

Application Requirements Specification For  
<MyGym Management System>

Prepared By:

<Jie Li>

<Yiqing Huang>

<Xin Cheng>

## Purpose and Scope Statement

### Purpose

The purpose of this project is to develop a management system for a fitness center. The system will streamline the interaction between admins and members, focusing on features such as profile management and course management.

### Scope

The project has two users — admin and member. Admins can view and edit member information, and view course registration information. Members can view and update their profile, view class listings, reserve and cancel class reservations. The mapping of classes between members and admins as well as the billing system is not part of the scope of this project and therefore will not be implemented. The system will focus primarily on the interaction between admins and members.

## Requirements Narrative

In this project, users are given an option to login as either an admin or a member. Once an option is selected the user will be able to log into the system using their email and password. If the entered credentials match with the database, the user will be sent to the next screen.

Users in this system can realize the following functions according to different types:

Members can perform the following functions:

1. **View profiles:** Members can view their personal profile information.
2. **Update profiles:** Members can modify and update their profile information.
3. **View course list:** Members can access a list of all available fitness courses.
4. **Reserve courses:** Members can reserve a spot in a fitness course.
5. **Cancel course reservations:** Members can cancel a previously reserved spot in a fitness course.
6. **Login and Logout:** Members can log in and out of the system.

Admins can perform the following functions:

1. **View member list:** Admins can access a comprehensive list of all members registered in the gym.
2. **Add members:** Admins can record new memberships.
3. **Update members:** Admins can update members information.
4. **Delete members:** Admins can delete non-renewing members.
5. **View registration list:** Admins can view course registration list.
6. **Login and Logout:** Admins can log in and out of the system.

# Objectives

Login view 1-1:

Gym Management

Welcome to MyGYM Management System !

Please Select your login as :

Login view 1-2:

Gym Management

Username:

Password:

Member Users' view as below:  
2-1 Members' Home page

Gym Management

Admin Users' view as below:  
3-1 Admin's Home page

Gym Management

2-2 View profile

Gym Management

[Home](#) | [My Account](#)

Username: yolandama  
Name: Yolanda Ma  
Email: yolandama@gmail.com

3-2 View members

Gym Management

[Home](#) | [Membership Management](#)

Member Name	Username	Email
Yolanda Ma	yolandama	yolandama@gmail.com
Matt Con	matt123	matt123@gmail.com
-	-	-

2-3 Update profile

Gym Management

[Home](#) | [My Account](#)

Username:

New Name:

New Email:

3-3 Add members

Gym Management

[Home](#) | [Membership Management](#)

Username:

Member Name:

Email:

(To be continued)

(Continued)

2-4 View courses

Gym Management

[Home](#) | [My Course](#)

**View courses**  
**Reserve courses**  
**My reservation**

Course List

Name	ID	Time	Coach	Equipment
Yoga I	0008	Mon 9:00AM	Lana	Yoga mat
Yoga II	0009	Thu 9:00AM	Julia	Yoga mat
Zumba	0010	Fri 7:00PM	Marco	N/A
--	--	--	--	--

3-4 Update members

Gym Management

[Home](#) | [Membership Management](#)

**View members**  
**Add members**  
**Update members**  
**Delete members**

Username:

Member Info

New Name:

New Email:

2-5 Reserve courses

Gym Management

[Home](#) | [My Course](#)

**View courses**  
**Reserve courses**  
**My reservation**

Course code:

Course Info

Course name: Yoga II

Time: Thursday, 9:00 AM

Coach: Julia

Equipment: Yoga mat

(Display after user input course code)

3-5 Delete members

Gym Management

[Home](#) | [Membership Management](#)

**View members**  
**Add members**  
**Update members**  
**Delete members**

Username:

Member Info

Member Name: Yolanda Ma

Email: yolandama@gmail.com

2-6 My reservation

Gym Management

[Home](#) | [My Course](#)

**View courses**  
**Reserve courses**  
**My reservation**

Checkbox	Name	ID	Time	Coach	Equipment
<input type="radio"/>	Yoga I	0008	Mon 9:00AM	Lana	Yoga mat
<input type="radio"/>	Zumba	0010	Fri 7:00PM	Marco	N/A
--	--	--	--	--	--

3-6 View register list

Gym Management

[Home](#) | [Course Management](#)

Register List

Course co	Course na	Usenam	Member na	Coac
0009	Yoga II	yolanda	Yolanda	Julia
--	--	--	--	--

2-7 Log out

Gym Management

Are you leaving?

3-7 Log out

Gym Management

Are you leaving?

## Functional Specification

MyGym Management System is basically divided between 2 users - Members and Admins.

The function will be invoked based on who logs in to the system.

### **Member Functions:**

1. viewProfile():

Description: Allows members to view their personal profile information.

Inputs: None

Outputs: Member details (Name, Username, Email).

2. updteProfile():

Description: Allows members to update their personal profile information.

Inputs: Personal information (Name, Email)

Outputs: Confirmation of successful update.

3. viewCourse():

Description: Permits members to view the list of available fitness courses.

Inputs: None.

Outputs: Course details (Course Name, Schedule, Capacity, etc.).

4. reserveCourse():

Description: Allows members to reserve a spot in a fitness course.

Inputs: Course Code.

Outputs: Confirmation of successful course reservation.

5. cancelReservation():

Description: Permits members to cancel a previously reserved fitness course.

Inputs: None.

Outputs: Confirmation of successful course reservation cancellation.

6. memberLogin():

Description: Allows members to log into the system.

Inputs: Username, Password.

Outputs: Successful login message.

7. memberLogout():

Description: Allows members to log out of the system.

Inputs: None

Outputs: Successful logout message.

### **Admin Functions:**

1. viewMember:

Description: Allows admins to access a comprehensive list of all members registered in the gym.

Inputs: None

Outputs: List of members (Name, Username, Email).

2. addMember():

Description: Permits admins to add membership information.

Inputs: Member details (eg. Name, Email, etc.)

Outputs: Confirmation of successful member addition.

3. updateMember():

Description: Permits admins to update a member's information.

Inputs: Member details (eg. Name, Email, etc.)

Outputs: Confirmation of successful member deletion.

4. deleteMember():

Description: Permits admins to delete a member from the gym management system.

Inputs: Username.

Outputs: Confirmation of successful member deletion.

5. viewRegistrationList():

Description: Permits admins to view the course registration list.

Inputs: None.

Outputs: List of registration information.

6. adminLogin():

Description: Allows admins to log into the system.

Inputs: Username, Password.

Outputs: Successful login message.

7. adminLogout():

Description: Allows admins to log out of the system.

Inputs: None.

Outputs: Successful logout message.

### **Class Needed in the project:**

- **Member Class:**
  - **Attributes:**
    - Username: String (Unique identifier for each member)
    - name: String
    - email: String
    - password (for login authentication)
  - **Methods:**
    - viewProfile(): Retrieves and returns details of the member (Name, Username, email, etc.).
    - updateProfile(): Updates the member's profile with the provided information.
    - viewCourse(): Retrieves and returns a list of available fitness courses.
    - reserveCourse(): Reserves a spot in a fitness course.
    - cancelReservation(): Cancels a previously reserved fitness course.
    - login(): Authenticate member login.
    - logout(): Logout the member from the system.
- **Admin Class:**
  - **Attributes:**
    - Username: String (Unique identifier for each admin)
  - **Methods:**
    - viewMember(): Display a list of all members (Name, Username, Email).
    - addMember(): Add a new member to the system and provide a confirmation.
    - updateMember(): Update a member's information.
    - deleteMember(): Delete a member from the system and provide a confirmation.
    - viewRegistrationList(): View course registration list.
    - login(): Authenticate admin login.
    - logout(): Logout the admin from the system.
- **Course Class:**
  - **Attributes:**
    - courseCode: String (Unique identifier for each course)
    - courseName: String
    - schedule: String
    - capacity: int
  - **Methods:**
    - getCourseDetails(): This method is designed to retrieve the details of a specific course identified by its CourseCode.
    - getAvailableCourses(): This method is intended to provide a list of all available courses in the system.
- **Reservation Class:**

- Attributes:
  - reservationCode: String (Unique identifier for each reservation)
  - Username
  - CourseCode
- Methods:
  - reserveCourse(): Reserve a spot for a student in a fitness course.
  - cancelReservation(): Cancel a previously reserved fitness course.

**Technologies Needed in the project:**

Java framework;

UI for application - Java Swing;

Database - file management: storage, read and write;

Json serialization and deserialization.



## Logic Specification

Users log in to the system from the UI interface. Based on the user's role, the system leads the user to the corresponding interface.

