

**Gym EasyBook**

**Students:**

|  |  |
| --- | --- |
| **Sama Haji** | **12218586** |
| **Dana Bustami** | **12217788** |
| **Taqwa Qadah** | **12325195** |

**Yana shrim 12217339**

**Supervisor:**

**Dr. Emmad I. M. Saadeh**

A project report submitted in partial fulfillment of the requirements for Software Engineering course in Computer science department.

An-Najah National University

2024/2025

**Table of Contents**

Table of Contents ii

List of Tables iv

Abstract v

**Chapter One**

**Introduction 1**

1.1 Introduction 1

1.2 Project Scope 1

1.3 Problem Specification 1

1.4 Goals and Objectives 1

1.5Motivation 1

1.6 System Requirement 1

1.7 Project Plan and Schedule 1

1.8Outline of the Project 1

**Chapter Two**

**Literature and Methodology 2**

2.1Introduction 2

2.2 Current Systems*(Advantages, disadvantages, and weakness)* 2

2.3 Proposed System*(Advantages, disadvantages, and weakness)* 2

2.4 Feasibility Study 2

2.5Methodology*(for example: Agile, Waterfall, etc.)* 2

**Chapter Three**

**System Analysis And Design 3**

3.1System Analysis 3

3.1.1 Requirement Collection 3

*Examples of methods:*

3.1.1.1 Interview outline 3

3.1.1.2 System Service Request 3

3.1.1.3 Prototype 3

3.1.1.4 *Other Methods* 3

3.1.2 Requirement Studied 3

3.1.3 Requirement Structured 3

3.1.4 DFDs 3

3.2 System Design 3

*For Examples:*

3.2.1 Static Context Diagram 3

3.2.2 Use Case Diagram 3

3.2.3 Sequence Diagrams 3

3.2.4 Activity Diagrams 3

3.2.5 Scenarios 3

3.2.6 State Diagrams 3

3.2.7 Class Diagram 3

3.2.8 Database Schema (Mapping “Tables”) 3

**Chapter Four:**

**System Implementation 4**

4.1 Programming Languages 4

4.1.1 Why Using *(for Example: Java*) 4

4.1.2 Why Using *(for Example: SQL Server)* 4

4.2Implementation 4

4.2.1 Introduction 4

4.2.2 Sample of Forms 4

4.2.3 Sample of Reports 4

4.2.4Pseudo Codes 4

**Chapter Five:**

**System Testing 5**

**5.1 Testing Plans 5**

**5.2 Types and Steps of Test**ing 5

*For Examples:* (Unit testing, Integration testing, System testing, …)

**Chapter Six:**

**The Conclusions 6**

6.1Conclusion 6

6.2 Future Work 6

**Appendix 7**

A:System Installation 7

**References 8**

**List of Tables**

Table 1: Project Plan and Schedule 1.7

**Abstract**

Gym EasyBook is a web-based scheduling and booking system specifically tailored for gyms and fitness clubs to make the process of managing class bookings easier. It provides a simple-to-use interface for gym members to view a list of available fitness classes such as Cardio, Zumba, and Weightlifting, along with real-time class capacity and number of current participants. The members can simply get registered, login, browse through their favorite classes based on time and category and go ahead with booking or cancellation at any point in time.

This system not only reduces administrative burden on gym staff but also eliminates issues like class overbooking, scheduling issues, and human errors. By digitizing and automating the entire process, Gym EasyBook enables gyms to optimize resource use, improve communication with customers, and provide a better and better-organized experience to all members. The site also includes notifications and alerts so users are notified about their exercise sessions and changes to timetables, making for a smarter, more efficient gym.

**Chapter 1**

**1.1 Introduction**

The Gym EasyBook is a smart web-based system developed to automate the booking process of the gym classes. Through this system, the gym members can book their preferred classes online without visiting the front desk. The system offers efficient organization, prevents overbooking, and gives real time confirmation on class availability.

**1.2 Project Scope**

This system will allow the gym members to:

* View available classes (Cardio, Zumba, Weightlifting…).
* View the number of persons in each class.
* Book or cancel class bookings online.
* Receive notification of their booking status.
* The administrators will manage class schedules, monitor booking limits, and set class capacities accordingly.

**1.3 Problem Specification**

The majority of gyms nowadays either manage class bookings manually or with basic spreadsheets, which leads to:

* Overcrowding in classes.
* Ineffective tracking of member attendance.
* Inability to manage easily cancellations or last-minute changes.
* The Gym Schedule Manager solves these problems by offering a dynamic, real-time.
* system accessible by members and gym staff.

**1.4 Goals and Objectives**

* Automate class booking.
* Avoid overcrowding by capping bookings at class capacity.
* Enable members to self-manage their bookings at any time.
* Provide administrators with the ability to easily control and modify schedules.
* Improve customer satisfaction and gym efficiency.

**1.5 Motivation**

As gyms expand and the range of classes increases, the need for a good booking system becomes ever more pressing. Manual systems can no longer handle the stresses of busy schedules and demanding customers. A computerized solution offers simplicity for members and enables the gym staff to focus on delivering a quality service rather than managing paperwork.

**1.6 System Requirements**

Functional Requirements:

* Members can register and log in.
* Members can see class schedules.
* Members can book or cancel a class.
* The system shows how many current attendants are in each class.
* Admins can add, edit, or delete classes.
* Admins can monitor class capacities and attendent lists.

Non-Functional Requirements:

* The system should be responsive and user-friendly.
* The system should have email notifications for booking confirmations or cancellations.
* The system must be secure to protect member information.

**1.7 Project Plan and Schedule**

|  |  |  |
| --- | --- | --- |
| **Task** | **Start date** | **End date** |
| Requirement Specification | 25/04/2025 | 30/04/2025 |
| System Design | 01/05/2025 | 04/05/2025 |
| Implementation | 05/05/2025 | 10/05/2025 |
| Testing | 11/05/2025 | .... |
| Deployment | .... | .... |

Tabel 1

**1.8 Outline of the project**

* **Chapter One** provides an introduction to the project, including the project scope, system requirements, goals, and project plan.
* **Chapter Two** discusses the literature review and methodology, highlights the differences between existing systems and the proposed system, and presents the feasibility study.
* **Chapter Three** focuses on system analysis and design, including requirement gathering, requirement analysis, and requirement structure.
* **Chapter Four** is dedicated to the system’s UML design.
* **Chapter Five** presents the system's conclusion and outlines future work.

Chapter Two

Literature and Methodology

**2.1 Introduction**

Most gym facilities still rely on traditional (manual) methods to manage their class schedules and bookings. Typically, members must visit the Reception to ask about available classes, sign up manually, or call to reserve a spot. This process might have the advantage of being internet-independent, but the disadvantages far outweigh this single benefit.

**Disadvantages of the current system:**

1. **Overcrowding in Classes** Without an automated limit on class sizes, it's easy for classes to become overcrowded, which affects quality and safety.
2. **Inefficient Booking and Attendance Tracking** Manual sign-ups make it hard for staff to track attendance accurately, and members may forget whether they’re registered or not.
3. **Limited Flexibility** Any last-minute changes or cancellations are difficult to manage manually, often leading to frustration for both staff and gym-goers.
4. **Lack of Real-Time Information** Members cannot check availability remotely or receive instant confirmation, which reduces convenience and user satisfaction.

**2.2 The proposed software system**

Gym EasyBook is a new web-based system designed to automate and improve the class booking experience in modern gyms. It enables gym members to search for available classes, view the number of current participants, and easily book or cancel their reservations online. all without needing to interact physically with staff.

**2.3 Proposed System (Advantages and Disadvantages)**

Advantages:

1.**automation of the process**:Gym EasyBook streamlines the class reservation process, reducing reliance on staff and manual systems.

2.**real-time remote access**:users can book a class , buy a membership or track it anytime and from anywhere through a web interface and receive instant updates and answers.

3.**more accuracy and control:** Reduces errors such as overbooking or lost registrations common in manual systems,and staff easy control of updating,adding or deleting any class.

4.**different registration system :**different interface for different members which allow coaches to manage classes , users to pick a service , even admin control panel for any updates to the software.

Disadvantages:

1.**Initial Setup Cost:** Setting up the system (development, hosting, and training) may involve upfront costs for the gym.

2.**Security Risks:** As with any web-based application, the system must be well-protected against data breaches and unauthorized access.

3.**adaptability and dependency**:some users may struggle with using the system and it must be well-protected against data breaches and unauthorized access.

**2.4 Feasibility Study**

**1.Technical Feasibility:**The system is technically feasible as it is developed using widely supported web technologies ( HTML, CSS, JavaScript, PHP, and SQL). Most gyms already have internet infrastructure ,making integration smooth. The project team also has the necessary technical skills.

**2. Economic Feasibility:** While the initial development and deployment may require investment, the long-term cost savings make it a good investment. Improved customer satisfaction and efficiency could lead to increased memberships and revenue.

**3. Operational Feasibility:** Gym EasyBook aligns with the operational needs of modern fitness centers. It simplifies daily tasks for staff and enhances the experience for users. Staff training is minimal, and the intuitive interface ensures members can use it with little guidance.

**4. Legal and Ethical Feasibility:**The system must satisfy data privacy regulations (such as GDPR if applicable), as it handles user information. Implementing proper authentication and encryption mechanisms ensures data protection and ethical use.

**5. Schedule Feasibility:** According to the project plan, the system follows a realistic schedule from requirements gathering to deployment. The timeline provides reasonable time for development, testing, and refinement before full implementation.

**2.5 Methodology**

**Agile methodology** was selected after researching for the creation of the Gym Schedule Manager system.

Reasons why Agile was selected:

* Quick delivery through short iterative cycles (sprints).
* Continuous feedback from gym members and administrators.
* Flexibility to adapt to changing requirements (e.g., adding new types of classes).
* Better risk management and early detection of issues.

Development Phases:

|  |  |  |
| --- | --- | --- |
| Release | Iteration | Tasks |
| Release A | Iteration 1 | Member Registration + Login system |
| Release A | Iteration 2 | View and Book Classes Module |
| Release B | Iteration 3 | Cancellation Management + Email Notifications |
| Release B | Iteration 4 | Admin Dashboard for Class Management |

Chapter Three

Requirements elicitation and analysis

**3.1 Requirements Discovery**

To correctly identify the requirements for the Gym EasyBook system, our team used multiple methods to gather detailed and relevant information from both users and gym staff and management The following approaches were used:

**Research:**

 We studied an existing gym in Nablus to identify the strengths and limitations of currently available systems. Gyms such as “hard rock gym)(dynamic gym)were analyzed in terms of usability, class scheduling features, and user interaction. This allowed us to extract core functionalities that are the most important requirements for an efficient and modern gym booking system.

**Interviews:**

We had interviews with gym administrators and regular members from several local fitness clubs. Gym staff told us their thoughts about the challenges with manual booking systems, including tracking attendance, class capacity management, and dealing with many members at the same time, Members emphasized the need for faster, flexibility, and real-time information regarding their bookings and registration.

**Observation of Work Environment:**

Team members visited gyms to observe how current booking systems function in real-time. We noted how class papers were filled and how cancellations or changes were dealt with. These observations highlighted key points and informed many of the proposed features of Gym EasyBook.

**Brainstorming Sessions:**

We had many meetings to summarise and discuss the key factors in the system. These include the Admin, who manages class schedules and capacity, and the Member, who browses, books, or cancels classes.

We also focused on prioritizing features such as automated notifications, real-time updates, and user-friendly interfaces.

Identified System Actors:

**Admin:**

* Adds, edits, and deletes class schedules.
* Monitors participant numbers.
* Sends notifications about schedule changes.

**Gym Member:**

* Registers and logs into the system.
* Views available classes and their current capacity.
* Books or cancels a class online.
* Receives confirmation and reminder notifications.

Us as developers :

* Constant updates.
* Fast respond to errors or glitches .
* Automating some processes.

**Functional Requirements**

These describe what the system should do—features and behavior expected by the users and administrators.

For Members:

* Register an account and log in securely.
* View a list of available fitness classes (e.g., Zumba, Cardio, Weightlifting).
* View the number of current participants per class.
* Book a class.
* Cancel a previously booked class.
* Receive email or system notifications confirming bookings or cancellations.
* View personal booking history.

For Administrators (Gym Staff):

* Add, edit, and delete class schedules.
* Set and update class capacity limits.
* Monitor number of bookings per class.
* View attendance and booking statistics.
* Send notifications or alerts (e.g., class cancellations or updates).
* Manage member accounts and handle registration issues.

**Non-Functional Requirements**

These specify how the system should behave—performance, usability, reliability, etc.

Usability:

The system should have a clean, intuitive, and user-friendly interface accessible even to non-tech-savvy users.

Availability:

The system should be available online 24/7, allowing users to book or cancel at any time.

Performance:

The system should process bookings and cancellations in real-time without noticeable delay.

Scalability:

It should support multiple users and allow future developments and updates (more classes, branches, or users).

Security:

* Member data (names, emails, passwords) should be securely stored and encrypted.
* Role-based access should be implemented (admin or member).

Responsiveness:

The interface should be mobile-friendly in the first place and function well on various devices and screen sizes to add full functionality.

Reliability:

The system should function correctly under normal usage and handle failures gracefully (e.g., failed bookings or network interruptions).

Maintainability:

The system codebase should be modular and well-documented for future updates or maintenance.

**3.2 Requirements classification and organization**

Requirements Organization

The requirements are organized according to the Incremental Development approach, where the system is built in multiple phases (increments), each offering some features.

**Increment 1:**

* Basic User Features.
* User registration.
* User login/logout.
* View available gym classes.

**Increment 2:**

* Booking System.
* Book a class.
* Cancel a booking.
* Email confirmation for booking/cancellation.

**Increment 3:**

* Admin Features.
* Admin login.
* Admin can add, edit, or delete classes.
* View participant number for each class.

**Increment 4:**

* System Enhancements.
* Improved user interface and experience.
* Security improvements.
* Notifications through email when a class is completely booked.

**Modules:**

1. **User Management Module**
   * Handles user registration, login, and profile modification.
2. **Class Booking Module**
   * Allows members to view, book, and cancel class bookings.
3. **Class Management Module**
   * Allows administrators to add, modify, and delete class sessions.
4. **Notification Module**
   * Automatically sends confirmations and reminders through email.
5. **Reports and Dashboard Module**
   * Produces statistics and analytics for administrators.
6. **Access Control Module**
   * Handles admin roles and system settings.

**Subsystems:**

1. **Authentication Subsystem**
   * Ensures secure login and user access for all modules.
2. **Booking Engine Subsystem**
   * Central logic responsible for class capacity, availability, and seat allocation.
3. **Communication Subsystem**
   * Outgoing email and system-alert messages.
4. **Database Management Subsystem**
   * User information; class timetables; reservation records; logs.
5. **Admin Control Subsystem**
   * An exclusive interface for configuring and managing the system.

**3.3 Requirements Prioritization and Negotiation**

The requirements were prioritized according to importance for core system functionality, user value, and implementation effort. The following categorization was used:

**High Priority (Must-Have):**

* User login and registration
* Class booking and cancellation
* Admin management of classes (add/edit/delete)
* Display classes with real-time data

**Medium Priority (Should-Have):**

* Reminders through email for class booking/cancellation
* Display number of participants for each class
* Admin dashboard and reports

**Low Priority (Could-Have):**

* Interface improvements
* User class preferences and filtering
* Optional class rating scheme
* Conflict Resolution and Negotiation

While gathering requirements, we got a number of contradicting demands such as:

* Some users were proposing unlimited freedom in booking classes, but others insisted on strict management to avoid overbooking of classes.
* Solution: We introduced the feature of limiting bookings by class capacity but hold back on reserve waitlisting for future updates.
* Admins needed manual control of classes, while members wanted everything automated.
* Solution: We retained the admin controls manual, but introduced automation for user-level actions like bookings and reminders.
* Some of the requests were beyond project scope (e.g., mobile app version).
* Resolution: Deferred to future development iterations due to time and resource limitations.

All prioritizations were reviewed in stakeholder meetings to validate that user satisfaction and business goals were aligned.

# **3.4 requirement specification**

The requirements specifications are well **Structured English**

**1– Show the available Gym Classes**

|  |  |
| --- | --- |
| Use case name | Show for available gym glasses |
| Actor | Gym member |
| Trigger | Member clicks 'available classes' on a the main page |
| Preconditions | The system must have existing classes stored in the database |
| Main steps | 1. The member logs in 2.The member clicks to ' **available Classes**'  3.The system retrieves all upcoming classes from the database  4.The system displays class name, date, time,the coach name and number of available seat |
| Postconditions | The member sees the updated list of gym classes |
| Exceptions | -If no classes are scheduled : System displays : No classes available  - If there's a system error :System displays :Unable to load the page |

**2- Book a Gym Class**

|  |  |
| --- | --- |
| Use case name | Book a gym class |
| Actor | Gym member |
| Trigger | Member clicks 'book' on a class thay want to join |
| Preconditions | The system must have existing classes stored in the database |
| Main steps | 1. Member logs in 2.The member navigates to ' **available Classes**'  3.The system displays the available classes  4.Member selects a class and clicks "Book" 5.System checks the availability to book  6.If a seat is available, system registers the booking 7.System sends confirmation message and updates availability |
| Postconditions | The member is registered for the class and the number of available seats is reduced by one |
| Exceptions | - Class already started : Cannot book - Member not logged in : Redirect to login page  -Class is full :System displays "This class is fully booked"  -Member already booked :System displays "You are already enrolled in this class" |

**3- Cancel Gym Class Booking**

|  |  |
| --- | --- |
| Use case name | Cancel gym class booking |
| Actor | Gym member |
| Trigger | Member clicks 'Cancel' on a previously booked class |
| Preconditions | The member must have an existing class booking |
| Main steps | 1. Member logs in  2. Member navigates to 'My Bookings'  3. Member selects a class to cancel  4. Member clicks 'Cancel' and confirms  5. System updates class availability and removes the booking  6. System display confirmation message |
| Postconditions | The booking is removed and class seat becomes available |
| Exceptions | - Class already started : Cannot cancel - Member not logged in : Redirect to login page  - Member has no existing booking :System displays “You have no booking to cancel” |

**4- Add NEW Gym Class**

|  |  |
| --- | --- |
| Use case name | Add new gym class |
| Actor | Admin |
| Trigger | Admin clicks 'Add class' from admin interface |
| Preconditions | Admin must be logged into the system |
| Main steps | 1. Admin logs into system 2. Admin clicks on 'Add class'  3.System displays a form to enter class details  4.Admin enters class name, type, date, time, and capacity  5.Admin submits the form 6.System saves the class and updates the class schedule |
| Postconditions | The new class is added to the system and becomes available for members to book |
| Exceptions | -Admin not logged in :System Redirect to login page  -Missing or invalid input :System displays “Please complete all required fields” -This class already exists in the schedule: System displays “Class already exists” |

**5- Edit or delete an existing Gym Class**

|  |  |
| --- | --- |
| Use case name | Edit or delete an existing Gym Class |
| Actor | Admin |
| Trigger | Admin clicks 'edit class' from admin interface |
| Preconditions | Admin must be logged into the system |
| Main steps | 1. Admin logs into system 2. Admin clicks on 'Edit class'  3.System displays a list of existing classes  4.Admin modifies class name or type or date or time or capacity or chooses to delete the class  5.The system locks the class temporarily to prevent bookings during the update  6.Admin submits the form 7.System saves the class and updates the class schedule |
| Postconditions | The selected class is either updated or removed, and the schedule reflects the changes accordingly |
| Exceptions | -Admin not logged in :System Redirect to login page  -Missing or invalid input :System displays “Please complete all required fields”  -Class has already started : System displays “Cannot edit or delete a class that has already started” |

# **3.5 Requirements validation**

Verifying and validating the system requirements is a key part of the documentation process. It ensures that the system is being built correctly and aligns with the actual needs of gym members and staff. It also makes sure the system can be tested effectively for both functional and non-functional performance.

To carry out this process, we first developed specific functionalities to test and verify each system requirement. Then, the system was reviewed by the target users to confirm that it met their expectations and followed the initial discovery phase.

During validation, the system was fully tested after every release and development cycle. The client remained involved throughout the process to make sure the system was heading in the right direction.

Customer feedback played a major role in confirming the accuracy of system features. It helped the development team improve performance, minimize errors, and ensure a smooth system launch without unexpected issues.

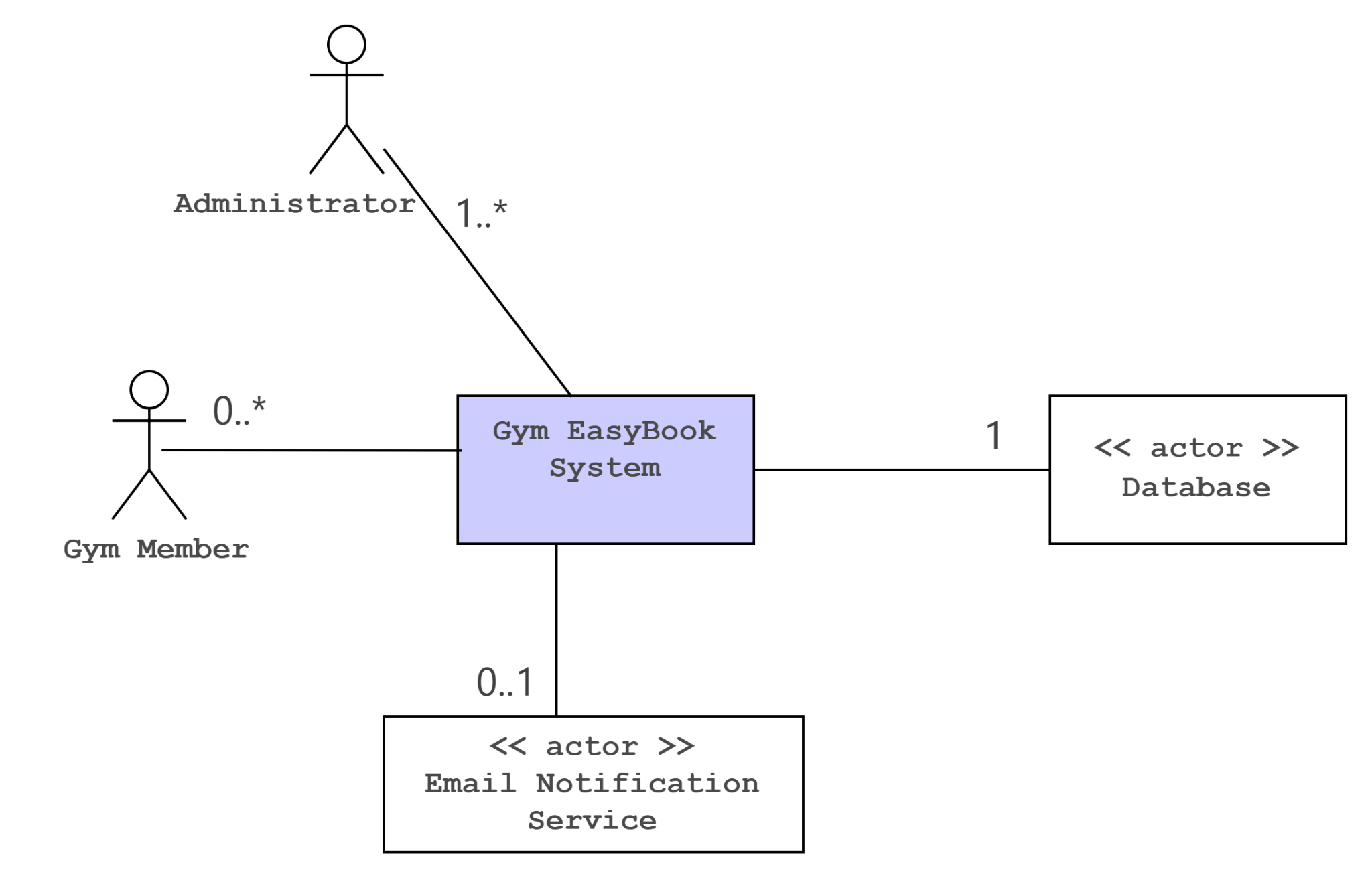
# **3.6 Requirements management**

# The Agile methodology was used in the development of Gym EasyBook, making it simple to adapt to needs changes. No extra procedures are required if the requirements change before development starts because it is simple to incorporate the changes. The impacted components will be updated to meet the new requirements if anything changes while the system is being created. The relevant components will be modified in response to feedback or new requirements if the system undergoes modifications after it is deployed.

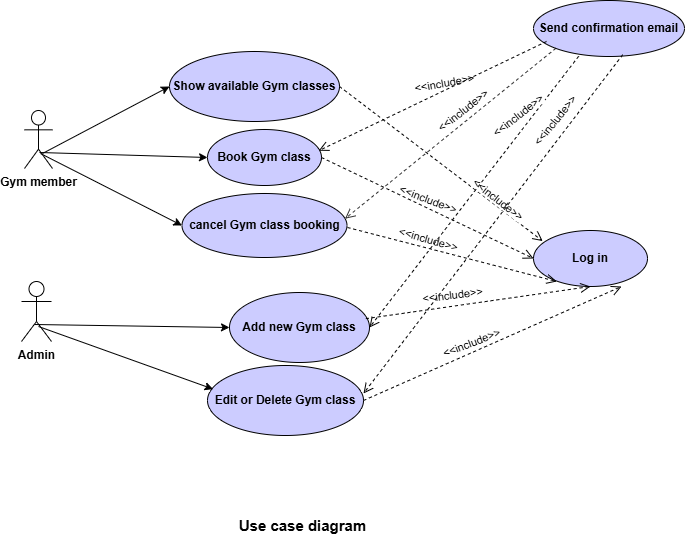
Chapter Four

System UML Design

1.Static context diagram

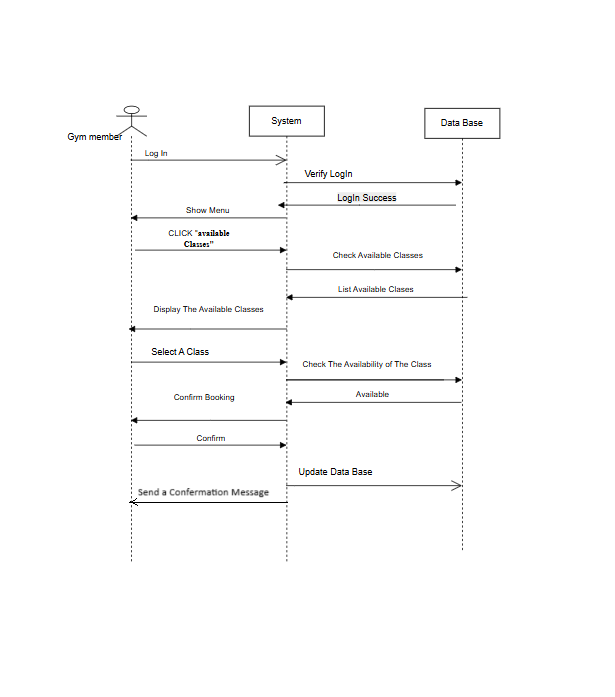


2.Use case diagram

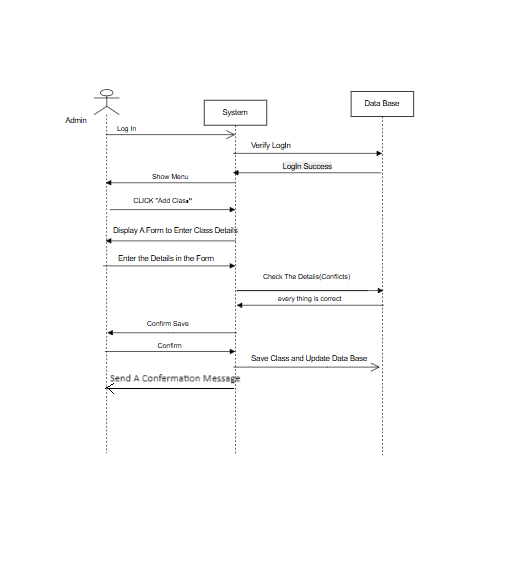


**3. Sequence diagrams**

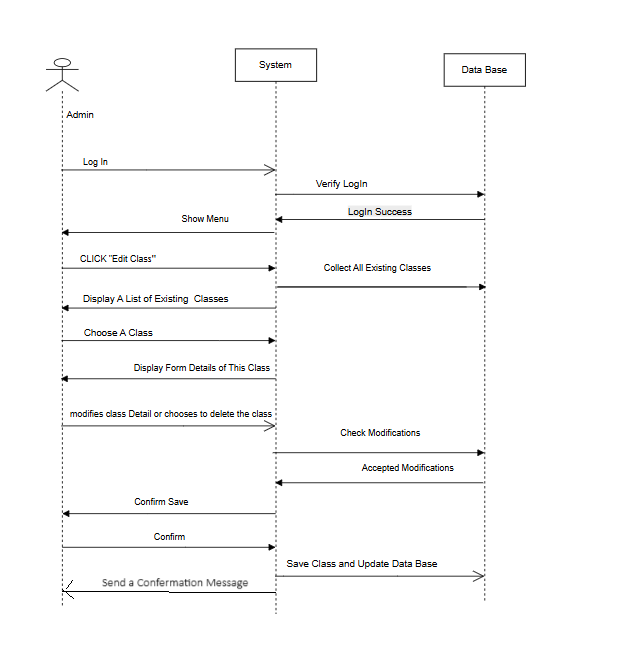
Book a gym Class



Add new gym class

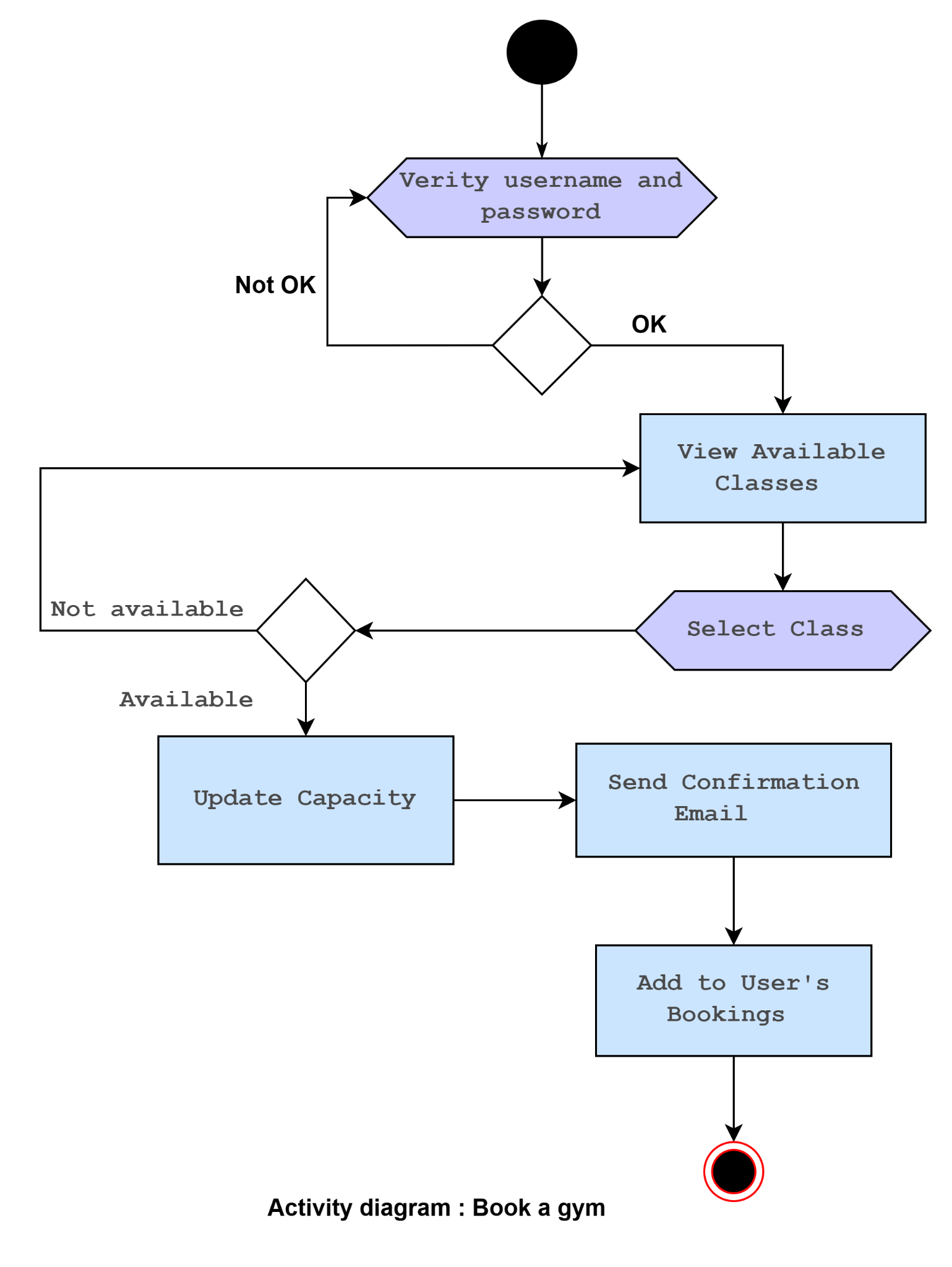


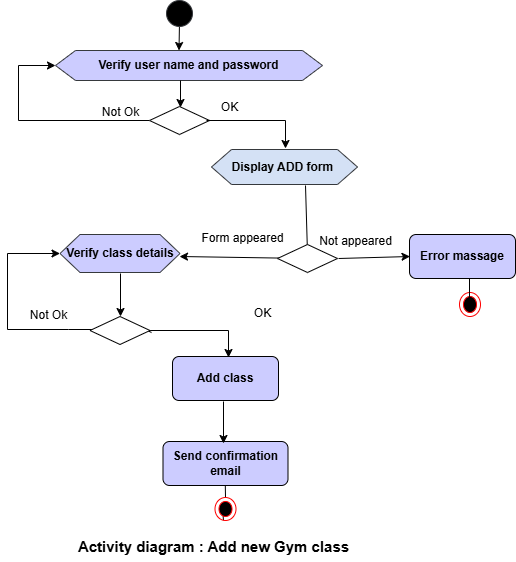
Edit or delete existing class



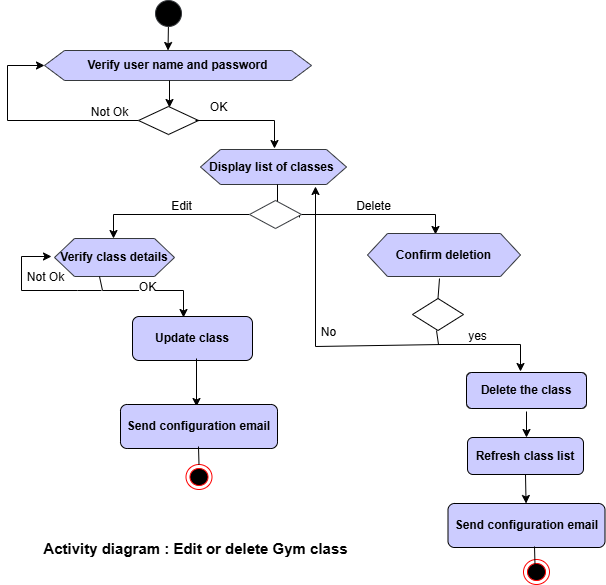
**4. Activity diagrams.**

Book a gym



Add new gym class

Edit or delete existing class



5. Class diagram

