software displays the gym events.
5. The employee asks the user for their username or name.
6. The user shares their username or name with the employee.
7. Employee searches for and adds the user to the event participation list.
8. Server adds the event to the user’s workout record.

Alternatives: If the employee has no wifi, they can still see the events stored locally, and they can add a user to the event, but it will not be updated in the server until they have a wifi connection.

Exceptions:

If the server is experiencing an error, an error message is displayed to the person.

If the user is not present, they cannot be added to the gym events.

Use case ID: 18

Name: See User Data

pre-conditions: User has registered for an account and is a current gym member. The user has a wifi connection.

post-conditions: User will be able to see their personal data.

Initiated by: User

Triggering Event: User taps on the "My Profile" or "My Workout History" button.

Additional Actors: N/A

Sequence of Events:

1. User taps on the "My Profile" or "My Workout History" button.
2. Server loads the data.
3. App displays the data.

Alternatives: If the user does not have wifi, they can only see data that is saved locally on their device.

Exceptions: If the server encountered an issue, it will return a message to the user saying that something went wrong when trying to retrieve data.

Use case ID: 19

Name: Uninstall App

pre-conditions: User has registered for an account and everything is up-to-date on the server.

post-conditions: Local data will be erased from the user's phone, but records will still be in the server.

Initiated by: User

Triggering Event: User taps on the button that uninstalls the app.

Additional Actors: N/A

Sequence of Events:

1. User taps on the button that uninstalls the app.
2. Locally-saved data will be erased from the device.

Alternatives: If the data is not up-to-date on the server, then that data will be saved to the server before the locally-saved data is erased.

Exceptions: N/A

## 2 Functional Requirements

### **F-1 - Setting Goals**

**Description:** The application must allow for users to set goals for themselves, like doing a specific workout or number of repetitions, which the app will use when keeping track of the user's exercise statistics.

**Rationale:** Being able to easily set and track goals is a great way for users to stay motivated as they can clearly see their progress, and strive for a specific goal in their fitness.

**Fit Criterion:** We will test that a user is able to set new goals, and that the completion status of these goals is automatically tracked by the application.

**Acceptance Tests:** Test 6 and Test 10

### **F-2 - Deleting Goals**

**Description:** Remove a specific goal the user has set before. The user can quickly delete it in the app by swiping the taskbar.

**Rationale:** The user may want to delete a goal for reasons such as the goal not being realistic, the user made a mistake, and so on.

**Fit Criterion:** We will test to see if the app still shows the goal in the server and user interface.

**Acceptance Tests:** Test 6

### **F-3 - Creating New Account**

**Description:** The user will set up their username, password, security question, email, age, sex, height, weight, and connect that data to the database by inputting



their unique identification number. They will also need to select if they want notifications for a workout reminder and the time.

**Rationale:** The user needs to be able to save their data into the server but need to first connect to it by having their username and password associated with their unique ID. Then, they need to fill out basic information before they do any workouts.

**Fit Criterion:** After the user logs out, they should be able to log back in and access their data as well as be able to request data from a smart machine.

**Acceptance Tests:** Test 1

#### **F-4 - Logging In**

**Description:** The user will input their username and password before tapping/clicking the “Sign In” button, which will load the home screen if their username and password is correct.

**Rationale:** The user needs to be able to access their data if they logout or need to log-in using another device.

**Fit Criterion:** The user will input the correct username and password before tapping/clicking on the correct button. The app will display the home screen.

**Acceptance Tests:** Test 2

#### **F-5 - Add Friend**

**Description:** The application should allow users to add other users as friends, which they can message and share information with, and they will be able to see their friends posts when browsing the social tab.

**Rationale:** Facilitating social interaction between users is another way of enhancing motivation and increasing user satisfaction with the app, as they can network with friends and find a sense of community and belonging in the app.

**Fit Criterion:** We will test that a test user account can add another account as a friend.

**Acceptance Tests:** Test 13

#### **F-6 - Delete Friend**

**Description:** The application should allow users to remove other users from their friends list, which means they will no longer be able to see that removed friend’s posts or information as well as not be able to message them.

**Rationale:** Users should be able to curate their friends list by both adding and removing friends.

**Fit Criterion:** We will test that a user that has added another as a friend can be removed from their friends list.

**Acceptance Tests:** Test 13

#### **F-7 - Checking-In**

**Description:** In the “My Workout History” tab, there will be a button called “Check-In” that the user will tap/click. The app will use the user’s location to determine if the user is in the gym, which will give the user a small amount of experience points/a reward if the location checks out.

**Rationale:** By checking in, the user will be given a little reward to give them some motivation to continue going to the gym.

**Fit Criterion:** The experience points should increase/reward dispersed and be updated in the server only when the device is inside the gym and is connected to wifi.

**Acceptance Tests:** Test 2

#### **F-8 - Requesting Workout Data**

**Description:** The application should be able to automatically (or manually) request and receive workout data from a smart equipment machine, which will forward it to the server.

**Rationale:** The application needs a way to interface with the smart equipment so that data about a user's workout can be retrieved for processing, display, and sending to the server.

**Fit Criterion:** We will test that after a workout on a given smart gym machine, the user’s application will be able to retrieve the workout statistics from that machine.

**Acceptance Tests:** Test 36

#### **F-9 - Load Account**

**Description:** Whenever the user switches screens or logs in, the server will return the data necessary to fill in the screen’s appropriate data fields.

**Rationale:** The user needs to have access and see their data in order to be able to decide which changes to make, see their progress, friends list, friend's achievements, demographics, goals, workout history, and so on.

**Fit Criterion:** We will test to make sure that every screen change displays the correct data. After every edit, the screen should display the updated data after leaving it and switching back to the screen.

**Acceptance Tests:** Test 36

#### **F-10 - Save Demographics**

**Description:** The user will be able to make changes to their demographics and select the "Save" button to have them applied.

**Rationale:** The demographics section needs to be saved only when the user is ready to save it.

**Fit Criterion:** After changing something in the demographics section and selecting "Save," we will switch the screen and return to the demographics section to see if the data loaded is correct, which reflects the state of the database.

**Acceptance Tests:** Test 4

#### **F-11 - Add Media**

**Description:** The person will select the "Add Media" tab. They will select under which area they want to upload the video or photo under. After adding a photo or video of valid size (5-20MB for photos, less than 1GB for videos, employees can upload larger media), and giving the media a title and description, it will be compressed before being sent to the server.

**Rationale:** So that users are able to share media with their friends, and employees can upload media showing members how to perform an exercise safely and correctly.

**Fit Criterion:** We will test that a normal user is able to upload several images under 20MB, and upload videos under 1GB.

**Acceptance Tests:** Test 34

#### **F-12 - Exercise Notification**

**Description:** According to the user's chosen schedule and form of notification, if they have not been to the gym by the time of their next scheduled workout, the app will remind them to go to the gym.

**Rationale:** To encourage the user to continue exercising and keep them accountable for their goals. It also helps them maintain a healthy habit.

**Fit Criterion:** Various different times and dates will be chosen. We will make sure the notifications are triggered at the right times. Testing will also be done for edge cases like when daylight savings goes into and out of effect.

**Acceptance Tests:** Test 14

#### **F-13 - See Friends' Achievements**

**Description:** The user should be able to view their friends achievements through the social tab of the app.

**Rationale:** To promote social interaction and enhance the motivation of our users, they will be able to see their friends achievements. This will also provide a satisfying sense of accomplishment for the users that complete achievements, because their friends can see the results of their hard work.

**Fit Criterion:** We will create a test user with a set of friends who have completed various individual achievements, and verify that the test user can view their friends achievements.

**Acceptance Tests:** Test 13

#### **F-14 - See Gym Events**

**Description:** The user should be able to see the events currently ongoing or soon planned to happen in their Gym, so that they can possibly attend them. Users can see if the machines are being used in the app to better organize their workout schedule.

**Rationale:** To enhance the integration between the Gyms and the application, users will be able to see ongoing events taking place in the Gym, to facilitate participation and provide user's a reason for attending through rewards/achievements.

**Fit Criterion:** We will create a test event before making sure that a user is able to see the test event on their application.

**Acceptance Tests:** Test 14

#### **F-15 - Set Gym Goals/Events**

**Description:** Gym employees should be able to set gym-wide goals and start events for the gym.

**Rationale:** To give users more activities and goals to meet other than those personalized for themselves, and to also give more motivation to users as the gym goals have unique rewards once completed.

**Fit Criterion:** We will test that a test employee account will be able to create goals and events for the entire Gym.

**Acceptance Tests:** Test 15

#### **F-16 - Add New Member**

**Description:** The gym employee will generate a new unique ID for the new gym member to use for their account creation. The database will create a new row for the user's demographic, login information, membership status, and membership expiration date.

**Rationale:** This will allow the gym member, and not any random person, to register for an account and start using the app. The membership status and expiration date added will give or deny permission to the user.

**Fit Criterion:** We will add a new user using the employee version of the software and make sure the correct fields in the database are created and filled out.

**Acceptance Tests:** Test 1

#### **F-17 - Add User to Event**

**Description:** For a gym event, a gym employee will check-in the member into the event, which will be added to the user's workout history.

**Rationale:** Having the gym employee personally check-in the member makes sure that the member actually went to the event. Meanwhile, it adds the workout to the user's history and level.

**Fit Criterion:** We will make sure that only an employee can add a person to the gym event and that the person can see the workout in their workout history.

**Acceptance Tests:** Test 35

### **3 Data Requirements**

#### **D1 - Length of Username**

**Description:** The username should be at least 8 characters long but no longer than 30 characters.

**Rationale:** The username should be easy to remember and for an employee to manage, but it should also be long enough to be secure.

**Fit Criterion:** Upon submission of a username, the length will be checked before adding it to, or updating, the server.

**Acceptance Tests:** Test 11

## **D2 - Complexity of Passwords**

**Description:** The length of password should be at least 8 characters and should not be longer than 16 characters. The password should have at least one lowercase letter, uppercase letter, number, and special character.

**Rationale:** The password should have some basic security features meaning that it should not be too short, and should have appropriate complexity to make it harder to crack, however, it should also not be so long such that people cannot remember it or can be used for attempted exploits.

**Fit Criterion:** Upon submission of a password, the length and contents will be checked to make sure it fulfills the complexity standards before adding it to, or updating, the server.

**Acceptance Tests:** Test 12

## **D3 - Workout History**

**Description:** The workout history record should contain: the type of exercise, date the exercise took place, duration of the exercise, and amount of calories burned from that exercise. If the workout can be associated with data from sensors, it will be included as well.

**Rationale:** The users must be able to see their workouts from the past. Also, the history of the user plays a role in calculating a new/modified workout, which takes place in the server.

**Fit Criterion:** Exercises with varying information will be added to an account, then after logging out and back in, the history with the correct information should still be there.

**Acceptance Tests:** Test 31

## **D4 - User Demographics**

**Description:** Will consist of: name, birth date, sex, weight, and height.

**Rationale:** The basic information will help the server calculate new or modified workouts for the user, and also the estimated calorie consumption for a given exercise. The name and a few other fields can help the communication between gym employees and users if there is an issue.

**Fit Criterion:** The fields should be filled out before a workout and appear in the database.

**Acceptance Tests:** Test 31

#### **D5 - User Goals**

**Description:** A single goal must have at least the exercise it concerns, and the target weight/speed/time/repetitions/sets/etc. An optional field is how long they have to reach it.

**Rationale:** The more detailed a goal, the more likely a user is to work for it. Also, it gives the user something to work for.

**Fit Criterion:** The goals should be saved in the server and be retrieved when loading the appropriate screen.

**Acceptance Tests:** Test 6

#### **D6 - Friends' Achievements**

**Description:** The user will be able to see (if allowed) their friends' level, goals reached, and media posted.

**Rationale:** Seeing the level, accomplished goals, and media of others motivates people to work hard to catch up or give them something to share as well.

**Fit Criterion:** Upon the post of a friend, the user should be able to see it anytime they go to the "Friends" tab or scroll up the history.

**Acceptance Tests:** Test 13

### **4 Performance Requirements**

#### **Speed and Latency Requirements**

##### **P1 - Speed and Latency**

**Description:** The application should provide a responsive and low latency experience for the user. The average time for a request to be processed and a response sent by the server should be no more than 200 milliseconds, providing the user is in a participating gym and using their members only wifi.

**Rationale:** To have a smoothly operating application that does not frustrate users due to poor performance or slow response times.

**Fit Criterion:** We will make a large amount of test requests of all types and sizes to the server, and measure the average response time, to see if it is under 200 milliseconds. We will also have 20 test users come in and rate the responsiveness of the application on a scale from 1-10. We will consider this test passed if the average score is greater than 7.

**Acceptance Tests:** Test 20

## **Precision or Accuracy Requirements**

### **P2 - Precision**

**Description:** The system must be able to accurately perform the calculations using the data from the smart equipment and the users demographic information. All calculations must be accurate up to the 4th decimal place.

**Rationale:** The application needs to accurately perform the calculations so as to give the users the most accurate results and information about their exercise statistics. Users might use the calories expended calculations run by the application for use in their dieting/weight loss goals, and as such a high degree of accuracy is required.

**Fit Criterion:** We will add several sets of workout data to a test user account with a given height, weight, sex, age and resting heart rate, and we will test that the result of all calculations is accurate up to the 4th decimal point.

**Acceptance Tests:** Test 33

## **Capacity Requirements**

### **P3 - Capacity**

**Description:** The application must be able to support up to 2000 simultaneous users, with possible spikes to ~3000 during peak times like 6:00-8:00 A.M., and 5:00-6:00 P.M.

**Rationale:** We expect to have a minimum of 10 gyms for the initial adoption of the application, with an expected maximum capacity of each gym around 200-300 people. The system needs to be able to properly function with at least 2000-3000 concurrent users using the application.

**Fit Criterion:** We will use automated testing to create 300 test accounts from 10 different Gym locations, who will each make concurrent requests to the server through the application for a 4 hour testing period.



**Acceptance Tests:** Test 20

## **5 Dependability Requirements**

### **Reliability Requirements**

#### **DP1 - Frequency of Failure**

**Description:** The application should suffer failures like a shutdown no more than a few times a year at most.

**Rationale:** It is our responsibility to make sure that users are satisfied with the application and quality of service we provide, and as such down-time should be minimized so that a high-quality experience can be provided as often as possible. As such, the service should be kept running stably, so as to not to affect users' workout routine, this should also be satisfied.

**Fit Criterion:** We will measure the number of server shutdowns and over a period of time to see if it meets our standards.

**Acceptance Tests:** Test 19

#### **DP2 - Solution of Emergency**

**Description:** There should be at least one IT person on call to fix server shut down and any other IT issues. They will be available from 8:00 A.M. to 6:00 P.M.

**Rationale:** To make sure the application can be back up and running as soon as possible for the users once an emergency arises.

**Fit Criterion:** We will measure the average response time and time for emergencies to be resolved.

**Acceptance Tests:** Test 26

#### **DP3 - Time Period Needed to Update**

**Description:** The server can only shut down to update the app between 12 A.M. and 5 A.M.

**Rationale:** The app needs to be available while the fitness center is open.

**Fit Criterion:** When the server is shut down to update, it is back up by 5 A.M.

**Acceptance Tests:** Test 28

## **Availability Requirements**

### **DP4 - Running Time**

**Description:** The app should be able to run 24 hours a day and 7 days a week without any problem in general. There should be a backup server ready to run in case of an emergency in the primary server.

**Rationale:** We want to make sure users are able to use the app anywhere, anytime.

**Fit Criterion:** The app should be accessible and working at all times of the day and any day of the week.

**Acceptance Tests:** Test 28

## **Robustness or Fault-Tolerance Requirements**

### **DP5 - Offline Mode**

**Description:** There should be an offline mode where the user can still access their workout data and temporarily store the current data in their local device. Once connected to stable wifi, the data will be updated on the server.

**Rationale:** Not everyone will have a stable wifi connection while working out, but they should still be able to access some of the application's features so as to provide an alternative that is able to seamlessly transition back to fully functioning once a wifi connection is secured.

**Fit Criterion:** Users are able to access the workout data offline.

**Acceptance Tests:** Test 29

### **DP6 - Emergency Contact**

**Description:** The application should support error checking to detect when there is a fault or failure. If such a fault or failure is detected, there should be an emergency phone number or other contact information displayed for users to provide a report or receive help.

**Rationale:** It allows the IT team to make sure the app goes back to work as soon as possible as they can receive reports.

**Fit Criterion:** Using a simulation of the app encountering an error, ensure that the contact information appears.

**Acceptance Tests:** Test 26

## **Safety-Critical Requirements**

### **DP7 - Data Leakage**

**Description:** There should be zero user data leakage. If this is not possible, it should be kept to a minimum, and the users immediately notified so that they can update passwords/etc to protect their privacy.

**Rationale:** To protect users' privacy. The information including address, phone number, and demographic information should not be exposed to unauthorized individuals.

**Fit Criterion:** Test that in a simulated data breach, the information received is still encrypted, and that the affected users receive a notification.

**Acceptance Tests:** Test 16

## **6 Maintainability and Supportability Requirements**

### **Maintenance Requirements**

#### **M1 - Solid Developer**

**Description:** After the app is released, there should be one to two developers left to maintain the application.

**Rationale:** There needs to be a skilled developer ready for app maintenance at any time to support a quality user experience. This makes future development and debugging quicker.

**Fit Criterion:** The debugger should write a report talking about how fast they fix issues and what the result is.

**Acceptance Tests:** Test 24

### **Supportability Requirements**

#### **M2 - Feedback**

**Description:** There should be a contact option in the application for users and gym employees to share feedback/submit bug reports for review.

**Rationale:** To improve the quality of the service, and because there will inevitably be bugs and oversights that we do not catch in our pre-release testing. This feedback can then be used to help guide our bi-monthly bug fixing updates.

**Fit Criterion:** Test that there is a working feedback system where users can submit bug reports and give feedback.

**Acceptance Tests:** Test 8

## **Adaptability Requirements**

### **M3 - Platforms**

**Description:** The app should be able to work on the latest and most popular versions of Android.

**Rationale:** We want the app to work with Android versions that are the most popular with people, so there can be a wider range of availability. The latest versions cannot be ignored since there are quite a large number of people that have it as well.

**Fit Criterion:** The functionality of the app should be tested in a phone with different versions of the system to make sure that the app functions properly.

**Acceptance Tests:** Test 25

## **Scalability or Extensibility Requirements**

### **M4 - Server**

**Description:** There should be enough backup servers ready to be used when the capacity requirements grows or when there is a sudden uptick in activity.

**Rationale:** The application must be prepared in case the servers become overworked when business and number of users grows.

**Fit Criterion:** There must be a suitable amount of extra servers, such that we have a number of backup servers equivalent to no less than 10% of our normally active servers.

**Acceptance Tests:** Test 21

## **Longevity Requirements**

### **M5 - Lifetime**

**Description:** The lifetime of the app should be at least the same as the gym, or as long as the Gyms maintain our contract/desire to keep using the application.

**Rationale:** As long as the gym is in business and our contract requires it, the app must run to retain their membership levels.

**Fit Criterion:** Ensure there is a maintenance team updating the app during the duration the gym is open.

**Acceptance Tests:** Test 22

## **7 Security Requirements**

### **Access Requirements**

#### **S1 - User Data Access**

**Description:** A user's data should be only accessible to the user themself.

**Rationale:** User data privacy is extremely important, and unless explicitly shared by the user, sensitive personal information should not be public for all to see.

**Fit Criterion:** The requirement would be tested by creating test accounts and attempting to access a designated account (by using another test account) to determine whether any unauthorized users were able to gain access to any sensitive data from the designated account (including attempting to login with the designated username, but a wrong password, and attempting the forgot password recovery).

**Acceptance Tests:** Test 23

### **Integrity Requirements**

#### **S2 - Database Protection**

**Description:** The application and server should protect against SQL injection attacks.

**Rationale:** SQL injection attacks are one of the most common ways for malicious actors to attempt to gain access, corrupt, or damage an SQL database. Protection against these attacks is critical to protect user data, and ensure the integrity of the database for normal intended operation of the application.

**Fit Criterion:** We will test the safety of the database by attempting a multitude of SQL injection attacks, like attempting to drop tables, access a user's data, or modify the contents of one or many instances/tables, and accept if the integrity of the database is preserved by not executing any of the malicious queries.

**Acceptance Tests:** Test 27

### **Privacy Requirements**

#### **S3 - Data Confidentiality**

**Description:** User passwords and sensitive data should be encrypted.

**Rationale:** The server/database will contain a vast amount of sensitive user data, alongside passwords that users may use for multiple services. If this information were to leak in the event of a data breach or other form of malicious entry, it would be a huge security risk for the users affected.

**Fit Criterion:** Test through a simulated database breach/dumping the state of the database during normal operation if there is any sensitive information left encrypted/stored in plain text.

**Acceptance Tests:** Test 23

## **Audit Requirements**

### **S4 - Membership Audit**

**Description:** The application will keep a log of user membership status which may be audited at any time.

**Rationale:** In case of disputes about user membership and access to the application, there will be a detailed log accessible by audit to maintain a paper trail.

**Fit Criterion:** We will create several test accounts, adding and removing membership status, and then test whether the membership status audit log properly contains all membership status changes/updates.

**Acceptance Tests:** Test 3

## **Immunity Requirements**

### **S5 - Anti-Virus**

**Description:** Both the user application running client side and the server must have antivirus software.

**Rationale:** Malicious software that has access to a user's device may compromise the application, and lead to vulnerabilities for both the individual user and the server. The server itself may also be attacked by malicious software, and as such both the client side user application and the server need to be running an antivirus.

**Fit Criterion:** Check that the installed antivirus software is working and up to date.

**Acceptance Tests:** Test 9

## 8 Usability and Humanity Requirements

### Ease of Use Requirements

#### U1 - Ease of Use

**Description:** Users should be able to become familiar with every functionality within 10 minutes.

**Rationale:** To not lose motivation or become frustrated with the application, it must be easily navigable. It should be kept as simple as possible for older users who are not familiar with using smartphones.

**Fit Criterion:** Have a test set of 50 users of varying ages and backgrounds test the application, and rate its ease of use on a scale of 1-10. If the average score is greater than 7, we will consider this requirement fulfilled.

**Acceptance Tests:** Test 19

### Personalization and Internationalization Requirements

#### U2 - Personalization

**Description:** Users should be able to change the language in the application, between light and dark themes, as well as activate a colorblind mode for users with visual impairments. Text size should also increase if the user increases it in their smartphone's settings menu.

**Rationale:** To give users more opportunities to customize and personalize their experience with the application, and also to provide a more quality experience for users with visual impairments.

**Fit Criterion:** We will test that we can change the language, theme, colorblind mode, and text size in the application.

**Acceptance Tests:** Test 17

### Learning Requirements

#### U3 - How to Use

**Description:** Users should be able to understand how to use the application after an instructional demonstration by gym staff when they first receive it.

**Rationale:** The application should be intuitive enough for users of all technological backgrounds that after a short demonstration they will be able to use the app.

**Fit Criterion:** We will have a group of 30+ test users who will receive a briefing on how to use the application. Then, we will instruct them to perform a series of in-app tasks, if ~90% of the test users are able to complete the tasks within 10 minutes we consider this a requirement fulfilled.

**Acceptance Tests:** Test 19

## **Understandability and Politeness Requirements**

### **U4 - Tutorial**

**Description:** The application should have a tutorial guiding the user to browse all functionalities. After the tutorial, users should be able to know how to check-in, set goals, how to retrieve their workout data from a smart machine, virtually socialize with friends, and track their records.

**Rationale:** Users may become dissatisfied if it takes a long period of time to learn how to use it. We aim to make users not lose interest in using the app and exercising.

**Fit Criterion:** Obtain feedback from users where 90% of them do not complain that the application is too hard to understand.

**Acceptance Tests:** Test 19

## **Accessibility Requirements**

### **U5 - Accessibility**

**Description:** The application should allow users with disabilities to have an equal experience. There will be a section for individuals with disabilities, where they can view extra videos and get additional information.

**Rationale:** Users with disabilities may have difficulties with certain exercises, and as such we will offer alternatives, and additional videos and information to accommodate them.

**Fit Criterion:** We will have legal expert(s) review our application to see if it is in compliance with the Americans with Disabilities Act.

**Acceptance Tests:** Test 7

## **User Documentation Requirements**

### **U6 - Help**

**Description:** There should be an easily accessible “Help” section of the app for providing answers to users in case that they have confusion about the app.



**Rationale:** It is important that users be able to find the appropriate help and information they need to use the application.

**Fit Criterion:** A help menu should be visible in the app.

**Acceptance Tests:** Test 17

## **Training Requirements**

### **U7 - Training for Employee**

**Description:** Employees need to be trained in how to use the application, and how to provide basic technical assistance to other users. They also have to know how to obtain the unique ID for gym members.

**Rationale:** To help improve the user experience for the customer and client, as users may feel more comfortable asking for help from a Gym employee rather than trying to navigate the app themselves.

**Fit Criterion:** All employees are able to tell the exact location of an app feature within a time span of about 5 seconds or less.

**Acceptance Tests:** Test 18

## **9 Look and Feel Requirements**

### **Appearance Requirements**

#### **L1 - Font and Color**

**Description:** The font and color palette should be simple and readable. The gym logo should be placed such that it is clearly visible, but not a distraction.

**Rationale:** The logo represents the gym and so it should be placed front and center, but in such a way that it won't negatively impact the user experience. The font and color palette should not be distracting or tire the eyes when stared at for too long.

**Fit Criterion:** App testers will make sure the visuals are fine to stare at for at least 5 minutes, and we will verify that they don't feel the Gym's logo is too distracting.

**Acceptance Tests:** Test 17

### **Style Requirements**

#### **L2-Publicity**

**Description:** There should not be advertisements in the application since the users are already paying for it through their membership. The overall appearance should be clear, clean, and organized. The app's icon should not be crowded.

**Rationale:** The publicity and appearance are one of the things that determine the success and user-friendliness of an app.

**Fit Criterion:** App testers will make sure the app is not overwhelming, and that they find the visuals appealing and not confusing.

**Acceptance Tests:** Test 17

## **10 Operational and Environmental Requirements**

### **Expected Physical Environment**

#### **OE 1 - Smart Machines**

**Description:** The gym will have several pieces of smart exercise equipment that can interface with the application for features such as automatic workout statistic recording and entry.

**Rationale:** The application is designed around and operates best when used in the gym with supported pieces of smart exercise equipment, as that provides the most convenient and highest quality experience for the users.

**Fit Criterion:** We will test that each gym has sufficient smart exercise equipment to service their peak daily users (at least one piece of smart equipment per user during peak hours).

**Acceptance Tests:** Test 30

### **Requirements for Interfacing with Adjacent Systems**

#### **OE 2 - Smart Machine Support**

**Description:** The application needs to be properly integrated with the smart equipment on-site at participating gyms, so that information such as workout statistics can be automatically recorded and entered into the application.

**Rationale:** For the intended experience, the application needs to be as convenient as possible for the user, so that they can focus on their actual activities rather than laborious manual entering of workout statistics. This will require the application to be able to interface and work seamlessly with the provided smart gym equipment. The user's physical exercise will be recorded by the smart machine, and will be sent via wifi to the user's device. This will have a polling rate of about

once every 5 minutes, or when the user leaves the machine, and it will send relatively small megabyte sized packets to the user's device.

**Fit Criterion:** We will test this by checking if the application supports at least 90% of the smart exercise equipment utilized by the gyms using the application, and if the information sent by the smart machines is properly received by the application on the user's device.

**Acceptance Tests:** Test 30

## **Productization Requirements**

### **OE 3 - Distribution**

**Description:** This product will be distributed via the app store on the user's device. By using their unique ID given when becoming a member, they will be able to create an account and log in.

**Rationale:** Every user has some app store installed by default on their device, and as such it will be the most easily accessible way for the application to be distributed. Anyone can install the application, but only current gym members will be able to access it via creating an account or logging in.

**Fit Criterion:** We will test that the application can be acquired through the app store, and that a simulated paying member can gain access to the application, whereas someone without a membership cannot.

**Acceptance Tests:** Test 25

## **Release Requirements**

### **OE 4 - Release Cycle**

**Description:** The application will have regular bi-monthly security, maintenance, and/or bug-fixing updates alongside a twice yearly new content release cycle which adds to the functionality or options of the application.

**Rationale:** To ensure the quality operation of the application and the best user experience, we will work to have an accelerated maintenance release schedule that quickly fixes any issues that may arise. Bi-annual content updates will also help to keep users engaged and stay as paying members.

**Fit Criterion:** We will meet this requirement by ensuring that we have bi-monthly maintenance releases, and a content release bi-annually.

**Acceptance Tests:** Test 22

## 11 Cultural and Political Requirements

### Cultural Requirements

#### CP1 - Cultural

**Description:** The application should use inclusive, welcoming language, and be supportive of users' efforts. The application should respect the diversity of its users through avoiding any discriminatory or offensive language regarding users characteristics.

**Rationale:** We must use very careful language when talking about a user's weight, body, health and fitness, because this can be a very sensitive topic for many people. Avoiding discriminatory or offensive language is also imperative to creating the welcoming environment we seek to establish as we support our diverse group of users.

**Fit Criterion:** We will have our application reviewed by a professional equality and diversity officer to see if it is inclusive and respectful of the diversity of our users.

**Acceptance Tests:** Test 32, Test 7

### Political Requirements

#### CP2 - Client Support

**Description:** As part of our negotiated contract with the gyms that provide the funding for us developing the software, we are to continue development and support of the application for at least 3 years. To make the client gyms satisfied we will offer bi-monthly maintenance updates, twice yearly content updates, and on-site support for the gym-hired tech team(s) in the event of outages or technical issues.

**Rationale:** To keep our clients happy we will continue to provide support and content updates for several years, so that the clients feel they are receiving a worthwhile return on their investment. Although technical support is not part of the contract, we will also work with the gym hired tech support to help keep the clients happy with the smooth operation of the application.

**Fit Criterion:** We will conduct regular surveys/meetings with our clients to gauge their level of satisfaction with our work.

**Acceptance Tests:** Test 22, Test 19

## 12 Legal Requirements

### Compliance Requirements

#### LE1 - Law

**Description:** The app should not violate any law issued by the government in the countries in which it is to be deployed. For example, the app should not have videos with nudity, violence or anything else that would violate obscenity laws.

**Rationale:** The application must follow the laws and regulations of the countries in which it will operate.

**Fit Criterion:** The app has to be examined by a legal team before it is released on the app store.

**Acceptance Tests:** Test 16

### Standards Requirements

#### LE2 - Company & Gym Standards

**Description:** The app should not violate the rules created by the software company and gym. It should also meet the standards for applications appearing on the Android app store, and receive an “E” rating. We will develop this software in accordance with the IEEE Code of Ethics.

**Rationale:** We must make sure the app will not violate any content rules, and that it is able to be released with an appropriate rating on the app store. Our software development team will operate under the principles specified in the IEEE Code of Ethics.

**Fit Criterion:** Before the app is released, we should work with the legal team to ensure that the standards of the app store are met, and we will also review the application to see if it is in compliance with our standards as specified in the IEEE Code of Ethics.

**Acceptance Tests:** Test 16

# 13 Requirements Acceptance Tests Requirements – Test Correspondence Summary

Test	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Test 11	Test 12	Test 13	Test 14	Test 15	Test 16	Test 17	Test 18	Test 19	Test 20	Test 21	Test 22	Test 23	Test 24	Test 25	Test 26	Test 27	Test 28	Test 29	Test 30	Test 31	Test 32	Test 33	Test 34	Test 35	Test 36	
Functional Requirements	F1									X																											
	F2					X																															
	F3	X																																			
	F4		X																																		
	F5												X																								
	F6												X																								
	F7		X																																		
	F8																																				X
	F9																																				X
	F10			X																																	
	F11																																	X			
	F12													X																							
	F13													X																							
	F14													X																							
	F15														X																						
	F16	X																																			
	F17																																				X
Data Requirements	D1										X																										
	D2											X																									
	D3																																X				
	D4																															X					
	D5					X																															
	D6												X																								
Performance	P1																			X																	
	P2																																	X			
	P3																			X																	
Dependability Requirements	DP1																		X																		
	DP2																									X											
	DP3																											X									
	DP4																											X									
	DP5																												X								
	DP6																									X											
	DP7																X																				
Maintenance & Support	M1																																				

Table 3 - Acceptance Tests Correspondence

## **Acceptance Test Descriptions**

### **1 - Account Setup/Adding New Members**

**Description:** Test that once an employee has added a user as a member in the database with a unique ID registered, the user is able to create an account, which is then registered on the server if the username is unique, otherwise prompt the user to use a different username.

### **2 - Logging In/Receiving Check-In Reward**

**Description:** Test if a user is able to log in to their existing account using their username and password, and if they are able to receive their daily check-in reward after selecting the “Check-In” button and being in a valid location.

### **3 - Membership Status**

**Description:** Test that a user’s membership status is correctly observed; Users whose membership expires lose access to their account while those with a membership are granted access.

### **4 - Manual Entry of User Data**

**Description:** Test that if a user manually enters data (exercises not from smart machines, demographic information, etc.), it is correctly recorded on the server and displayed in-app in the appropriate screens.

### **5 - Automatic Entry of User Data**

**Description:** Test that user exercise data from a smart machine is correctly recorded on the server and displayed in-app when automatically processed by in-Gym machines interfacing with the app.

### **6 - Set/Remove Goals**

**Description:** Test that the user is able to set their own exercise goals if they have filled out all the required fields and data requirements. Also, test that the goals can also be removed. Make sure these modifications are updated on the server.

### **7 - ADA Compliance**

**Description:** Test that the application is in compliance with the Americans with Disabilities Act.

### **8 - Feedback System**

**Description:** Test that the user is able to send in feedback/submit bug reports through the in-app feedback system.

## **9 - Server Antivirus**

**Description:** Test that the server is running an up-to-date version of its antivirus software and properly functioning.

## **10 - Complete Goals**

**Description:** Test that the user is able to automatically complete goals when reaching the required threshold, and receive appropriate rewards.

## **11 - Username Length**

**Description:** Test that the user cannot make their username longer than 30 characters when attempting to register.

## **12 - Password Complexity**

**Description:** Test that the user's password is of appropriate length and complexity (between 8 and 16 characters, including at least one of each of the following: one uppercase letter, one lowercase letter, one number, and one special character).

## **13 - Add/Remove/View Friends**

**Description:** Test that the user can add new friends, remove friends, and view their list of in-app friends along with their achievements, socials and posts. Note that the friends must also be registered gym members.

## **14 - Notification System**

**Description:** Test that the user receives a notification when an in-gym event is happening, or scheduled to happen, or when a gym-wide goal is set. Also, test that users receive a notification when they haven't been to the gym and/or completed their scheduled workout yet. Test whether the app can properly display the usage status of fitness equipment.

## **15 - Gym Event/Goals Creation**

**Description:** Test that authorized users (such as gym employees) can create gym events and/or goals in-app, which users will then receive a notification for (see acceptance test 14).

## **16 - Legal Check**



**Description:** Test that the app does not violate the local laws and should meet the requirements of safety.

### **17 - User Interface (UI)**

**Description:** Test that the color palette of the app is not harsh on the eyes, the font is readable, the myGymClub logo is on the UI, and there are no advertisements. Test that functionality mentioned on the app. Test that users are able to change fonts, themes, and colorblind mode. Test that the user interface is displayed as designed on the selected devices. Test drop-down boxes and animations to see if they are working properly.

### **18 - Employee Training**

**Description:** Test that the employees who are supposed to help users with using the app are familiar with the functionality of the app.

### **19 - Feedback and Survey**

**Description:** Test that functionalities work by obtaining feedback from users doing some questionnaire surveys or through the feedback system.

### **20 - Performance and Capacity**

**Description:** Test that the average time for a request to be processed and a response sent by the server should be no more than 200 milliseconds, and that up to 3000 concurrent users can be supported by the application.

### **21 - Backup**

**Description:** Test that there are enough new servers in the warehouse and are ready to be used when there is an emergency.

### **22 - Lifetime and Release Cycle**

**Description:** Test that the lifetime of the app is the same as the gym by making a contract, and that, for the extent of the application's lifetime, it receives regular bi-annual content updates, and bi-monthly bug fixes.

### **23 - User Data Security**

**Description:** Test that a user's data is accessible only to themselves, that usernames and passwords are encrypted in the database, and that there is no data leakage from the server/database.

### **24 - App Developer**

**Description:** Test that the app is able to be updated on time by experienced developers.

### **25 - Platform and Distribution**

**Description:** Test that the app is able to run on valid versions of Android, and that the app appears on the app store. Make sure that users without a gym membership are unable to gain access to the application beyond the login screen.

### **26 - Emergency**

**Description:** Test that the app shows contact information of the IT department when the app shuts down, and the tech personnel is able to pick up the phone call or reply to emails.

### **27 - Database Security**

**Description:** Test that the database on the server is protected against SQL injection attacks and other forms of malicious code.

### **28 - Serve & Update Time**

**Description:** Test that the app is only updated in the time period between 12 A.M. and 5 A.M., and that users are able to use the app 24/7 when the app is going to update.

### **29 - Offline Mode**

**Description:** Test that there is an offline mode in the app when the device does not have an internet connection where users are still able to access their workout data.

### **30 - Physical Location and Integration**

**Description:** Test that the gyms have smart equipment, and that the application can interface with and pull data from at least 90% of the equipment employed by the Gyms.

### **31 - Personal Information**

**Description:** Test that the app keeps track of users' workout history and demographic information under their profile.

### **32 - Cultural**

**Description:** Test that the application is respectful and inclusive of the diversity of our users by having it reviewed by a professional equality and diversity officer, and receiving a satisfactory review.

### **33 - Precision**

**Description:** Test that the result of all calculations is accurate up to the 4th decimal point.

### **34 - Uploading Media**

**Description:** Test that users are able to upload images under 20MB, and upload videos under 1GB, and that said media is received and properly compressed.

### **35 - Add User to Event**

**Description:** Test that Gym employees are able to add users to events, and that users can see that they have been marked as attending the event under their workout history.

### **36 - Request Data From Server**

**Description:** Test that users are able to properly receive workout, demographics, etc. data from the server when requested by the application.

## **III Design**

### **1 Design Goals**

- Scalability
  - The application should be able to handle expanding operations to a growing number of Gyms, so as to allow us to increase our client base without compromising on the performance and integrity of the application. While the initial number of Gyms adopting the application might be small, the system should be designed to grow and handle the requirements of a much larger user base if need be.
- Minimal Saved Records in Database
  - A single gym alone can have multiple customers, and data can be quickly used up in the database. Having the least amount of workout logs is one of the factors that can cause a rapid use of data. Therefore, we would like to save the least amount of necessary data and be smart with the setup of the database.
- Ease of Use
  - The age range of gym members is large. To appeal to all age groups, having the application be as simplified and straightforward as possible

would help give a satisfying experience to every member using the application.

- Fault Tolerance
  - The application should be able to handle unexpected conditions and faults occurring without the entire system being compromised. For example, in the event of a server outage, the application should be able to handle this and possibly distribute the load to other servers, such that user's are still able to use the application in some capacity (rather than having to have a hard reset/application downtime while the issue is fixed, preventing any user access)

## **2 Current System Design**

Should a client already have their own database that contains their customers' accounts and/or workout-related information and want to continue using past information, this project would have to map pre-existing data to the database the project will set up.

The current system used by the Gyms would be to just provide workout machines and items like weights, etc., to their members, without an accompanying application. Their system may involve some server or database of user records, or it may even just be stored locally on the owner's computer.

## **3 Proposed System Design**

### **3a Initial System Analysis and Class Identification**

Our analysis resulted in realizing the main entities of the project: the gym member, the administrator, the server, the database, and the application.

The administrator's job is to create gym events, create gym goals, and update the membership status of each person connected to the gym. The server uses this data to make connections in the database for the gym member. It also stores all the data in the application for each member, which consists of personal demographics, workout data, records, goals, achievements, and posts.

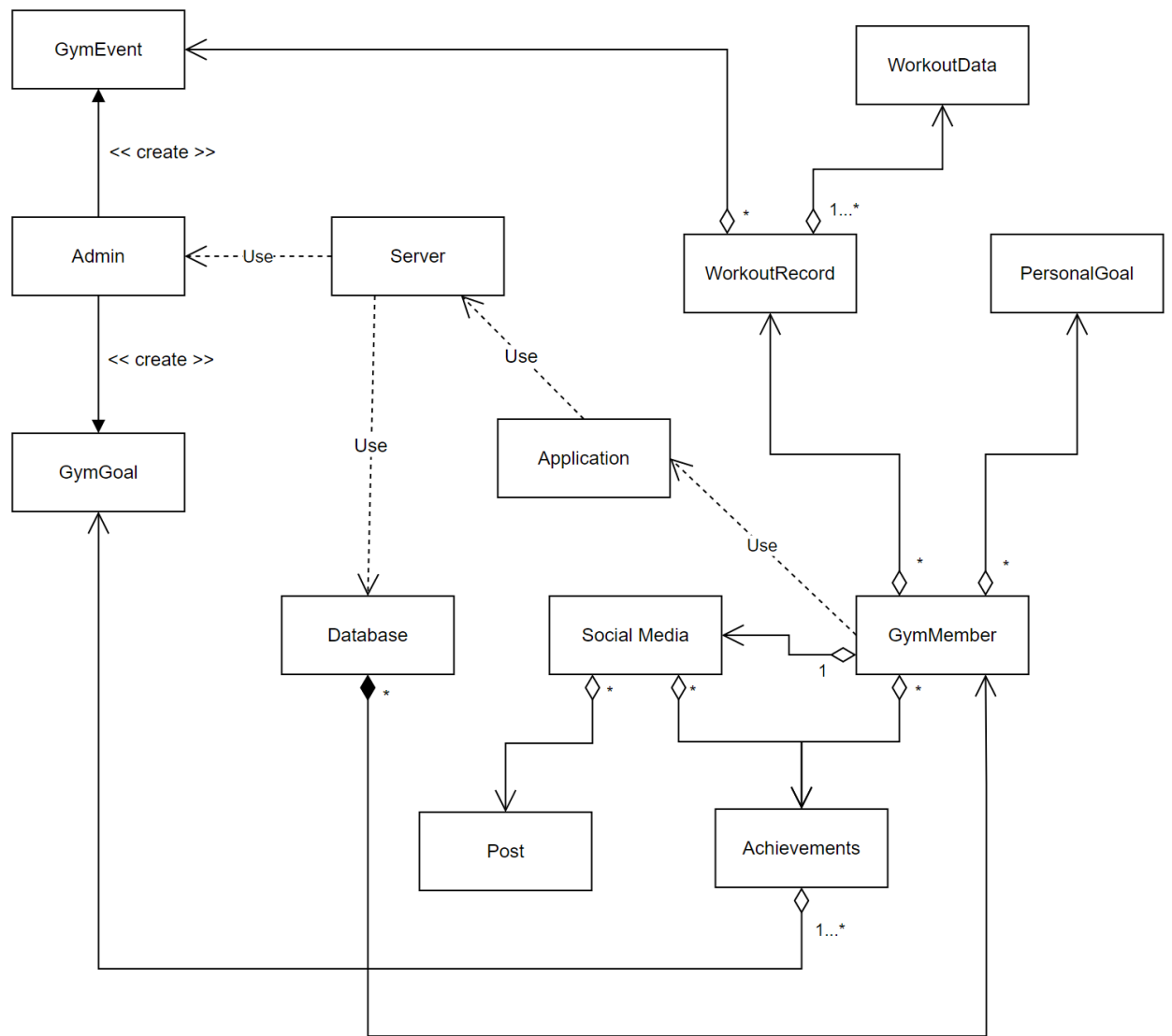


Figure 7 - Initial Overall Class Diagram

### 3b Dynamic Modeling of Use-Cases

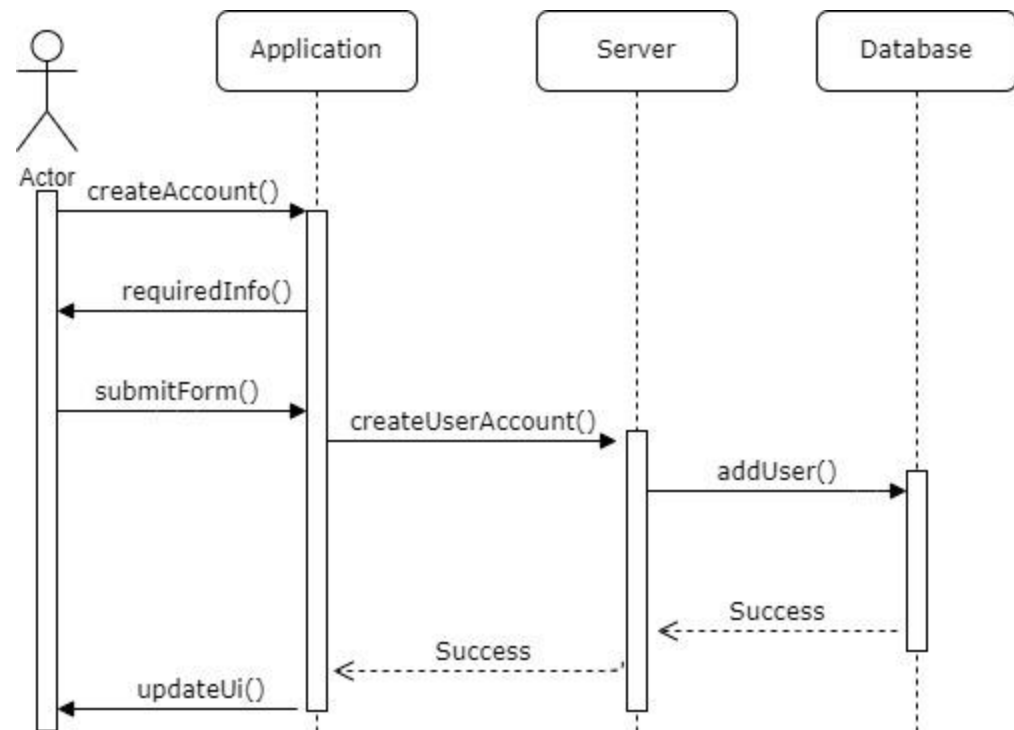


Figure 8 - Account Creation Sequence Diagram

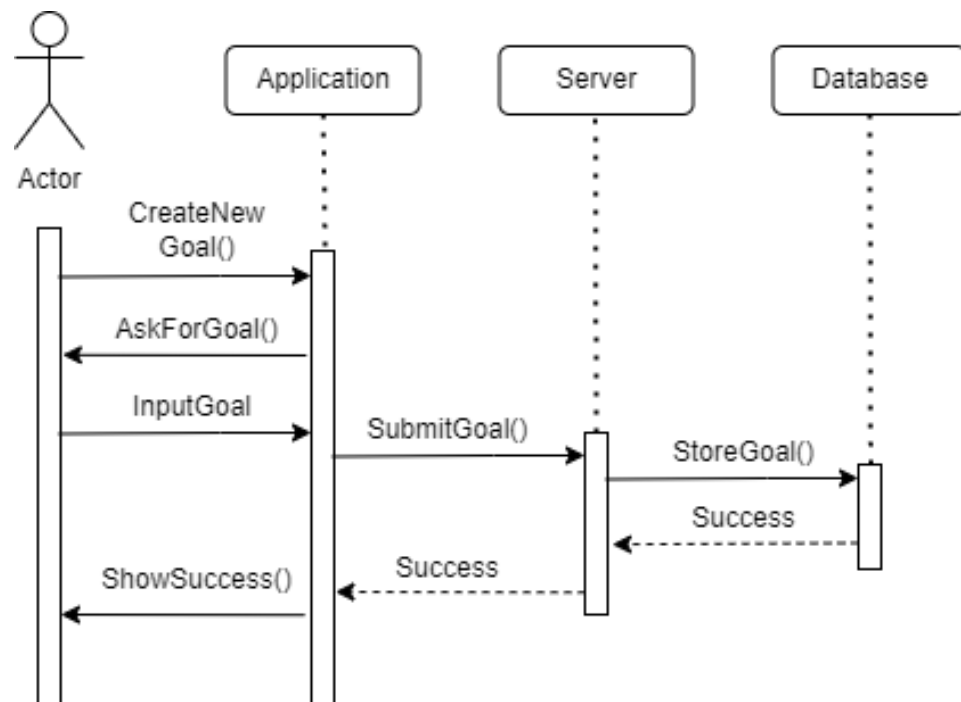


Figure 9 - Set Goal Sequence Diagram

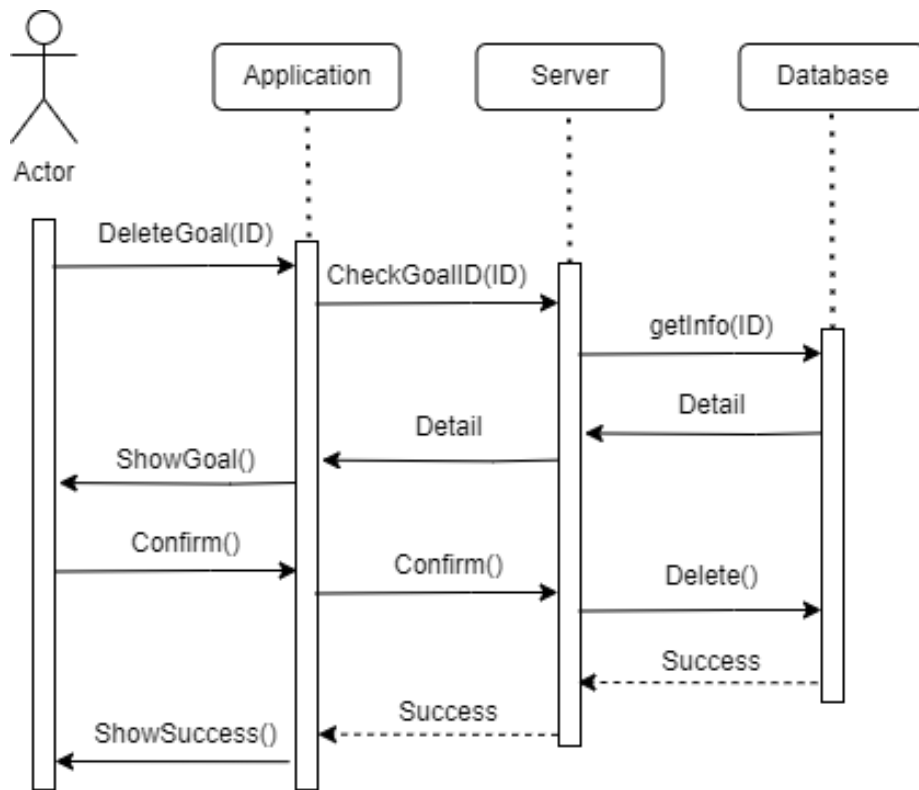


Figure 10 - Delete Goal Sequence Diagram

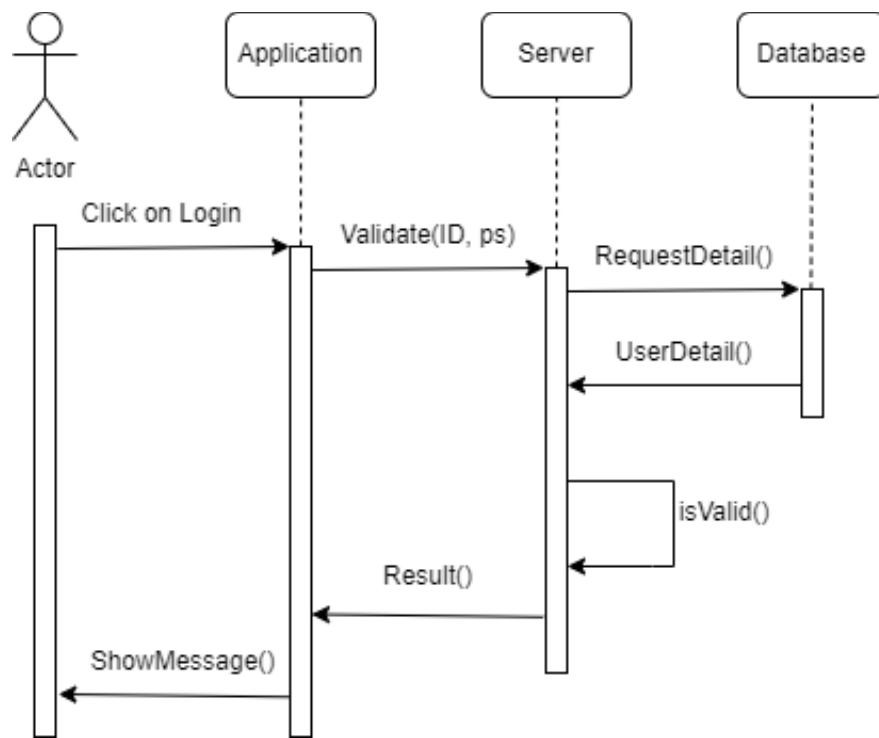


Figure 11 - Login Sequence Diagram

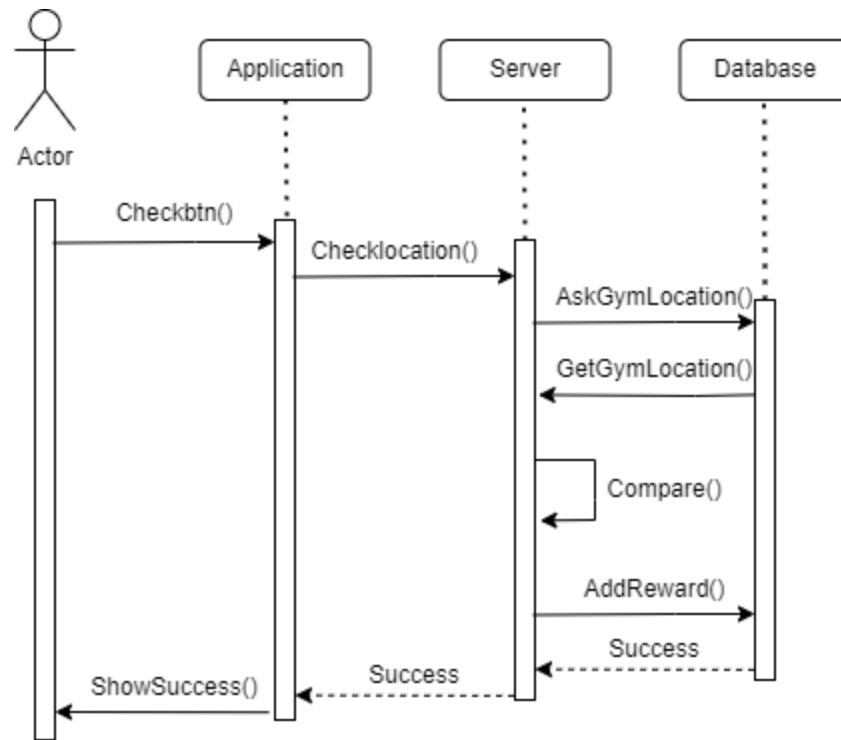


Figure 12 - Checking In Sequence Diagram

### 3c Proposed System Architecture

The system architecture we determined to be best suited for this project is the client-server architecture. It allows for a central interface for users to send requests and receive data from. Considering that it is most likely that a large amount of people will be accessing it at certain times, it is best to have multiple servers to spread the workload, especially if there are multiple clients (Gyms). To ease the work of the application itself, the servers will do the calculations as well as determining what users can and cannot see. For example, out of all the posts a member makes, only their friends can see them. The server(s) will make those connections. It would also do all the saving of data, accessing/loading of data, sharing of administrative-created goals/events, keeping track of completed goals, sharing social media posts, sharing achievements, and adding members to gym-created events. Also, the crashing of one server would not shut down the services for all the clients or their customers.

### 3d Initial Subsystem Decomposition

After realizing the main entities (member, administrator, server, database, and application), we realized that the server and database are closely tied together and can be grouped into its own subsystem. Considering that the application and member work closely together, that can become its own subsystem. As for the other parts closely tied to the member, they can be grouped into a social media and workout



subsystems that form one of the largest features in the project. The administrator can stay as it is.

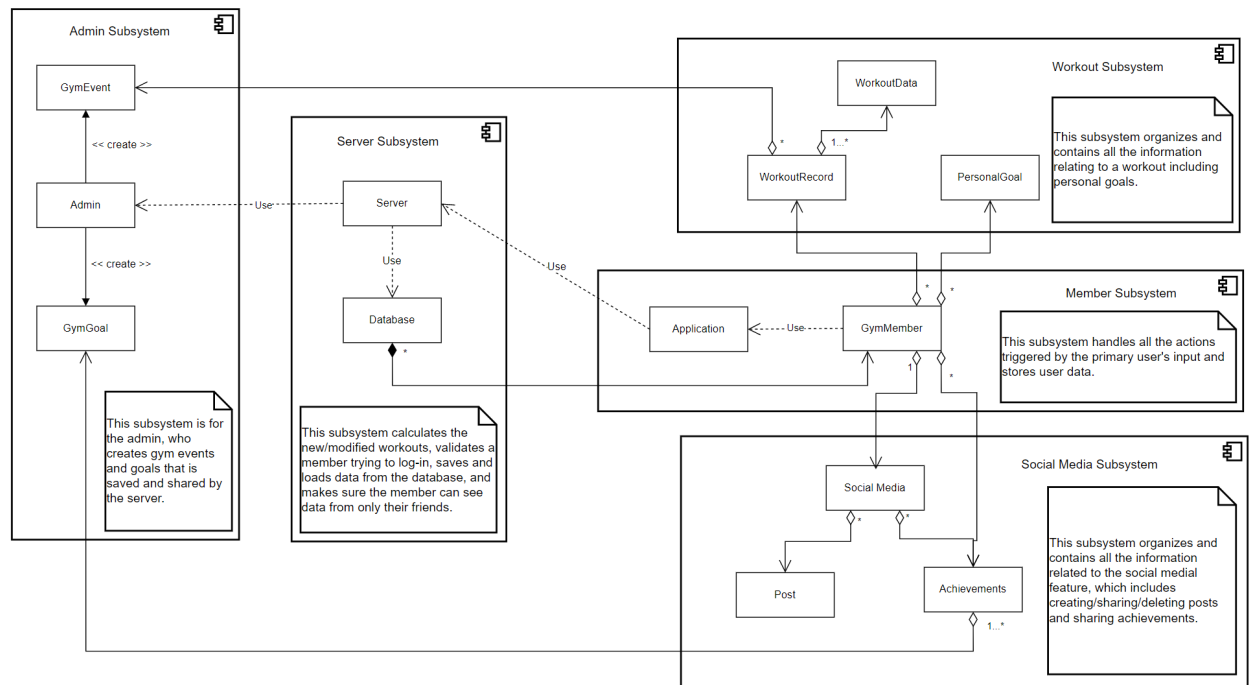


Figure 13 - Initial Class Diagram, Broken into Subsystems

## 4 Additional Design Considerations

### 4a Hardware / Software Mapping

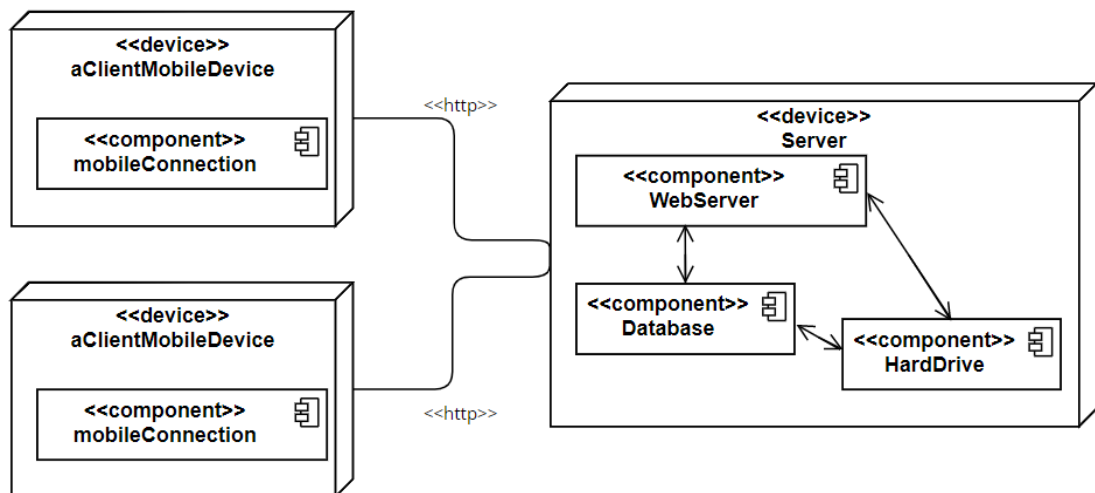


Figure 14 - Hardware / Software Map

#### **4b Persistent Data Management**

User Data is to be stored in the Database (account information like name, age, user statistics, etc.). The Workout Session Table will store all user workout sessions, each user's account is stored in the User Account Table, in which the username is the primary key, and is used as a foreign key in other tables. When the system is shut down for any reason (I.e., maintenance or outages) user data on the database is written to disk, if it is not already. Server will have a backup generator guaranteed to have at least enough power to write the contents of the entire database to disk in the event of a power outage. When the system resumes normal operations, the user data is read from the disk and the database restored.

#### **4c Access Control and Security**

The gym member should not have full access to the server. They should only be able to send requests for data belonging to them and social media posts as well as achievements that are shared by their friends, after they have been authenticated by another entity. For this reason, we will implement our server with a proxy design pattern. User applications will interact with the server through a proxy server, which allows for a layer of indirection between the actual server and the user application. This indirection means that potential malicious/erroneous actions would be caught/detected before being actually run on the real server.

#### **4d Global Software Control**

A user can only be logged-in on two devices at the same time, this allows the usage of a smartphone as well as any smaller smart device such as a smartwatch, which has some sensors that could provide additional data on the physical state of the user. The server needs to keep track of how many devices are currently connected to a single account. Should both devices be smartphones, if both need to be authenticated, one will be done before the other to prevent conflicts in the server-client system.

Since we are using the client-server model, there will be a need to use proper asynchronous programming, where the client application will await the result of a request made to the server, then perform some action once it receives the requested data. The user may make multiple requests at once, so it is important that each request is not blocked (i.e. we don't block the application thread waiting for the result of a request).

#### **4e Boundary Conditions**

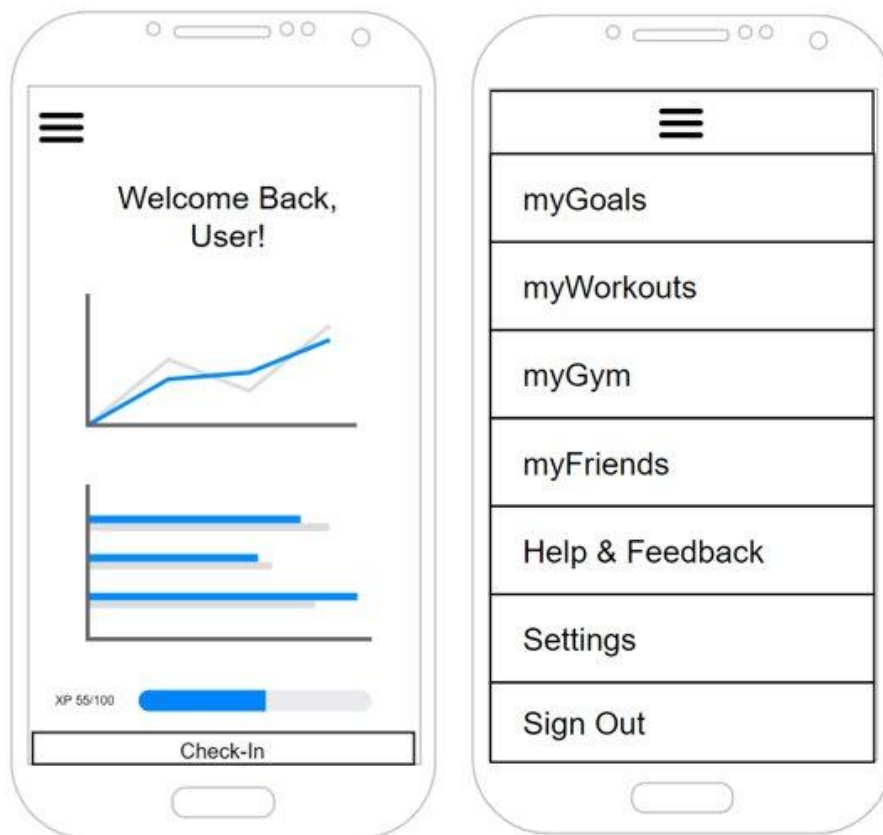
Our servers will be created by us. In the event of a shutdown due to maintenance, crashing, or some other issue, the back-up database would be loaded in from the disk that stored that data. The generator-backed-up server is guaranteed to have sufficient power to write the contents of the entire database to the disk in a power outage situation. When the system resumes normal operations, the user data is read from the

disk and the database restored. There will be a class to record the information/status of the server during shutdown events, which will create a new instance so that the system administrator can see whether the shutdown caused an erroneous state/error in the server or database, and that can be rectified. However, if any of the previously mentioned events occur, the gym member will still be able to modify and access locally-saved data. When operations resume to their normal state, the data will be updated in the database.

During a start-up, if the server is starting up normally, it will proceed as such, but if the server is starting-up after an unexpected shutdown event, checks will need to be run to see if there is any erroneous state of the server/database, and if so, an exception will be created, and the system administrator will be contacted to investigate.

Should there be any networking errors, any saving and loading of data will not be able to be processed since accessing the proxy server would result in a failure.

#### **4f User Interface**



*Figure 15 - UI Mockup*

#### **4g Application of Design Patterns**

Not applicable.

## 5 Final System Design

Almost every subsystem was able to be translated to a package. Nothing was changed in most of the packages, however, the Server package now contains the proxy server for security purposes. Otherwise, all the other relationships between the classes are the same. Now, the Proxy Server takes the place of the Server. The Proxy Server and Real Server both implement the Server interface.

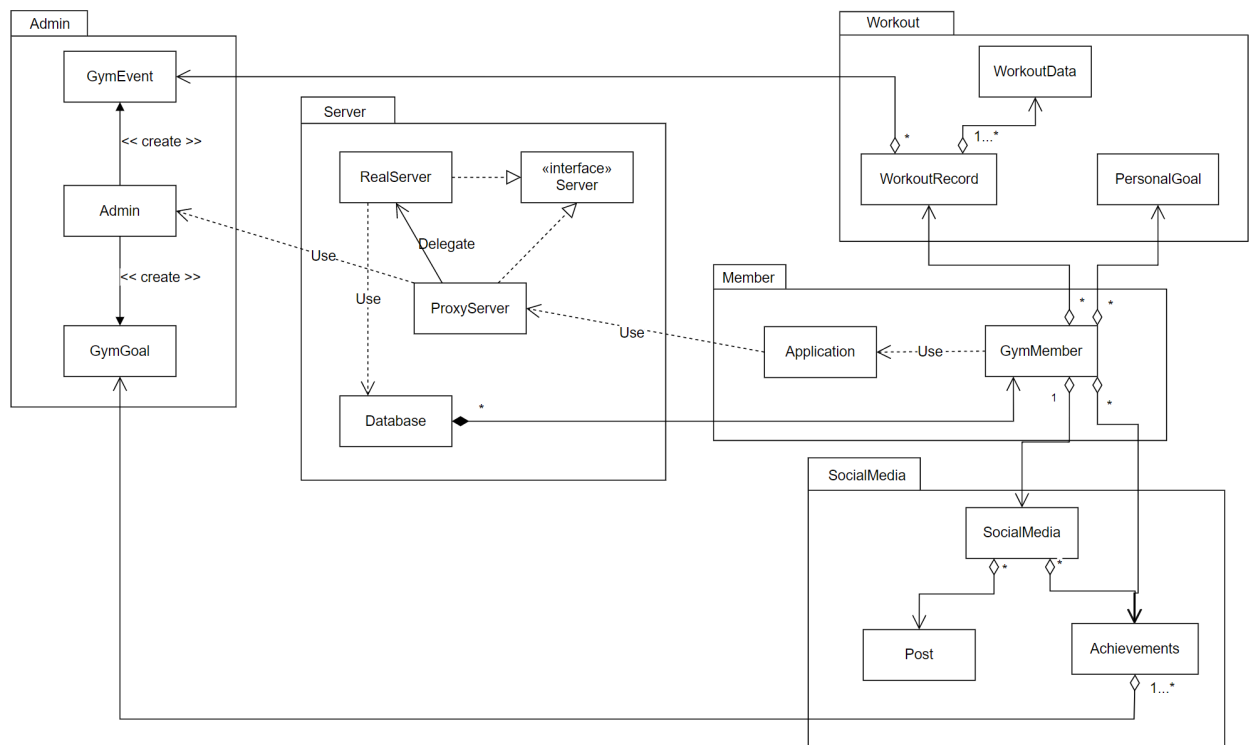


Figure 16 - Final System Design Diagram

## 6 Object Design

### 6a Admin Package

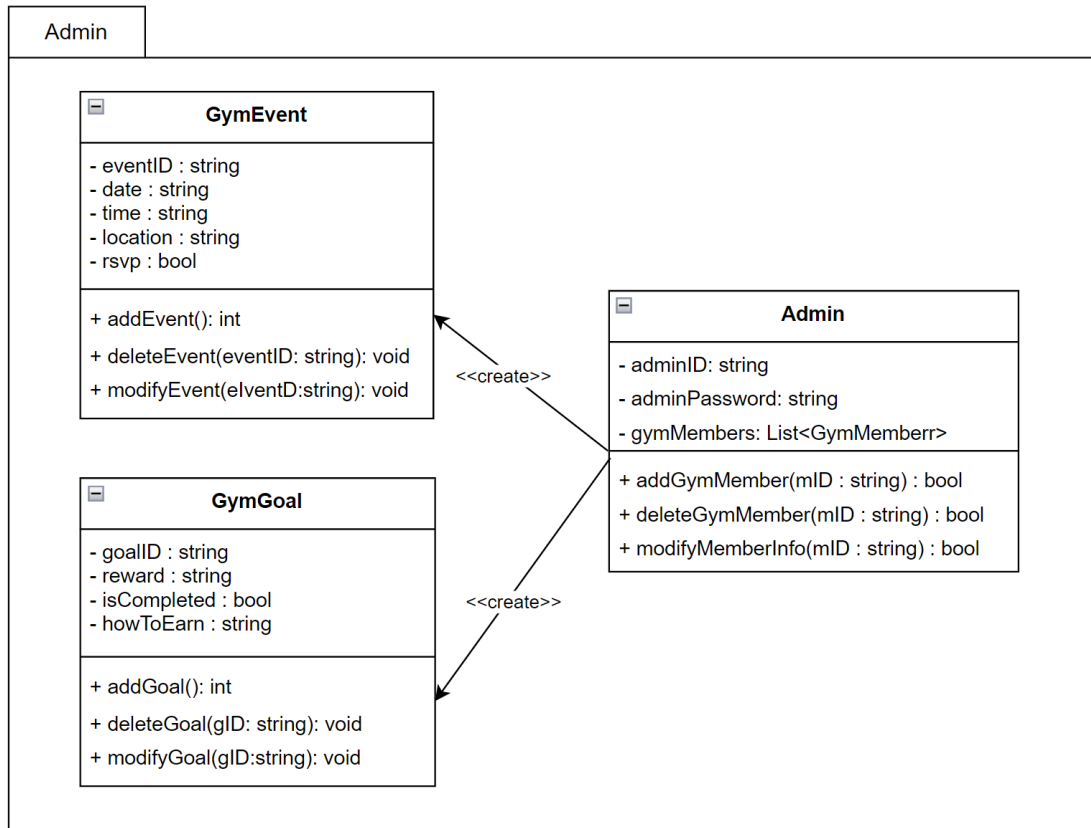


Figure 17 - Admin Package

## 6b Server Package

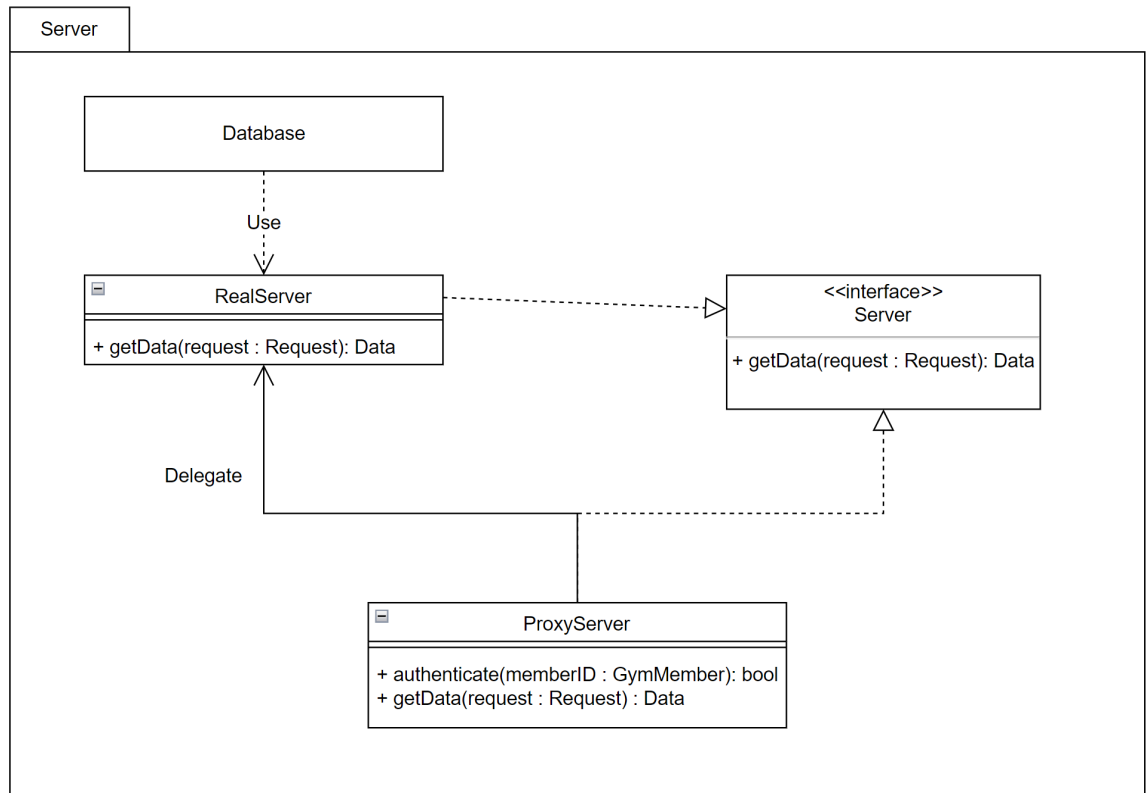


Figure 18 - Server Package

## 6c Workout Package

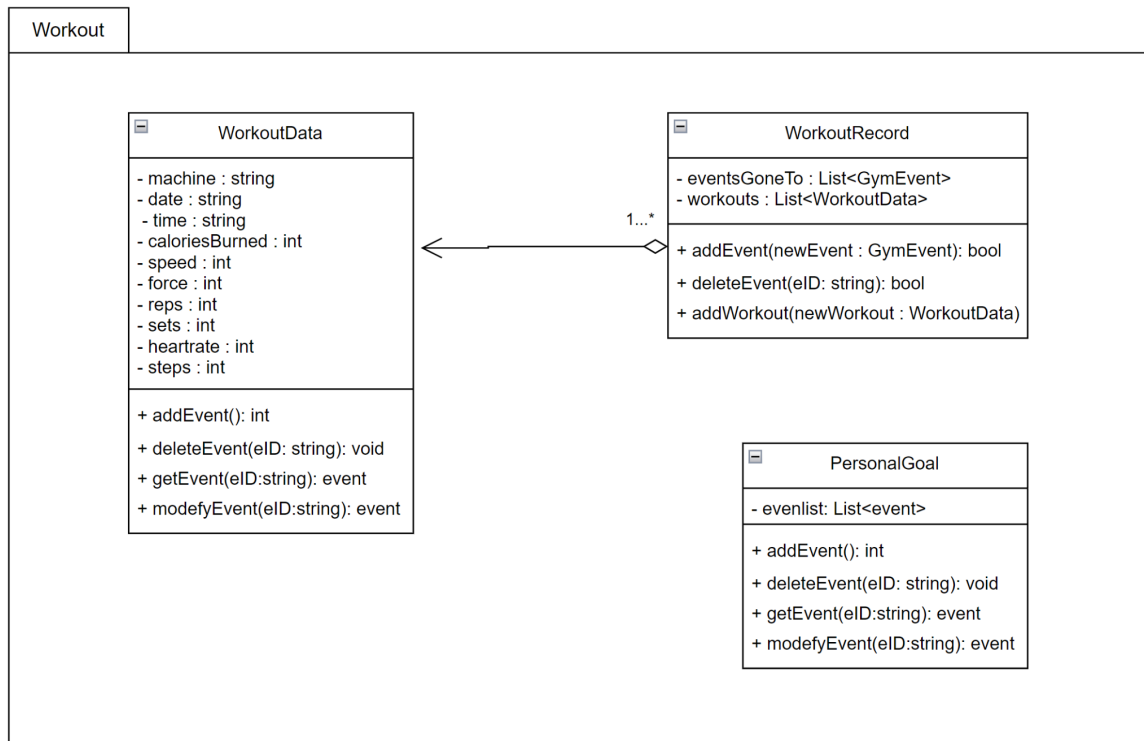


Figure 19 - Workout Package

## 6d Member Package

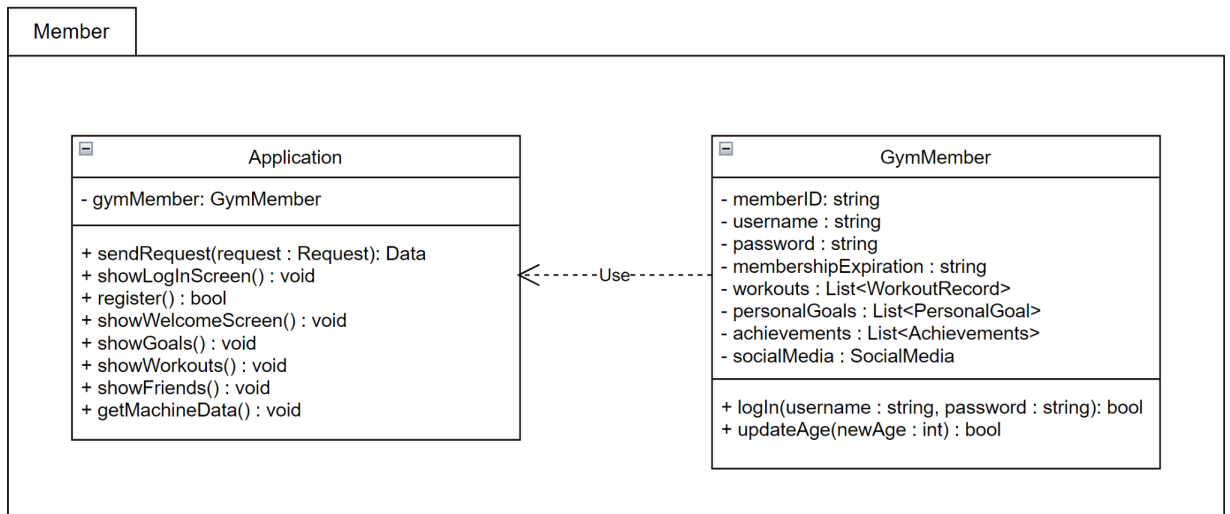


Figure 20 - Member Package

## 6e SocialMedia Package

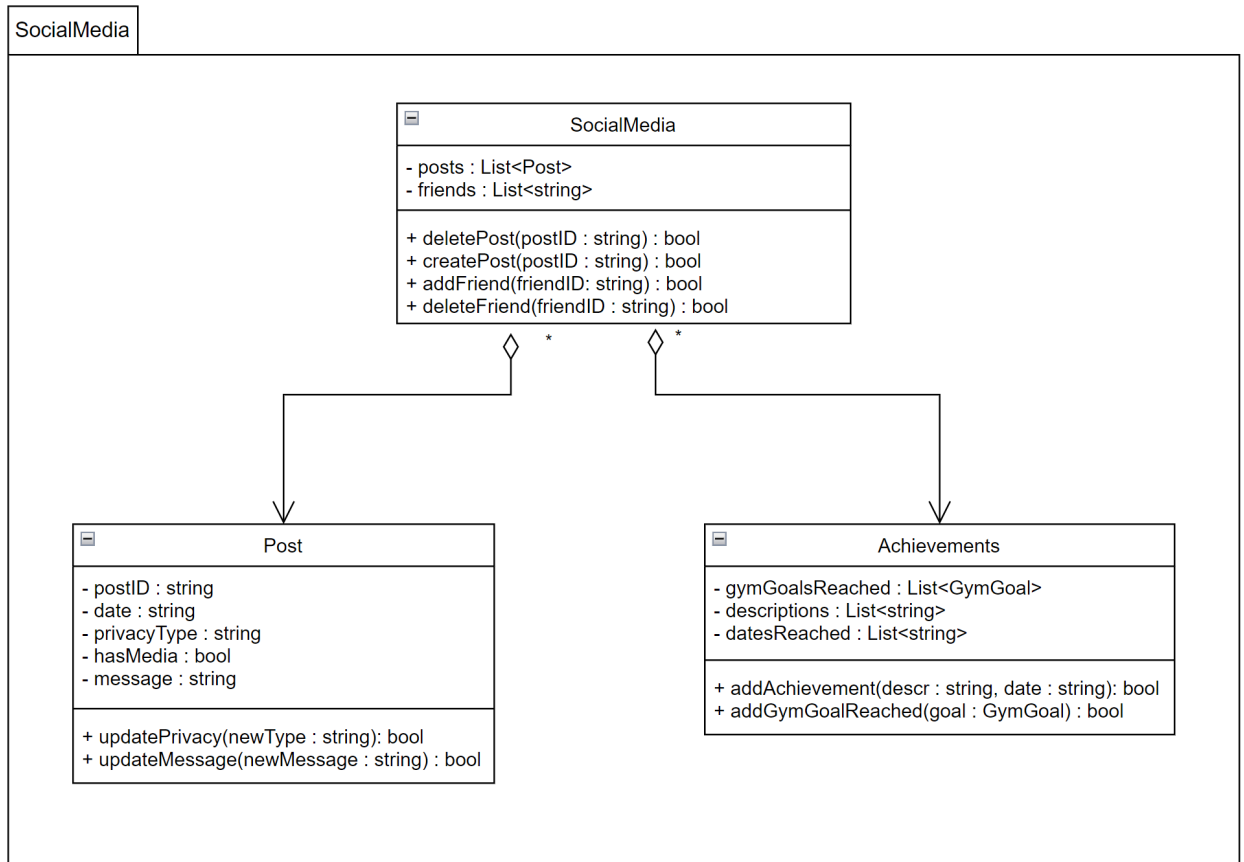


Figure 21 - SocialMedia Package

## IV Project Issues

### 1 Open Issues

One of the issues we have encountered is the large amount of data in a single record that has to be saved. While the records can be engineered as efficiently as possible, the issue of large amounts of memory consumption still remains. It can cause the servers to be slow when multiple people try to access it. Also, it can be quite costly depending on how and where user data is stored.

### 2 Off-the-Shelf Solutions

#### Ready-Made Products

One of the products we can use instead of developing is a database system. There are plenty of services that offer this such as MySQL, SQL Server, MS Access, Oracle, Sybase, and so on.



## **Reusable Components**

To make the Android mobile application, integrated development environments (IDEs) such as Android Studio, Eclipse, Visual Studio - Xamarin, and IntelliJ IDEA can be used. The listed IDEs can also connect to databases.

## **Products That Can Be Copied**

If a client Gym already has an application of sorts that can be legally copied or modified, then it is possible that certain features do not need to be added or modified.

# **3 New Problems**

## **Effects on the Current Environment**

The employees at the fitness center will have more responsibilities than before, as they will have to be trained on how to use the application, such as learning how to: add a new member to the gym's records, and connect a smart gym equipment with a user's device to receive data.

Another effect that the system could have on the working environment would be the crashing or lagging of servers due to a larger-than-originally-planned amount of people trying to access the server simultaneously.

## **Effects on the Installed Systems**

Should user data be stored in the same database as the client's other records, it could put further strain on the server, leading to slowdowns, especially if the client already has a lot of records.

## **Potential User Problems**

It is possible for the application to have issues concerning readability and usability if it is used on a smaller-than-expected screen. It is also possible that even with user testing, people will still say that the application is hard to navigate with all the features present.

## **Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

Deviations in the sensor readings of the equipment may cause deviations from the actual movement thus recording not completely accurate data.

If we rely on another service that provides the database system, when they experience an issue, our technical team cannot do anything to try to resolve the issue besides working closely with them and waiting for them to fix such issues.

## **Follow-Up Problems**

Not Applicable.

## **4 Migration to the New Product**

### **Requirements for Migration to the New Product**

If the client already has an application, we can work with the system they have in place, create a new system that works with their existing one, or create a new system entirely. It all depends on the client's wants and needs.

### **Data That Has to Be Modified or Translated for the New System**

Should the client already have a database with data, depending on the wants and needs of the client, the data will be translated/reformatted and/or preserved as necessary.

## **5 Risks**

If initial adoption or user participation is low, the social aspect of the application would not work as well as intended, and it might have trouble gaining traction if it doesn't start out with a respectable sized userbase who can advertise it to family and friends.

Because of our dependability on power and other electronic systems, should one fail, it can make the application or a feature unusable for a period of time. In the early stages of the rollout, too many users in a short period of time can cause data congestion affecting the application's data backup upload. We will need to upgrade the corresponding database/server after the project has a certain number of users.

## **6 Costs**

Depending on the number of developers working on the project, it can take anywhere from one to three years to complete. No matter the size of the development team, factoring the other services that might be used such as a database system, the financial cost can be in the range of a few hundred thousand dollars or even in the several million dollars, depending on the development time, size of the team, and the cost of the physical hardware necessary.

## **7 Waiting Room**

An idea worth considering is making a better user interface if possible. There are a lot of features in the myGymClub application, and even if the UI is carefully developed, there is always something to improve for the users.

Another feature worth mentioning is having an electronic shop (eShop) that also has some items that reward users for reaching gym-specific goals. For example, some common items various businesses offer are electronic gift cards and VIP discounts offered by different gyms.

Some fitness centers have locations all over the world, so having the mobile app also available in those areas would possibly benefit the Gym company. Incentives and goals should be set in different countries and regions according to the local user habits.

## **8 Ideas for Solutions**

Using SQL or some other relational database is advised for fast access and modification of a large quantity of user data (which is likely to see heavy load during peak gym hours). Several different tables could be used, with the username of the user as a primary key (making sure to never use the password as a primary or foreign key for security reasons!), so as to provide a unified interface for accessing/modifying that user's data.

## **9 Project Retrospective**

One of the large issues in writing the description part of the project was what exact features we would be including in the project. Some of them seemed more wants than needs, but it was a bit difficult to determine which ones would be included to make the project have sufficient features in order to make the project worth it.

One of the frustrations we had was to list down every functional requirement as detailed as possible in order for the project to not to be ambiguous. This process resulted in a large amount of requirements for each part, which may have been more than necessary.

Creating the UML diagrams, we understood roughly what the classes and subsystems were supposed to do, which made it mostly easier to do. It did take us a few tries to truly understand the relationships between classes. When we had to create the more-detailed package diagrams, it also took a while to flush out the basics of what we would need in terms of methods and attributes. We had taken some time to understand what our diagrams all meant, which resulted in a reasonable design section. From this, we realized that we need to become a bit more familiar with the meanings of different connections used in UML diagrams.

In terms of group meetings, we could have met more times each week in person so that we would not rush for things every week. In the week before the sections were due, we spent a lot of time rushing, which would have been avoidable by meeting for a bit more than four times in-person. In addition, meeting more in-person instead of on discord would have allowed us to complete tasks more efficiently, and it took us a while to realize that.

## **V Glossary**

**Achievement** – A task set by the gym member or employee(s) and is done successfully by the user.

**Event** – A planned social occasion set by someone.

**Experience** – A unit of measurement that quantifies the user's experience and progression throughout their time at the gymnasium. Also known as exp or XP.

**Gym/Gymnasium** – A club, building, or large room, that usually contains special equipment, where people go to physically exercise (also known as working out) and/or participate in athletics. Also known as a fitness center.

**Level** – A position on an imaginary numeric scale that quantifies a user's experience. It is also a positive whole number.

**Member-Only Gym(s)/Gymnasium(s)** – A club, building, or large room, that usually contains special equipment, where people go to physically exercise (also known as working out) and/or participate in athletics as long as they are a member of the group. Can also be called: (a) member-only fitness center(s).

**Reward** – A gift given to a gym member in recognition for their effort, achievement, or time.

**Smart Gym Equipment** – Special electronic machines that are used during physical exercise to enhance a person's physical state. These machines calculate information about the exercise the person is doing such as how many calories were burned, how much time they have been on the machine, and so on. Also known as a: smart gym machine.

**SQL** - Structured Query Language. Used for managing or stream processing data stored in a relational database management system.

## VI References / Bibliography

- [1] “Apps & Games Content Ratings on Google Play.” *Google Play Help*, Google.
- [2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.
- [3] A. Verma, “Top 5 Free IDE to Speed Up Android App Development.” *Your Team in India*, Your Team in India, 2021.
- [4] Draw IO, "draw.io," [Online]. Available: <https://www.draw.io/>. [Accessed 2022].
- [5] *Google Play Protect*, Google Developers.
- [6] “IEEE Code of Ethics.” *IEEE*, IEEE.
- [7] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.
- [8] L. Jordan, “Compression Speeds and Settings for 4K Video.” *LARRYJORDAN: Edit Smarter*, Thallo LLC, 2015.
- [9] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.
- [10] “Mobile App: Planet Fitness.” *Mobile App | Planet Fitness*, Planet Fitness Franchising, LLC.
- [11] Robertson and Robertson, Mastering the Requirements Process.

## VII Index

Constraints	25, 26	Project Issues	88-91
Design	75-88	References	93
Glossary	82	Requirements	28-69
Index	94	Tests	70-75
Project Description	9-28		