

ZFit

GYM Management System

Final Project Report



Sri Lanka Institute of Information Technology IT2080

Information Technology Project

ITP25_B2_W225

October 2025

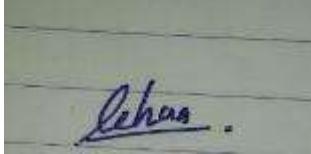
IT Number	Student Name	Student E-mail Address	Contact Number
IT 23 5621 10	H R G N Karunathilaka	IT23562110@my.sliit.lk	0764891241
IT 23 5418 70	N M B M L B Nawaratne	IT23541870@my.sliit.lk	0763541666
IT 23 5804 80	K B P Kavisika	IT23580480@my.sliit.lk	0718827129
IT 23 7536 62	O A l De Silva	IT23753662@my.sliit.lk	0740357412
IT 23 7414 78	B Dhayabari	IT23741478@my.sliit.lk	0770772330

Declaration

This project report is our original work and the content is not plagiarized from any other resource.

References for all the content taken from external resources are correctly cited. To the best of our knowledge, this report does not contain any material published or written by third parties, except as acknowledged in the text.

Authors:

<u>Author SID</u>	<u>Author name</u>	<u>Signature</u>
IT 23 5621 10	H R G N Karunathilaka	
IT 23 5418 70	N M B M L B Nawaratne	
IT 23 5804 80	K B P Kavisika	
IT 23 7536 62	O A l De Silva	
IT 23 7414 78	B Dhayabari	

Date: 18/10/2025

Abstract

ZFit Gym Management System is a sophisticated online portal that will be developed to automate and simplify the daily running of ZXcuses Gym in Colombo. The project was launched to solve the problems of inefficiency of manual record-keeping, fragmented management processes, which caused delays, errors, and user experience. ZFit builds in basic capabilities of membership, attendance and access control with QR codes, staff and shift scheduling, trainer booking, facility booking and secure payment processing with local gateways.

MERN stack MongoDB, Express.js, React and Node.js and Next.js and Tailwind CSS as the frameworks were used in order to design and implement the system with optimized performance and responsive user interfaces. Agile methodology was used in all parts of the development lifecycle and with the help of modern tools, Figma, Postman, GitHub, Jira, and PayHere. The outcome is a scalable, user-friendly and secure Software as a Service (SaaS) application that offers a high level of accuracy in operational processes, enhances customer satisfaction and real time data insights to be available to managers in making decisions.

Moving ZXcuses Gym out of the old manual system and into a fully digitalized facility helps ZFit increase the interaction of members, decrease the administrative load, and makes the gym competitive in the new reality of the modern fitness industry.

Acknowledgement

It is our greatest pleasure to say that we have been able to mature our project successfully, ZFit Gym Management System which was created by ZXusses Gym and supported by number of individuals. Our sincerest gratitude to our ITP module supervisor and mentors who have helped us with support feedback and encouragement through the development process. Their feedback played critical role improving our gym management system and making it a realworld solution that provides real business value.

We highly appreciate the cooperation and assistance of the management of ZXusses Gym and staff. Their willingness to discuss their observations of operation and the real world problems in the sphere of fitness gave us a very solid background on how to create a solution that best addresses the real needs of both the gym administrators and its members. The collaboration was critical in transforming ZFit into a functional and trustworthy platform.

Lastly, we recognize the effort and team spirit of our group members whose contribution and support turned out to be a success in this project. We also owe our friends and families a lot of service since they consistently encouraged us, remained patient and motivated, which enabled us to remain focused and to accomplish our goals successfully.

Table of Contents

Declaration.....	i
Abstract.....	ii
Acknowledgement	iii
Chapter 1. Introduction	1
1.1 Background	1
1.2 Problem and Motivation.....	1
1.2.1 Problem	1
1.2.2 Motivation.....	2
1.3 Literature review- Summary	3
1.4 Aim and Objective	4
1.4.1 Aim	4
1.4.2 Objectives	4
1.5 Solution overview	5
1.6 Methodology	5
1.7 Structure of the Report.....	9
1.8 Git Hub Repository	10
Chapter 2. Literature review	10
Chapter 3. Requirements.....	12
3.1 Stakeholder Analysis	12
3.2 Requirements Analysis	13
3.2.1 Functional Requirements	13
3.2.2 Non-Functional Requirements	16
3.3 Requirements Modeling.....	16
Chapter 4. Design and Development	17
4.1 Onion Diagram.....	17
4.2 Data Flow Diagrams	18
4.3 Use case Diagram	19
4.4 ER Diagram	20
4.5 Relational Schema	21
4.6 High Level System Design Diagram	22
4.7 Use Case, Sequence Diagrams, Activity Diagrams	23
4.7.1- H R G N Karunathilaka - IT 23 5621 10	23
4.7.2- N M B M L B Nawaratne - IT 23 5418 70	26

4.7.3- K B P Kavisika - IT 23 5804 80	29
4.7.4- O A I De Silva - IT 23 7536 62	32
4.7.5 - B Dhayabari - IT 23 7414 78.....	35
Chapter 5. Testing.....	38
5.1 Karunathilaka H R G N IT 23 5621 10	38
5.2 Nawaratne N M B M L B IT 23 5418 70	42
5.3 K B P Kavisika IT 23 5804 80	47
5.4 B Dhayabari IT 23 7414 78.....	50
5.5 O.A.I. De Silva IT 23 7536 62	55
Chapter 6. Evaluation and Conclusion.....	60
6.1 Evaluation	60
6.2 Conclusion	61
References.....	62
Appendix A – Work Done by each Member	63
Appendix B – Contribution to Final Report.....	64

LIST OF TABLES

Table 1 - List of Abbreviations.....	viii
Table 2 - Literature review	11
Table 3 - Stakeholder Analysis	13
Table 4 - Table of Test Case 1	38
Table 5 - Table of Test Case 2	39
Table 6 - Table of Test Case 3	40
Table 7 - Table of Test Case 4	41
Table 8 - Table of Test Case 5	43
Table 9 - Table of Test Case 6	44
Table 10 - Table of Test Case 7	45
Table 11 - Table of Test Case 8	46
Table 12 - Table of Test Case 9	47
Table 13 - Table of Test Case 10	48
Table 14 - Table of Test Case 11	49
Table 15 - Table of Test Case 12	50
Table 16 - Table of Test Case 13	51
Table 17 - Table of Test Case 14	53
Table 18 - Table of Test Case 15	54
Table 19 - Table of Test Case 16	55
Table 20 - Table of Test Case 17	56
Table 21 - Table of Test Case 18	57
Table 22 - Table of Test Case 19	58
Table 23 - Table of Test Case 20	59
Table 24 - Appendix A – Work Done by each Member.....	63
Table 25 - Appendix B – Contribution to Final Report.....	64

LIST OF FIGURES

Figure 1 - Onion Diagram.....	17
Figure 2 - Data Flow Diagrams	18
Figure 3 - Use case Diagram.....	19
Figure 4 - ER Diagram.....	20
Figure 5 - Relational Schema.....	21
Figure 6 - High Level System Design Diagram.....	22
Figure 7 - Use Case 1	23
Figure 8 - Sequence diagram 1	24
Figure 9 - Activity Diagram 1	25
Figure 10 - Use Case 2	26
Figure 11 - Sequence Diagram 2	27
Figure 12 - Activity Diagram 2	28
Figure 13 - Use Case 3	29
Figure 14 - Sequence Diagram 3	30
Figure 15 - Activity Diagram 3	311
Figure 16 - Use Case 4	32
Figure 17 - Sequence Diagram 4	33
Figure 18 - Activity Diagram 4	34
Figure 19 - Use Case 5	35
Figure 20 - Sequence Diagram 5	36
Figure 21 - Activity Diagram 5.....	37

List of Abbreviations

Abbreviation	Description
ER	Entity Relationship Diagram
DBMS	Database management System
SD	Sequence Diagram
SDLC	Software Development Life Cycle
SQL	Structured Query Language
UI	User Interface
QR	Quick Response
API	Application Programming Interface

Table 1 - List of Abbreviations

Chapter 1. Introduction

1.1 Background

Zfit is a comprehensive gym management system developed for ZXCuses Gym in Colombo. ZXCuses is a modern and customer focused gym which locates in Colombo and plays an important role in health, fitness, mental well-being and personal development through personalized workout plans and modern exercise facilities. As a high-end fitness center, they are committed to providing a supportive and professional environment that inspires individuals to achieve their personal fitness goals. Whether someone is new to their fitness journey or an expert, they offers a wide range of services designed to meet individual needs, placing greater emphasis on both personal physical and mental health, making the gym an essential part of everyday life for many. They provides members with personal trainers for their individual fitness goals and mental wellbeing. Their certified trainers create structured, result driven programs that help members to train smarter but not harder. Among the services provided by ZXCuses Gym are personalized oneon-one training, senior fitness programs, pre and postnatal sessions, body conditioning, and muscle rehabilitation. Gym includes the latest state of the art fitness machines and technologies which encourage members with their maximum performance and safety. Also, ZXCuses gym go beyond the gym floor with Healthy nutrition plans, and Members can get nutrition and food guidance. ZXCuses gym continues to rely on manual record keeping and manual management strategies leading to inefficiency in day-to-day business functions. Without a proper system for membership, inventory, reservations, attendance, payment, and schedule management, all members are inconvenienced by delays and errors. Avoid those challenges, we decided to develop a comprehensive GYM Management System finetuned for ZXCuses gym. This System is fully automated with following key functions.

1.2 Problem and Motivation

1.2.1 Problem

In the finest industry today, gym is finding it harder to efficiently manage its day-to-day operations. These are the challenges which may be affected by their daily operations. It can be both small to Medium sized depend on the manual or semi digital systems such as paper records, spreadsheets, or disconnected software for managing memberships, attendance, trainer schedules, payments, and fitness tracking. This fragmented process often leads to errors, delays, and poor customer experience. The existing procedure is too time-consuming for both clients and employees. For instance, when a client signs up, his or her data must be manually entered to billing, attendance, and communication. Trainers find it hard to keep members' plans personal because they do not have centralized access to client data and information. Handling bookings for classes or personal training also can have too much manual coordination. This can lead to results in double

booking, missed sessions or customer dissatisfaction. It can be challenging for the growth of each company with other competitors. There is an increasing demand for gyms to offer such as digital services such as mobile apps, real time updates, online workout plans, and monitoring and tracking progress. Without an efficient gym management system, these services are hard to offer. So, the integrated system should be included with the solutions to prevent these problems. Especially with the competitiveness with the fitness market there is a chance that gyms will lose their members to more tech savvy competitors who can provide a better smoother and enjoyable experience for their customers. To get more ideas about this, the main problems are:

- Inefficient manual or disconnected systems
- Difficulty in managing client data and bookings
- Limited ability to offer digital services
- Poor user experience and operational delays

These are some matters that hinder a gym's ability to increase their retain members and grow in this fast-paced environment with more efficiency.

1.2.2 Motivation

Having a Gym Management System provides offers many benefits to the gym owners, employees/staff and members of the gym. The motivation of this automate process is to reduce manual workload and increase the overall experience to be improved. It will enhance the user experience as well. For all the club owners and managers, the system provides centralized management of control over all the principal activities (core activities) membership management, class scheduling, billing and reporting, trainer availability. This automation reduces errors, frees administrative workload and allows the staff to deliver a better service. Trainers also enjoy and get benefits having live access to member profiles, fitness goals, attendance history and performance data. This gives the chance for them to do better tailor exercise programs and client progress in real time. This improves communication between trainers and members through messages or alerts /notifications. Members are more convenient and engaged. They have more features like online check-in, class booking, attendance tracking, personalized dashboards, such a system can also allow better experiencing, they gain more control over their fitness journey. A mobile friendly system can also allow receive updates, reminders, and access virtual training plans from anywhere /anytime. Overall, the Gym Management System addresses important pain points and provides measurable improvements/advantages such as

- Time savings through automation
- Increased accuracy in bookings and billing
- Better data insights for decision-making
- Higher member satisfaction and retention

- Competitive edge in the fitness market

Through this system, a gym can evolve from a traditional gym into a smart, user-centric service provider. It supports current needs, supports, operational needs but it also opens up possibilities and prepares the gym for future development and innovation.

1.3 Literature review- Summary

The literature review draws a comparison between ZFit Gym Management System and top fitness management applications available in the market - Mindbody, Glofox, Zen Planner, Gym Master, and Virtuagym based on major operational and user-friendly applications.

As the analysis shows, ZFit has a more extensive and more localized set of features than their competitors. Unlike most international sites which focus on generic international market, ZFit offers complete local integration with Sri Lankan payment and SMS gateways with support of LKR currency which will reduce forex charges and make ZFit more accessible to local gyms.

Another feature that is noticeable in ZFit is its use of a comprehensive manager dashboard, shift management and staff management modules, which are either non-existent or partially accessed in other systems. Membership management with expiry control, resource reservation without conflict, and attendance management with QR code are some of the features that make ZFit more appropriate to small to mid-size gyms that require a full-fledged automated platform.

Although international platforms such as Mindbody and Glofox offer advanced member portals and booking opportunities, they do not offer local operations and payment options. Comparatively, ZFit provides a full package branded solution that is a fully customizable one, capable of addressing the administrative and member oriented requirements effectively within the Sri Lankan fitness industry setup.

1.4 Aim and Objective

1.4.1 Aim

The main objective of this project is to transform the GYM management system into an efficient and user friendly full featured digital system. The need for such a digital system has arisen due to problems such as manual record keeping and outdated management methods, delays, errors, and the need for telephone or face to face discussions. This project is expected to automate the management of the required operations such as member registrations, membership management, employee shift management, staff management, attendance management, resource allocation, payment processing, etc. through a digital system. This system will enable the company to provide systematic, accurate and convenient service to its stakeholders. This Zero excuses system facilitates this operation by incorporating modern features such as registering attendance through a passcode, secure payments, restricting access to the system based on the role of system users, improving the security and user friendliness of the system, and providing the necessary data for account control, reporting, and decision making.

1.4.2 Objectives

- To develop a user friendly and responsive web application that allows members to register, manage their personal profile, purchase and renew their membership from any device.
- To provide a membership management system that allows the management board to create, update, activate and deactivate membership packages and provide advance notice to members on membership termination.
- To implement a system for collecting member data through a QR code and a manual registration option to ensure accurate and assessed member data is recorded
- To create a system that allows managers to assign shifts and control employee shifts, approve or cancel leave, generate reports and allow employees to track their shifts and request changes.
- Creating a system that allows members to book trainers and facilities and include a mechanism to prevent double bookings
- Incorporating a highly secure payment system with payments by credit debit card, discount codes and automatic receipt issuance
- The system is designed to provide role-based access controls and data security

1.5 Solution overview

ZFit gym management system is built to automate and computerize the running of a gym with key functionalities of the management system of a gym including membership management, reservation management, trainer management, facility management and fund management. The system is designed to be commercialized as a Software-as-a-Service (SaaS) platform which can be implemented by gyms, fitness centers, and wellness institutions of all sizes.

1.6 Methodology

Our project aim's to deliver a reliable, scalable and industry ready gym management system that can handle up to 1000 members with no issues. The system will handle major business functions like member & membership, staff & shift management, attendance & access control, payments & invoicing, resource reservation & bookings management and other core features using modern web development technologies, tools and best practices.

Requirement Engineering Methods

To get a better understanding of the requirements of the stakeholders, variety of requirements engineering methods were conducted by us. Through those requirement engineering methods we have gained insights about the problems with the current system and business functions we have to implement in the new gym management system.

Stakeholder Interviews:-

Interviews were conducted with the managers, staff and trainers to get a better understanding of the difficulties they face with the current system which is a manual record keeping system. These interviews let us get a better understanding of the user requirements and functionalities that needed to be included in our system.

Observation of Current Process:-

The current process which is a manual record keeping method, was carefully observed by our team and we were able to collect impactful information. We were able to identify the systemic drawback of the current system and gained insight on how we can provide a more smoother experience for the stakeholders involved.

User Stories and Use Cases:-

To refine the collected data about the user requirements from a user perspective, we used user stories and use case diagrams methods. This way we were able to get a better insight into specific needs and expectations of the users (members, staff and managers). This way we were able to convert the data about real world workflow into well refined business functional requirements.

Alternatives – Surveys, IEEE 830 SRS, MoSCoW prioritization and Kano Model were other identified requirement engineering methods. But for this small scale project they were too complex and higher time/cost consuming.

Design Methods

- **Wireframing with Figma :-** Used the industry standard UI/UX design tool Figma to design wireframes to design user-friendly user-interfaces.
- **Database Schema Design with Draw.io :-** Entity relationship diagrams were designed for schema planning for MongoDB using draw.io.
- **System Diagram with Draw.io :-** A high level system diagram was designed with major system components and interactions included to better visualize the system's architecture. This process helps the team and the stakeholders better understand the system flow
- **Data Flow Diagram with Draw.io :-** To ensure the proper data handling DFD were used to map how data moves within the system.
Flowchart with Draw.io :- To better understand the sequence of operations and workflow, flowchart was designed.
- **Onion Diagram with Draw.io :-** To better understand the stakeholders and components involved with the system, this diagram was drawn.

Development Tools and Technologies

We have chosen MERN stack for modern industry standard web development in combination with Next.js for better optimized frontend

1. Next.js

Next.js is a React-based framework for building fast, scalable web applications. It offers features like server-side rendering (SSR), static site generation (SSG), API routes, and optimized performance. Ideal for SEO and large-scale apps.
Reference: nextjs.org [1]

2. React

React is a JavaScript library developed by Facebook for building user interfaces. It allows developers to create reusable UI components and manage application state efficiently using a virtual DOM.

Reference: reactjs.org

3. Express.js (ex)

Express.js is a minimal and flexible Node.js web application framework. It simplifies backend development by handling routing, middleware, and HTTP requests. Commonly used in RESTful APIs.

Reference: expressjs.com

4. Node.js

Node.js is a runtime environment that allows JavaScript to run on the server side. It's event-driven, non-blocking, and ideal for scalable, real-time applications.

Reference: nodejs.org

5. MongoDB

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. It's schema-less, making it ideal for agile development and applications that require scalability.

Reference: [mongodb.com](https://www.mongodb.com)

Alternatives – For developing frontend there were many other options such as Vue.js in combination with Nuxt.js. But the bigger community around Next.js and React provided a better support and resources for development. For backend development there were many other good alternatives like Java with Spring boot and PHP with Laravel. But our familiarity with JavaScript based and being able to create the whole application in JavaScript made the developer experience better as well as reduce the complexity of handling different programming languages. So we chose ExpressJs with NodeJs runtime for our backend. For the database we had two options. No-SQL and SQL databases. But we went with No-SQL database MongoDB because of its simplicity and its flexibility.

Apart from this tech stack we use following tools and technologies in our project.

1. Visual Studio Code (VS Code)

Visual Studio Code is a lightweight yet powerful source-code editor with support for JavaScript, Node.js, and modern frontend frameworks. Its integrated terminal, debugger, and extensions make it ideal for full-stack web development projects

like this gym management system. Its speed and customization options make it preferred by many developers (Microsoft, 2023).

Reference: <https://code.visualstudio.com>

2.Git and GitHub

Git is a distributed version control system, and GitHub is a cloud-based platform for hosting and managing Git repositories. Together, they enable team collaboration, version tracking, and seamless code integration. For this project, GitHub supports task management, issue tracking, and deployment pipelines, which streamline agile development processes (Chacon & Straub, 2014).

Reference: <https://github.com>

3.Postman

Postman is an API development and testing tool that simplifies backend and validation. It allows developers to structure, document, and test RESTful API endpoints efficiently. Postman's user-friendly interface and support for collections and environments make it ideal for ensuring the reliability of APIs in the gym management system (Postman, 2024).

Reference: <https://www.postman.com>

4.Tailwind CSS

Tailwind CSS is a utility-first CSS framework that accelerates frontend development by allowing developers to style components directly within HTML. It reduces the need for writing custom CSS and improves design consistency. For this project, it ensures rapid UI prototyping and responsive design, which are essential for member and staff-facing interfaces (Tailwind Labs, 2024).

Reference: <https://tailwindcss.com>

5.PayHere

PayHere is a Sri Lankan online payment gateway that supports card payments, mobile wallets, and bank transfers. It offers secure and easy integration for small businesses. For this gym management system, PayHere is suitable due to its local relevance, ease of use, and support for recurring membership payments (PayHere, 2024).

Reference: <https://www.payhere.lk>

6.Mongoose

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides schema-based solutions to model data and includes built-in validation,

type casting, and business logic hooks.
Reference: mongoosejs.com

7.Jira

Jira is a popular project management tool developed by Atlassian, widely used for tracking tasks, bugs, and project progress, especially in Agile and Scrum environments. It helps teams plan sprints, manage backlogs, and visualize workflows using boards and reports. Jira improves team collaboration, transparency, and productivity by offering real-time tracking and automation features.

Reference: atlassian.com/software/jira

1.7 Structure of the Report

The present report is a detailed report about the process of design, development and implementation of the ZFit gym management system, which is to be developed to serve ZXcuses gym in Colombo. The chapters are based on major phases of the project lifecycle such as conceptualization and the system analysis, implementation, testing, and evaluation.

Chapter 1: Introduction

Narrates about the ZFit project, background as well as issues that were experienced with the current manual system and the reason why a digital solution was necessary. It describes the purpose of the project and its goals and the overview of the proposed solution, as well as provides a summary of the general structure of the report and a connection to the repository on GitHub. In the methodology section that requirement-gathering methods, design techniques, and technologies, e.g., MERN stack, Next.js, Tailwind CSS, and such tools as Figma, Postman, GitHub, and Jira, were employed.

Chapter 2: Literature Review

Gives a comparative analysis of ZFit and other popular gym management software in the world (e.g., Mindbody, Glofox, Zen Planner, GymMaster, Virtuagym). It brings the limitations that exist and the necessity of a localized Sri Lankan system to light.

Chapter 3: Requirements

Detects and identifies system requirements based on the gathered requirements from stakeholders like managers, trainers, and members. It describes both functional and non-functional requirements with the help of a stakeholder analysis and requirement models (use-case diagrams, ER diagrams, and activity flows)

Chapter 4: Design and Development

Describes the design process by use of visual models like Onion Diagrams, Data flow Diagrams (DFDs), Use-Case Diagrams, ER Diagrams and Relational Schemas. It encompasses also individual sequence and activity diagrams of each team member which show module specific development in the system architecture.

Chapter 5: Testing

Records the test procedures and plans that have been applied in checking the performance and reliability of each of the modules. Each developer (user management, inventory, payments, booking and fitness tracking) is included with inputs, expected and actual output and pass/fail to indicate system correctness.

Chapter 6: Evaluation and Conclusion

Assesses the system with reference to original goals, strengths, weaknesses, and user reviews. It has ended up by how ZFit has been successful in substituting the manual procedures of ZXcuses Gym, enhanced efficiency, and presented a scaleable SaaS-related gym management resolution.

Appendices

Appendix A: Group work done by every member.

Appendix B: The role of every member in the final report documentation.

1.8 Git Hub Repository

Git hub link - <https://github.com/gaindunuhansith/ZFit>

Chapter 2. Literature review

Feature	Our Product	Mindbody	Glofox	Zen Planner	Gym Master	Virtua gym
---------	-------------	----------	--------	-------------	------------	------------

Member Portal	<input checked="" type="checkbox"/> (Fully branded portal for registration, profile, bookings, payments)	<input checked="" type="checkbox"/>				
Manager Dashboard	<input checked="" type="checkbox"/> (Full control over all the gym functions in a centralized dashboard)			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Member Management	<input checked="" type="checkbox"/> (Comprehensive dashboard for members and staff)	<input checked="" type="checkbox"/>				
Membership Management	<input checked="" type="checkbox"/> (Fully customizable membership management with expiry and access control after expiry)	<input checked="" type="checkbox"/>				
Reservation and Booking Management	<input checked="" type="checkbox"/> (Ability to reserve any gym resource without conflicts)					
Attendance and Access Control	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	
Staff Management	<input checked="" type="checkbox"/> (Comprehensive staff management dashboard)					
Shift Management	<input checked="" type="checkbox"/>					
Local Integration	<input checked="" type="checkbox"/> (Local integration with local payment and SMS gateways)					
LKR Payments	<input checked="" type="checkbox"/> (Full support for LKR payments through many local payment methods bypassing huge forex fees and taxes)					

- Available

- Partial or Limited Availability

- Not Available

Table 2 - Literature review

In Conclusion,

There are many similar gym management systems available in the market today. Most of them offer multi-tenant SaaS applications and not tailor made to specific each gym needs. For our literature overview, we compared our solution with the industry leading gym management software according to their features and cost. This comparison helped us to gain insight into each platforms' strengths, identify common limitations and create a solution more suitable toward our clients needs than the alternatives in the current market.

Chapter 3. Requirements

This chapter presents the requirements gathered for the Online Library Management and Reservation System. It identifies the stakeholders, analyzes system functionality and quality expectations, and models the essential processes through diagrams.

3.1 Stakeholder Analysis

The stakeholders involved in the system and their responsibilities are described below.

Stakeholder	Description	Responsibilities / Expectations
Super Admin (Librarian)	The overall system administrator with full access to all modules.	Manage users, books, staff, inventory, and finance data; oversee operations; approve or delete records; generate summary and audit reports.
Inventory Manager	Manages the book inventory and ensures stock accuracy.	Add, update, and view books; monitor stock levels; handle damaged or missing books; generate inventory reports; ensure real-time data consistency.
Finance Manager	Handles the library's financial operations.	Manage book-related financial records such as purchase costs, damaged book expenses, and fines; generate financial reports and

		summaries; ensure accuracy in cost and value calculations.
Staff Member	Supports day-to-day library operations and manages Book Management and Decision Management.	Add and edit book records within assigned permissions; approve or reject member reservations; manage book issuance and returns; report damaged books or issues to the management.
Member (Student / User)	A registered user who interacts with the system for borrowing and reserving books.	Search, view, borrow, and reserve books; check due dates; renew or return borrowed books; receive notifications and updates about their activities.

Table 3 - Stakeholder Analysis

3.2 Requirements Analysis

The following sections outline both functional and non-functional requirements derived from stakeholder expectations and system objectives.

3.2.1 Functional Requirements

User Management

1. User Registration & Authentication
 - Users can log in using their credentials
 - Users must verify their account via email or OTP
 - Users can reset passwords via email if forgotten.
 - Role-Based Access Control
 - Define user roles- Admin (Librarian), Library Member
 - Profile & Account Management
2. Admins can manage all accounts, including suspending/deleting users
3. Role Based Access Control for the admins
4. Profile Management
5. Report Generation

6. Personalized, pre-scheduled alerts for due dates, reservations, or library events.

Book Management

1. Search & Filter Books
 - Enables users to search and filter books by title, author, category, or availability status for quick access.
2. Book Borrowing History Tracking
 - Keeps a record of all borrow and return activities, including dates and user information, for each book.
3. Book Reservation & Request System
 - Allows members to reserve available books or request unavailable books to borrow when restocked

Inventory Management

1. Add, Update, Delete Book Records
 - Staff can add new books and edit or delete existing records with key details like title, author, category, and quantity.
2. View and Search Inventory
 - Users and staff can view the list of available books and search by title, author, category, or ISBN.
3. Track Lost/Damaged Items
 - Staff can mark books as lost or damaged, and inventory will be adjusted automatically.
4. Low Stock Alerts (FOL Method)
 - The system will generate alerts using the First Order Level (FOL) formula: $FOL = (\text{Average Daily Usage} \times \text{Lead Time}) + \text{Safety Stock}$. Alerts notify staff when a book's quantity drops below this level.
5. Inventory Reports
 - Staff can generate reports on stock levels, low-stock items, and most borrowed books.

Finance Management

1. Automated fine calculation

- The system must calculate fines for overdue book returns based on fine rate.
- 2. Fine notification
 - Users should receive automatic notifications by email when a book is reserved, when book is due on date and when a fine is paid.
- 3. online fine payment
 - Users must be able to view fines and pay them online.
 - Users should receive digital receipts after payments.
- 4. Finance dashboard
 - Librarian should be able to view fine payments by user and date.
 - Librarian should be able to generate income reports from fines and subscriptions.
- 5. Invoice and receipt generation
 - After any payment the system must generate a downloadable digital invoice or a receipt.

Tables and PC reservation Management

- 1. User Authentication
 - Only registered users can access the reservation system.
- 2. View Availability
 - Users can view real-time availability of tables and PCs, including time slots and dates.
- 3. Make a Reservation
 - Users can select a time slot and reserve a table or PC for a specific date.
- 4. Modify or Cancel Reservation
 - Users can update or cancel their reservation before a predefined cutoff time.
- 5. Admin Approval (Optional)
 - Admin can optionally review and approve reservations (if needed for manual confirmation).
- 6. Reservation Conflict Handling
 - The system prevents double bookings by checking for availability before confirming a reservation.
- 7. Reservation Limits
 - The system enforces limits such as max hours per day/week per user or advance booking rules.

8. Notification System
 - Users receive email or system notifications upon successful reservation, cancellation, or upcoming booking reminders.
9. Reservation History
 - Users and admins can view past reservations.

3.2.2 Non-Functional Requirements

- Performance - Each CRUD or search operation should complete within 3 seconds.
- Security - Role-based access control and password encryption are required.
- Usability - Simple, consistent, and responsive user interface for all roles.
- Reliability - System ensures real-time updates and data synchronization across all modules.
- Maintainability - Modular and well-documented code to simplify future upgrades.
- Auditability - Every important action (add, delete, update, report generation) must be logged with timestamp and user ID. [2]

3.3 Requirements Modeling

To better visualize and structure the system requirements, the following models were used:

- Use Case Diagrams
Represent key user interactions with the system, such as managing book reservations, handling inventory updates, processing financial reports, and managing user accounts.
- Entity-Relationship (ER) Diagram
Illustrate how main data entities such as Books, Members, Staff, Inventory, Finance Records, and Reservations are related within the database to ensure data integrity and efficient management.
- Activity Diagrams
Detail the flow of actions in processes like book borrowing, inventory updates, and financial reporting to identify decision points and dependencies.
- Wireframes / UI Mockups
Designed for the main interfaces of the application (e.g., Login Page, Admin Dashboard, Book Reservation Page, Inventory Management Page) to ensure usability, clarity, and consistent design throughout the system.

Chapter 4. Design and Development

4.1 Onion Diagram

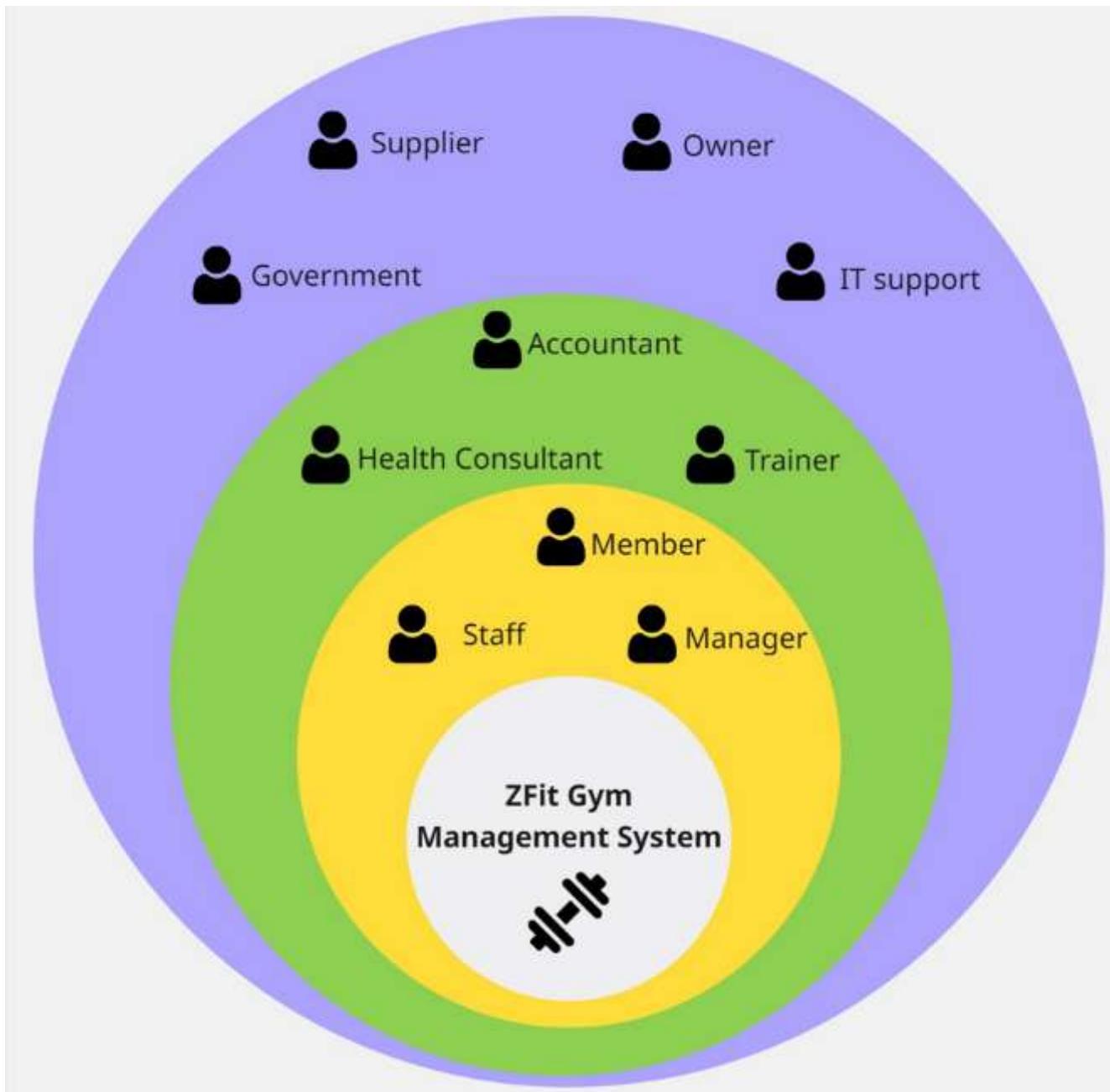


Figure 4.1 - Onion Diagram

4.2 Data Flow Diagrams

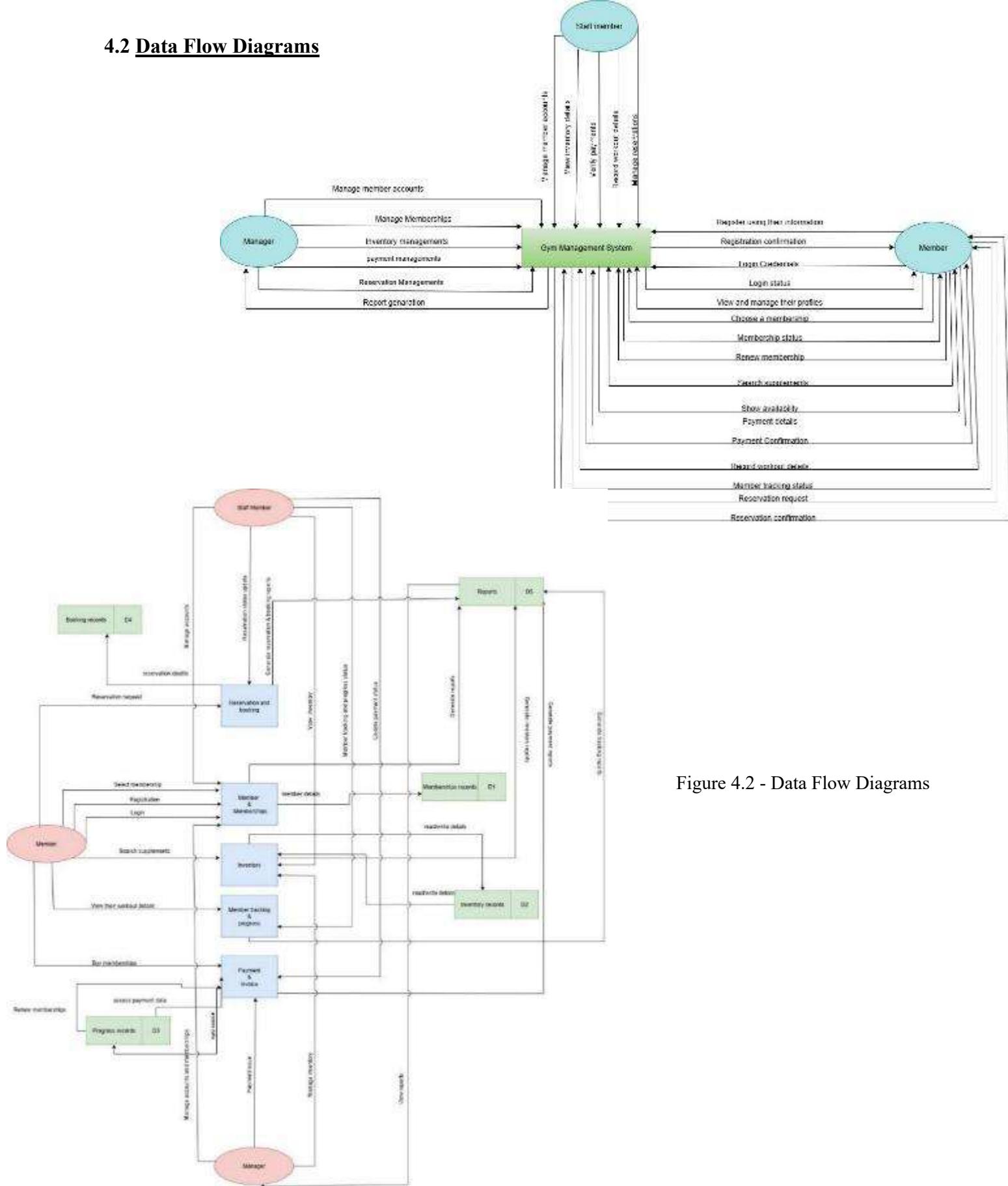


Figure 4.2 - Data Flow Diagrams

4.3 Use case Diagram

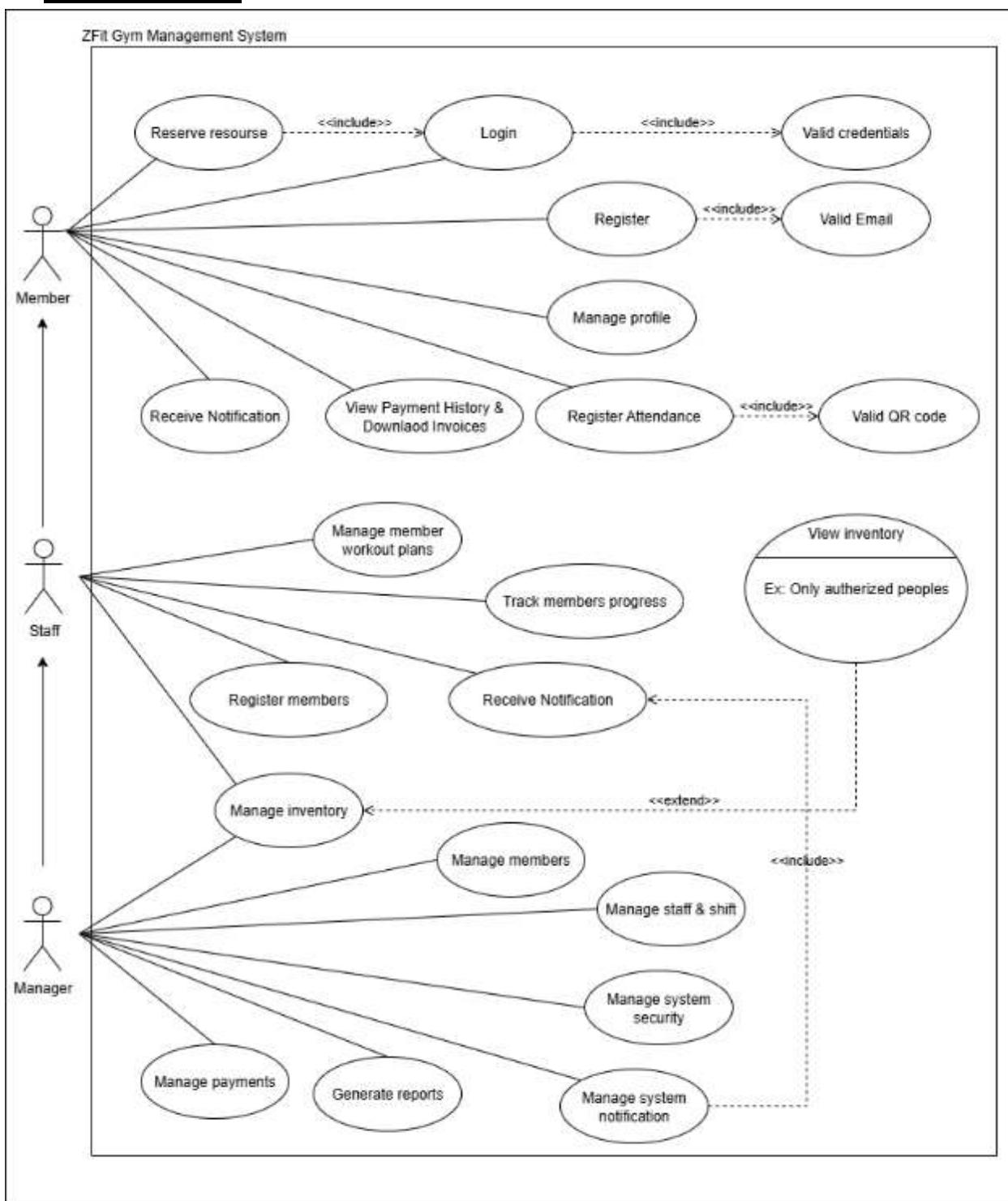


Figure 4.3 - Use case Diagram

4.4 ER Diagram

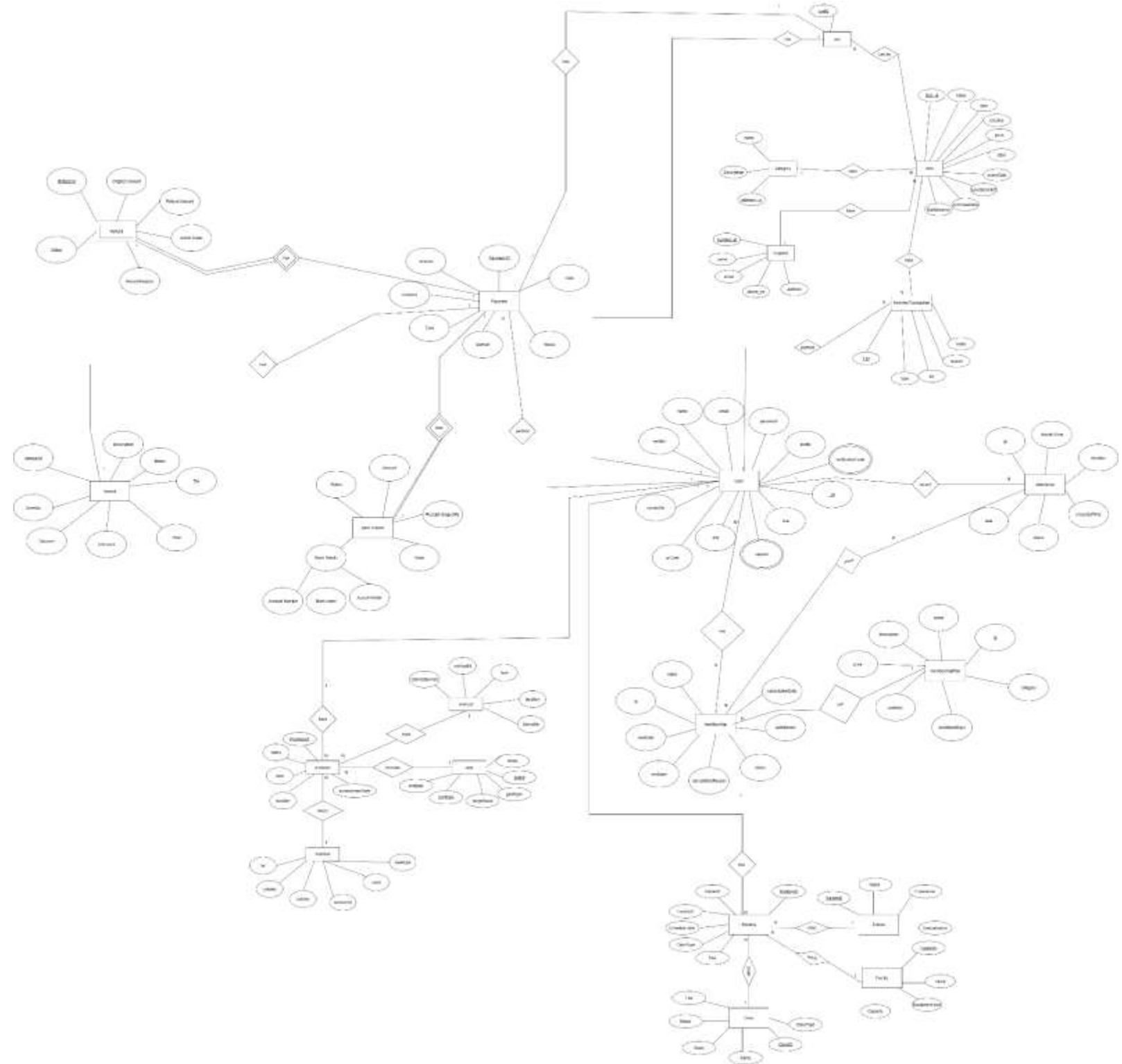


Figure 4.4 - ER Diagram

4.5 Relational Schema

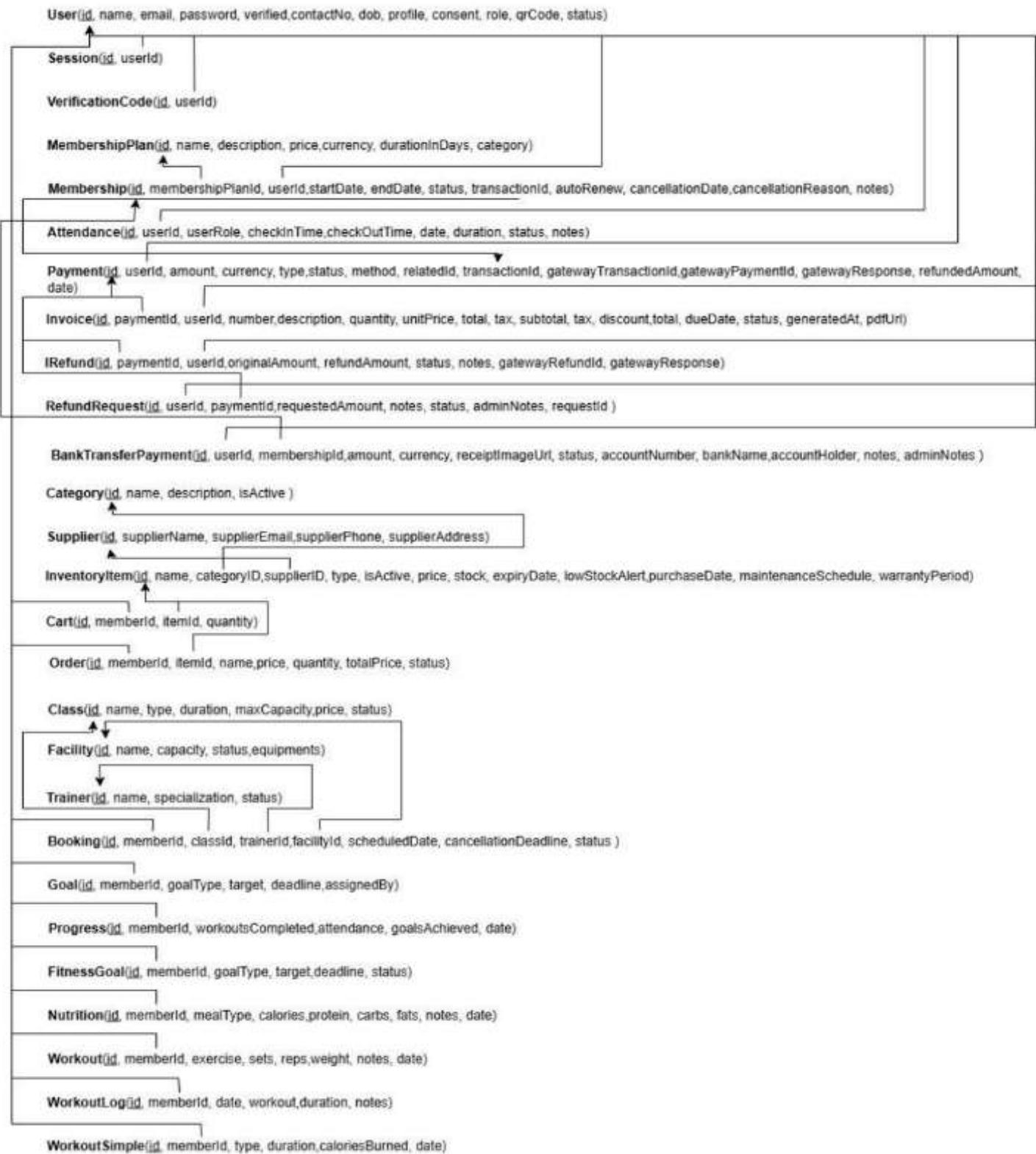


Figure 4.5 - Relational Schema

4.6 High Level System Design Diagram

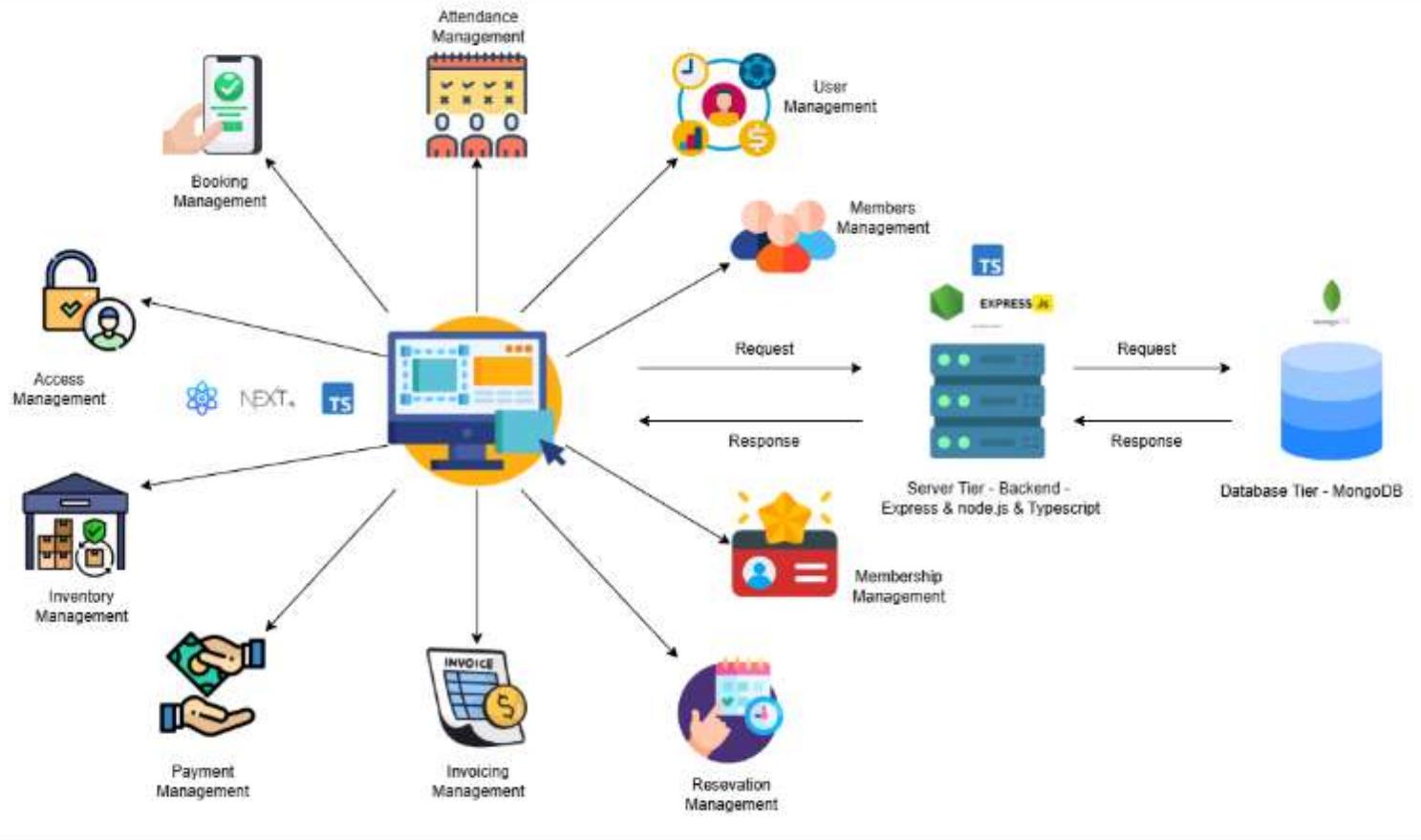


Figure 4.6 - High Level System Design Diagram

4.7 Use Case, Sequence Diagrams, Activity Diagrams

4.7.1- H R G N Karunathilaka - IT 23 5621 10

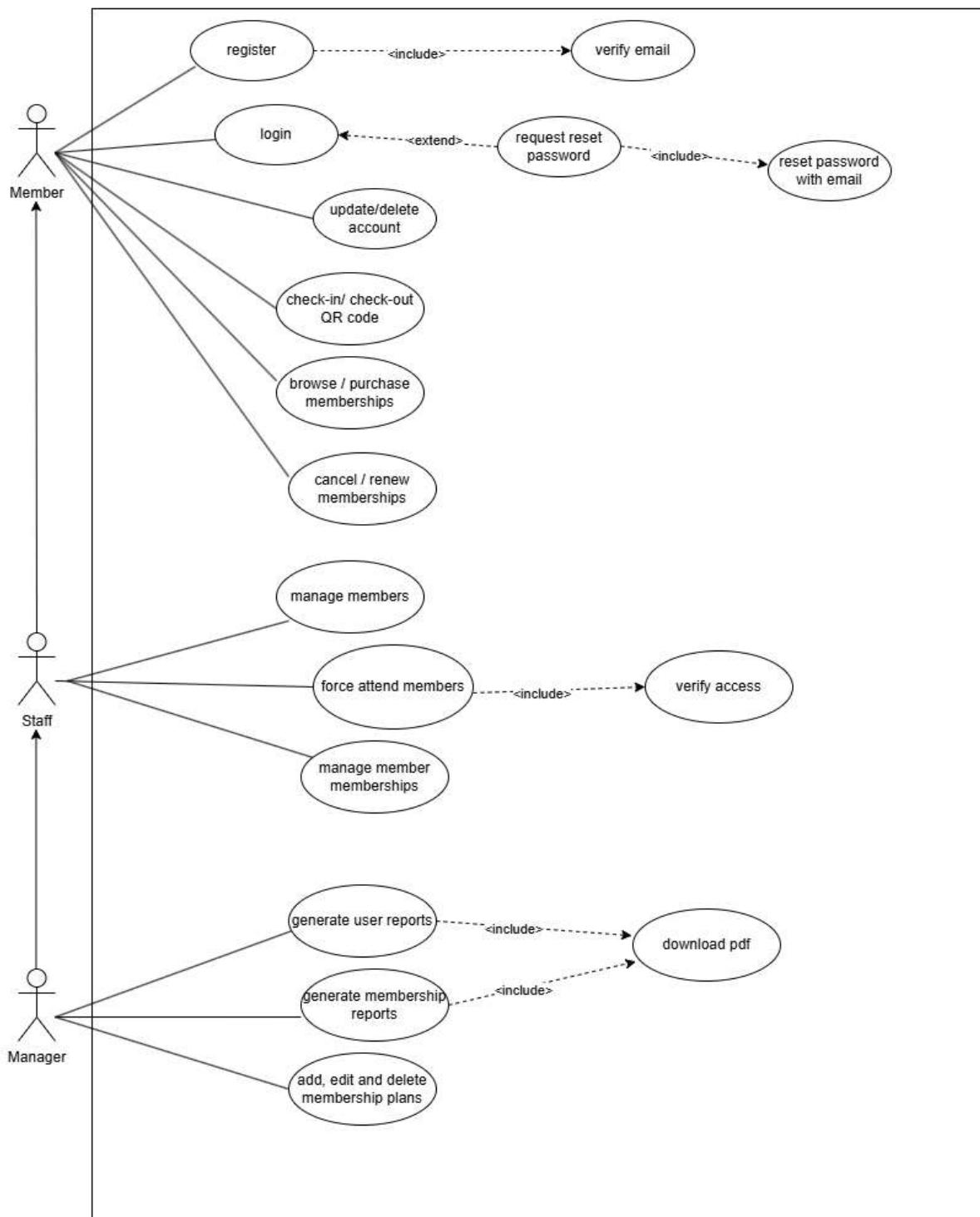


Figure 4.7 - Use Case

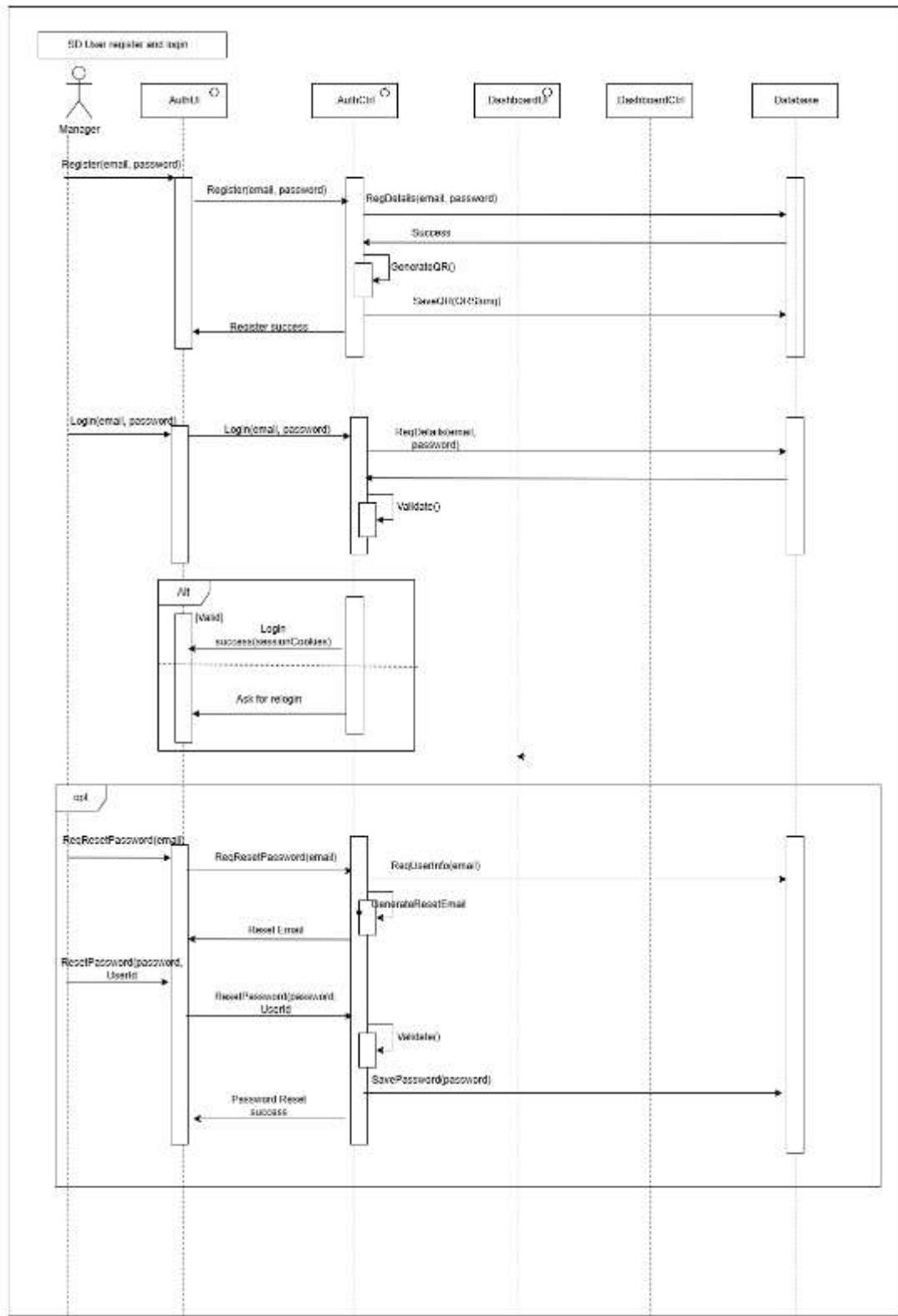


Figure 4.8 - Sequence diagram 1

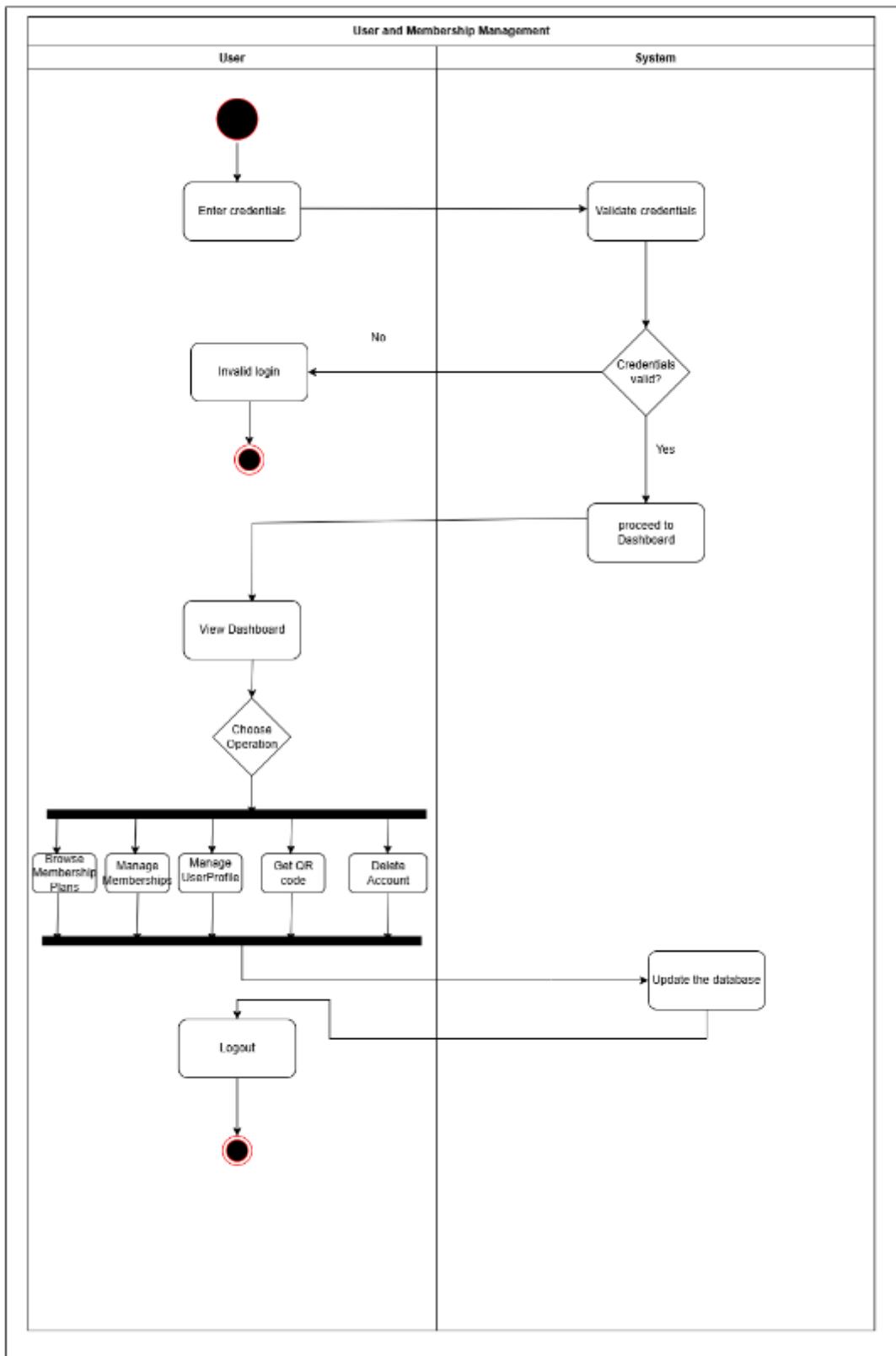


Figure 4.9 - Activity Diagram 1

4.7.2- N M B M L B Nawaratne - IT 23 5418 70

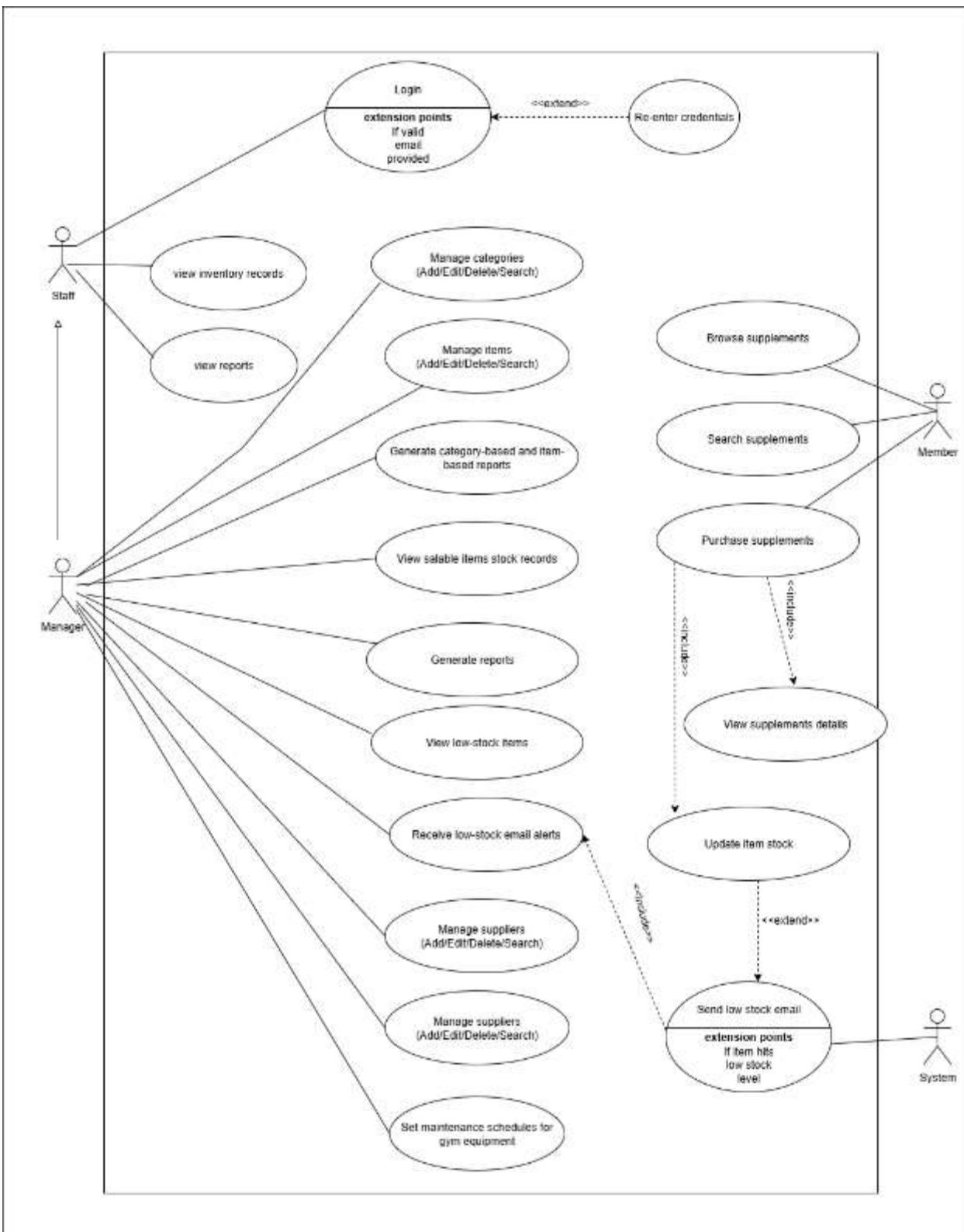


Figure 4.10 - Use Case 2

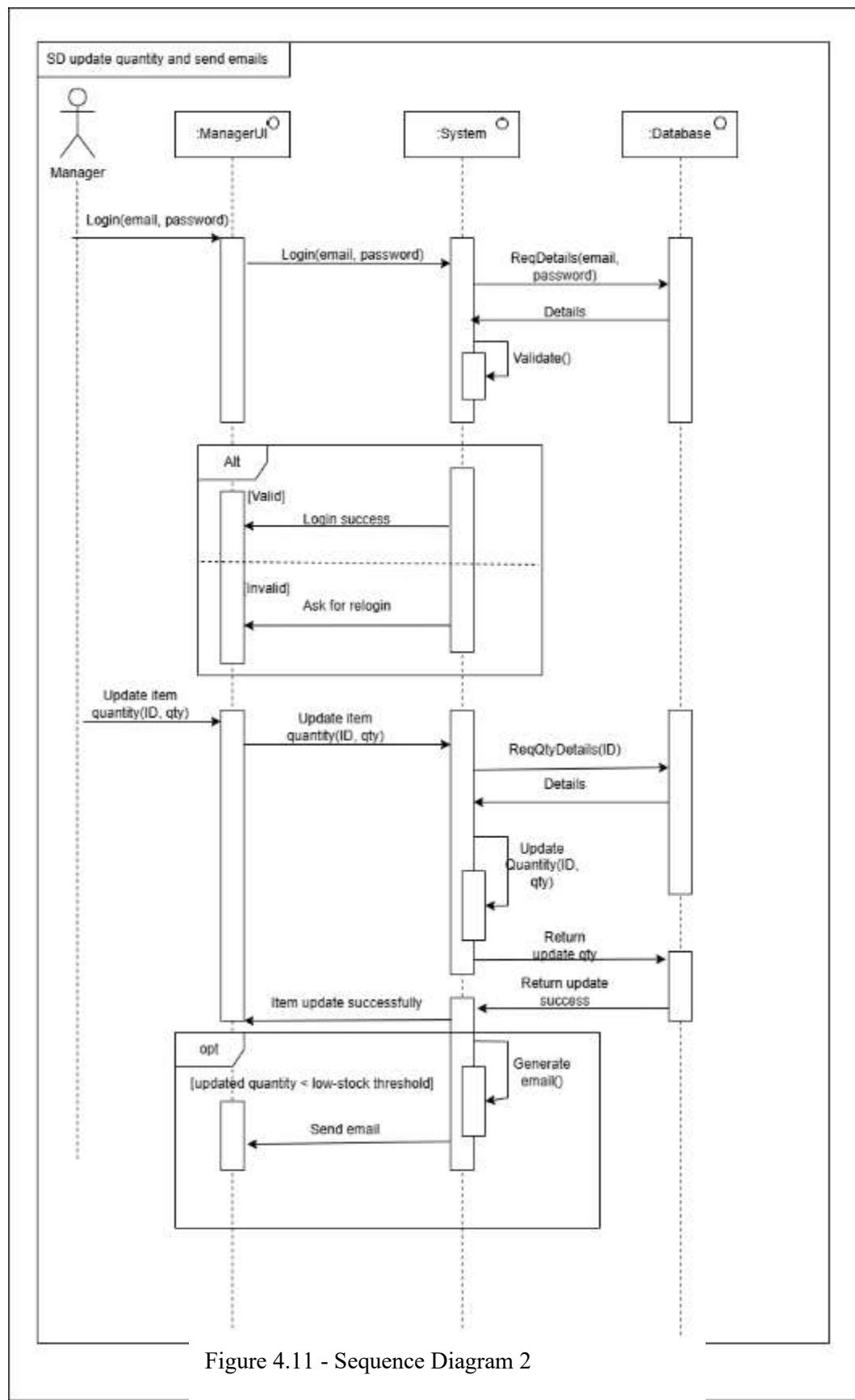


Figure 4.11 - Sequence Diagram 2

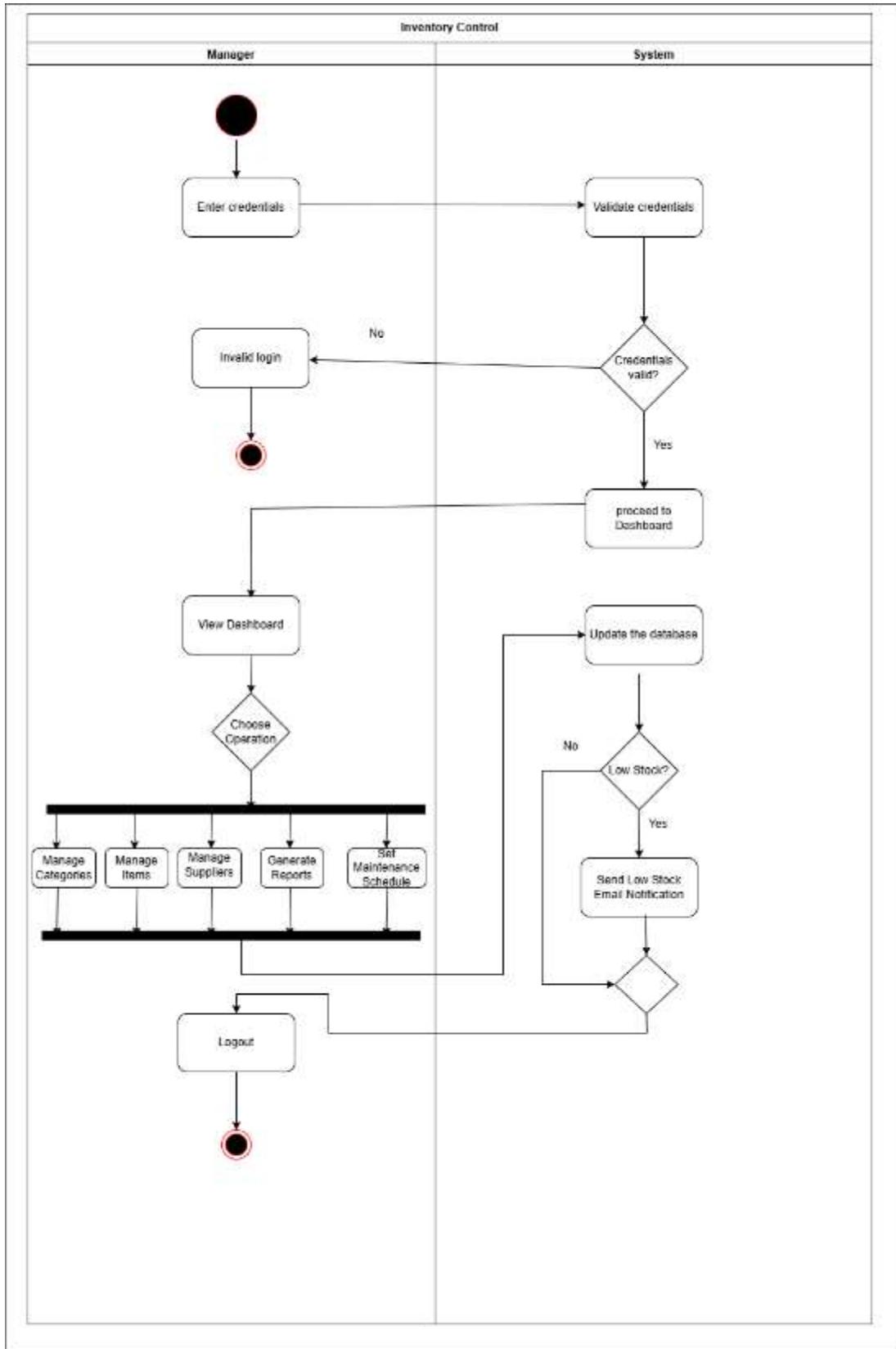


Figure 4.12 - Activity Diagram 2

4.7.3- K B P Kavisika - IT 23 5804 80

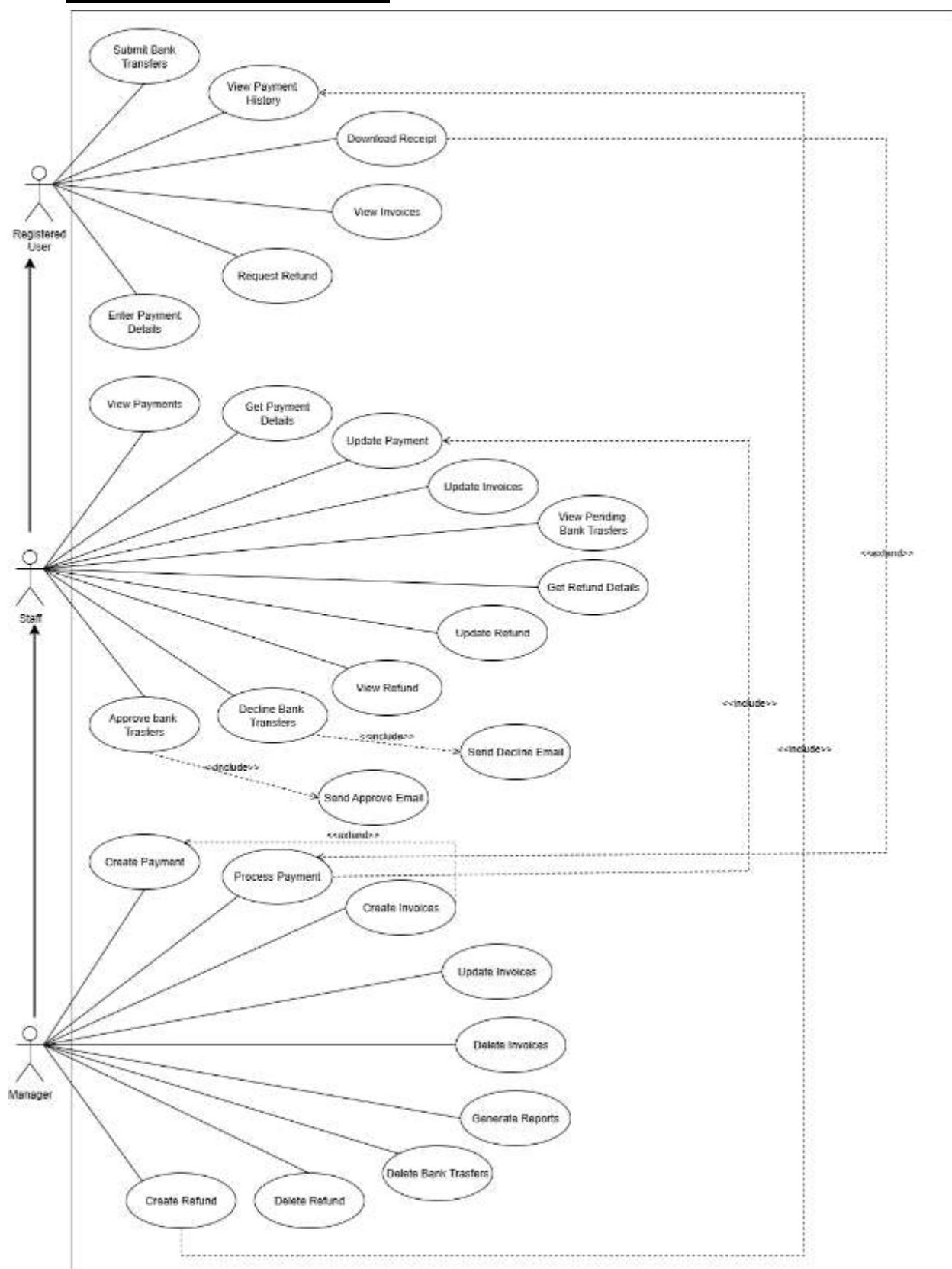


Figure 4.13 - Use Case 3

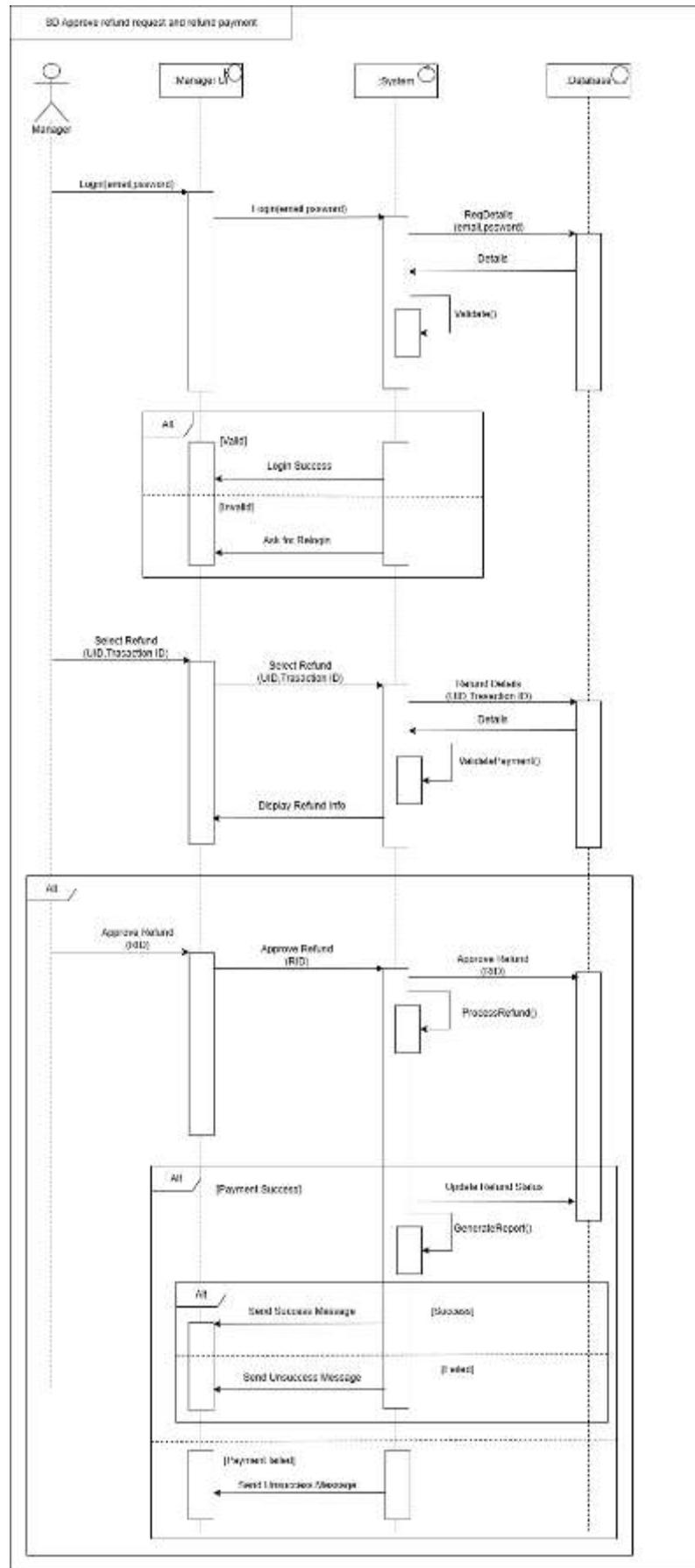


Figure 4.14 - Sequence Diagram 3

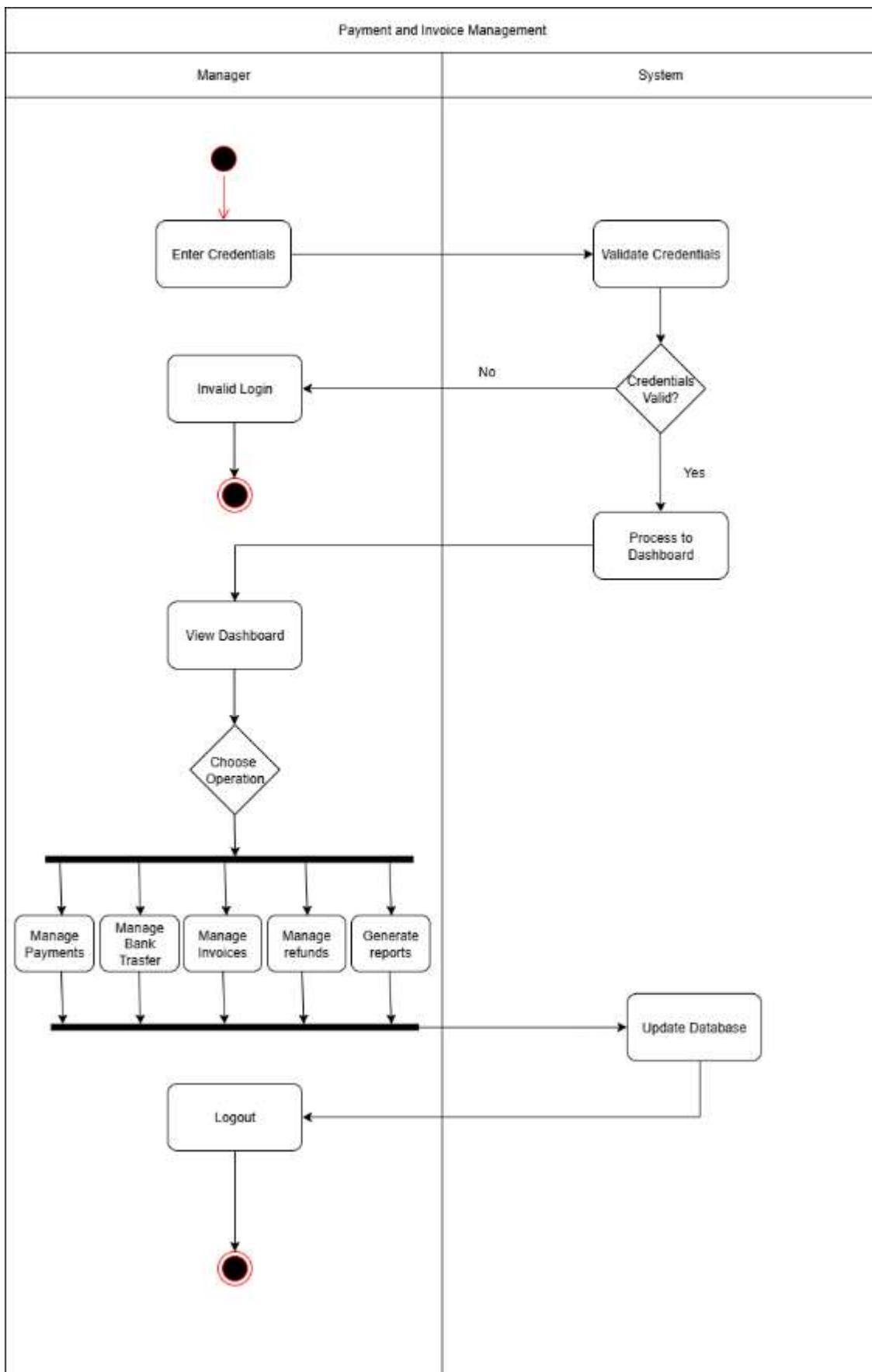


Figure 4.15 - Activity Diagram 3

4.7.4- O A I De Silva - IT 23 7536 62



Figure 4.16 - Use Case 4

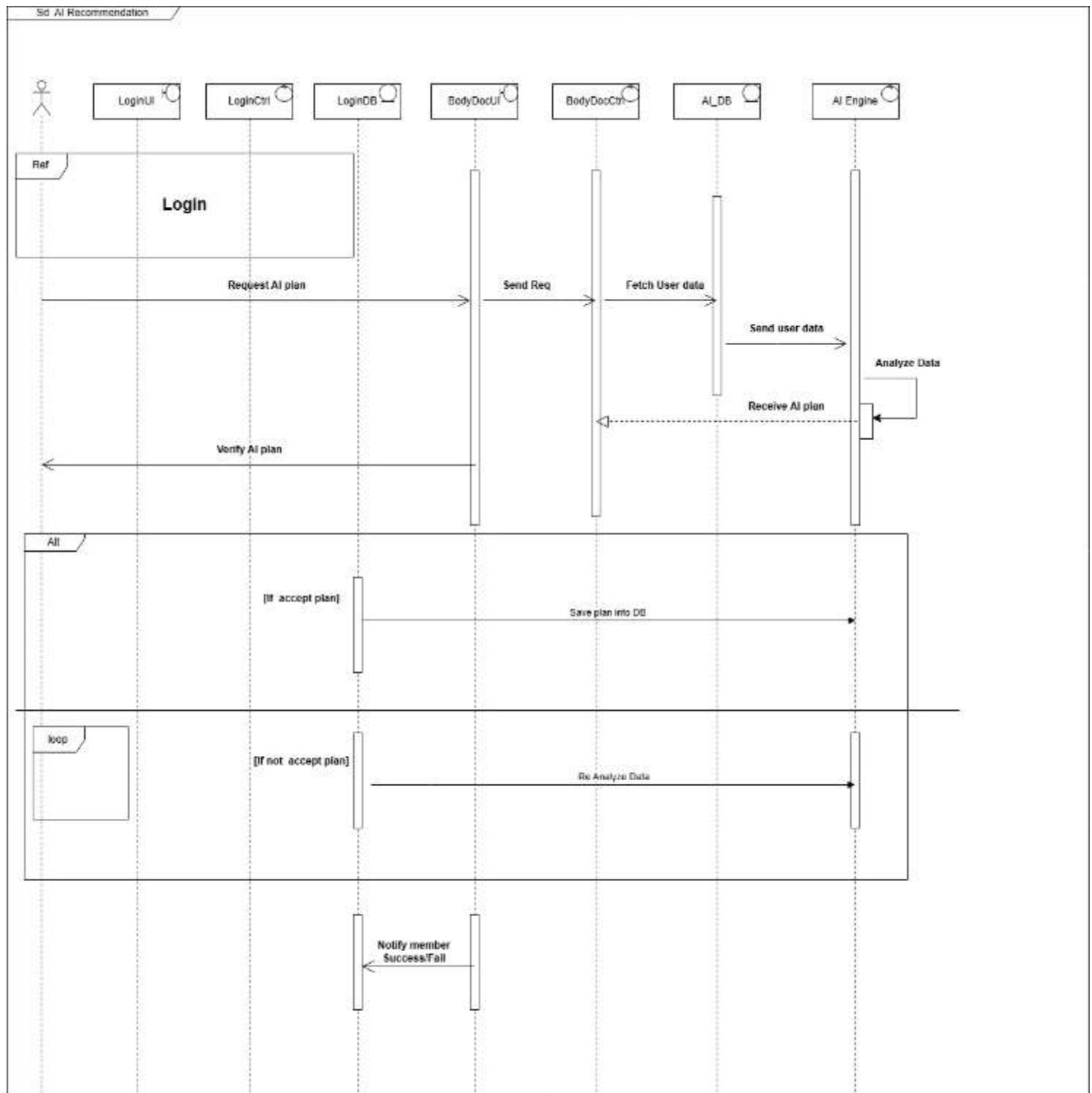


Figure 4.17 - Sequence Diagram 4

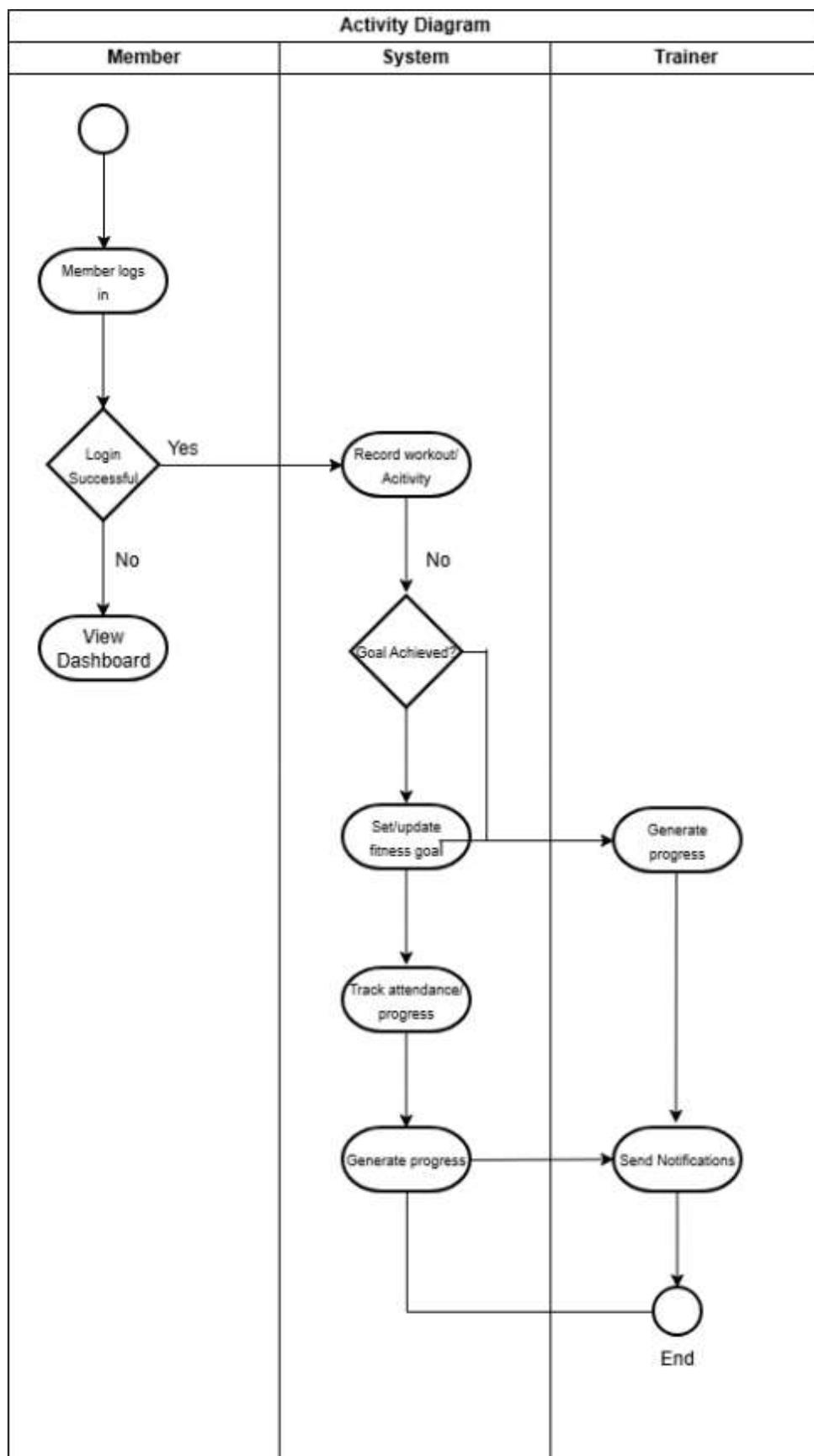


Figure 4.18 - Activity Diagram 4

4.7.5 - B Dhayabari - IT 23 7414 78

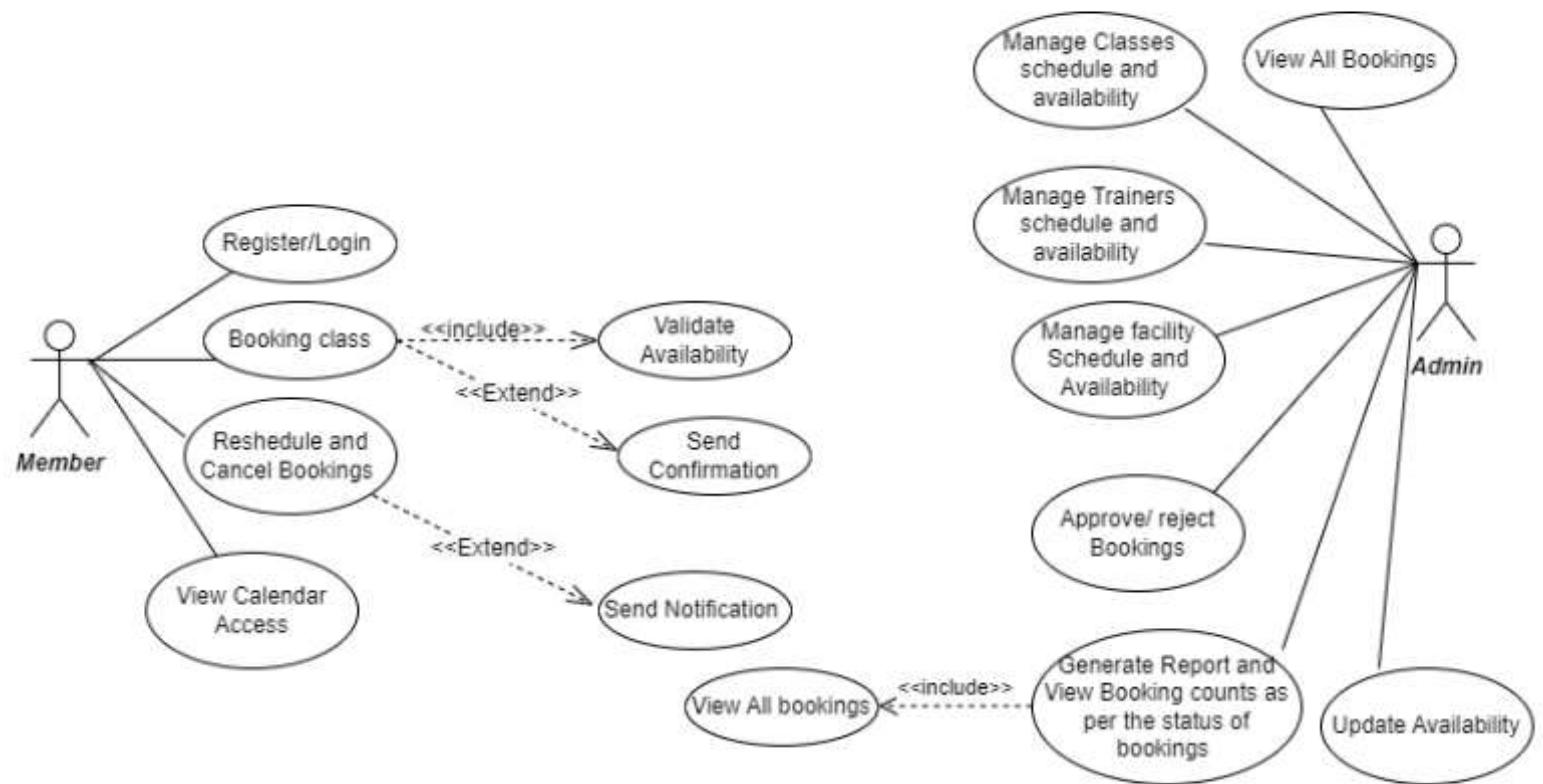


Figure 4.19 - Use Case 5

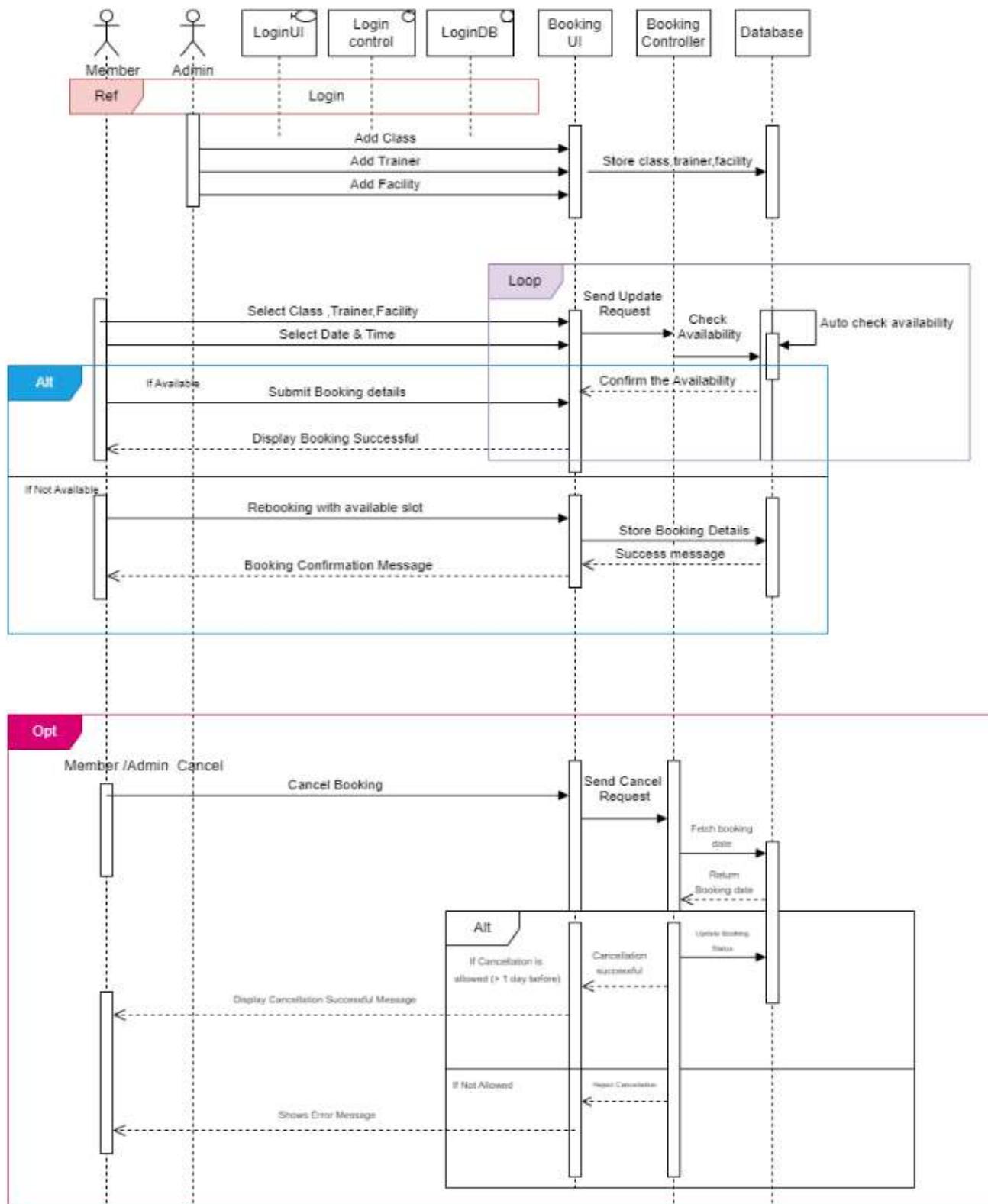


Figure 4.20 - Sequence Diagram 5

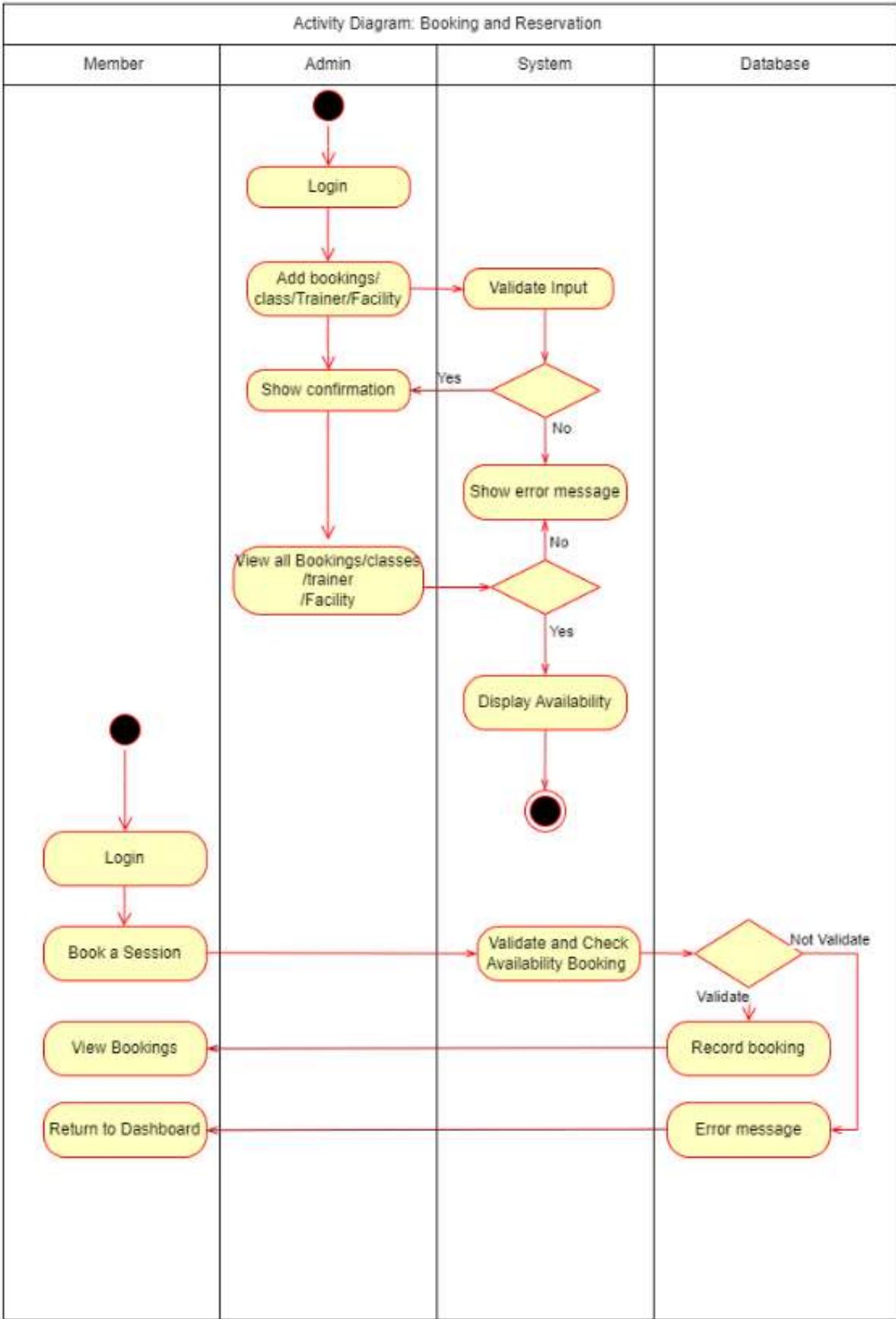


Figure 4.21 - Activity Diagram 5

Chapter 5. Testing

5.1 Karunathilaka H R G N | IT 23 5621 10

Test Case ID: UM001
Testing Function: User registration and profile management
Test Priority: High
Module Name: User Management
Dependencies: Valid email service, database connection, frontend registration form
<p>Test Steps:</p> <ol style="list-style-type: none"> 1. Fill in all required fields (name, email, phone, password, role) 2. Accept GDPR and marketing consent 3. Submit registration form 4. Verify email verification process 5. Login with new credentials 6. Update user profile information 7. Verify profile changes are saved

Test ID	Test Input	Expected Output	Actual Output	Result(pass/fail)
UM01	Name: "John Doe", Email: " john@example.com ", Phone: "+94712345678", Password: "ValidPass123!", Role: "member", GDPR: true	User account created successfully, verification email sent	User created, email sent	Pass
UM02	Name: "", Email: "invalid-email", Phone: "123", Password: "weak"	Validation errors for all invalid fields	Form shows specific validation errors	Pass
UM03	Valid registration data, but duplicate email	Error: "Email already exists"	Duplicate email error displayed	Pass
UM04	Valid user login, then update profile with new address	Profile updated successfully	Profile changes saved	Pass

Table 4 - Table of Test Case 1

Test Case 2: Membership Management - Membership Creation and Status Updates

Test Case ID: MM001
Testing Function: Membership lifecycle management
Test Priority: High
Module Name: Membership Management
Dependencies: Valid user account, membership plan exists, admin/staff login
<p>Test Steps:</p> <ol style="list-style-type: none"> 1. Navigate to membership management section 2. Create new membership for existing user 3. Select membership plan and set dates 4. Submit membership creation 5. Update membership status (pause/resume) 6. Cancel membership with reason 7. Verify membership history and status changes

Test ID	Test Input	Expected Output	Actual Output	Result(pass/fail)
MM01	User ID: valid_user_id, Plan ID: premium_plan, Start Date: today, Auto-renew: true	Membership created successfully with active status	Membership created, status: active	Pass
MM02	Valid membership ID, Status: "paused", Reason: "Medical leave"	Membership paused successfully	Status updated to paused	Pass
MM03	Valid membership ID, Status: "cancelled", Reason: "Moving abroad"	Membership cancelled with reason recorded	Status: cancelled, reason saved	Pass
MM04	Invalid user ID, Valid plan ID	Error: "User not found"	User validation error	Pass

Table 5 - Table of Test Case 2

Test Case 3: Membership Plans Management - Plan Creation and Modification

Test Case ID: MP001
Testing Function: Membership plan CRUD operations
Test Priority: Medium
Module Name: Membership Plans Management
Dependencies: Admin/manager login, database access
<p>Test Steps:</p> <ol style="list-style-type: none"> 1. Navigate to membership plans section 2. Create new membership plan with all details 3. Set pricing, duration, and category 4. Save the membership plan 5. Edit existing plan details 6. Update price and duration 7. Verify plan is active and available for selection

Test ID	Test Input	Expected Output	Actual Output	Result(pass/fail)
MP01	Name: "Premium Plan", Price: 5000, Duration: 30, Category: "weights", Currency: "LKR"	Membership plan created successfully	Plan saved with all details	Pass
MP02	Name: "Premium Plan" (duplicate), Price: 3000, Duration: 15, Category: "yoga"	Error: "Plan name already exists"	Duplicate name validation error	Pass
MP03	Valid plan ID, Update price: 6000, Duration: 45	Plan updated successfully	Price and duration changed	Pass
MP04	Name: "", Price: -100, Duration: 0	Validation errors for all invalid fields	Form shows validation errors	Pass

Table 6 - Table of Test Case 3

Test Case 4: Attendance Management - QR Code Check-in/Check-out System

Test Case ID: AM001
Testing Function: QR code based attendance tracking with membership validation
Test Priority: Medium
Module Name: Attendance Management
Dependencies: Valid user with active membership, camera access, QR scanner functionality
<p>Test Steps:</p> <ol style="list-style-type: none"> 1. Navigate to attendance scanning page 2. Grant camera permissions 3. Scan valid member's QR code 4. Verify membership validation 5. Process check-in/check-out 6. Verify attendance record creation 7. Test with expired membership

Test ID	Test Input	Expected Output	Actual Output	Result(pass/fail)
AM01	Valid QR code from user with active membership	Check-in successful, attendance record created	User checked in, record saved	Pass
AM02	Valid QR code from user with expired membership	Error: "No active membership found. Access denied."	Membership validation error	Pass
AM03	Invalid QR code format	Error: "Invalid QR token"	QR validation error	Pass
AM04	Same user scans QR twice (check-in then check-out)	First scan: check-in, Second scan: check-out	Status toggled correctly	Pass

Table 7 - Table of Test Case 4

5.2 Nawaratne N M B M L B | IT 23 5418 70

Test Case ID:	TC-INV-001
Testing Function:	CRUD operations for inventory items & categories
Test Priority:	High
Module Name:	Inventory management – Gym equipment & supplements
Dependencies:	Database(InventoryItem, Category, Supplier tables)

Test steps:

1. Open Add Item form, enter valid details → verify record inserted into DB.
2. Add Item with missing required field (Name empty) → verify validation error shown.
3. Update an existing Item's stock count → verify DB record updated.
4. Delete Item by ID → verify record removed.
5. Add Category → verify entry saved with status = "Active".

Test Case ID	Input	Expected Output	Actual Output	Status
TC-INV-001	Add item (valid details: Protein Powder, Category: Supplements, Type: Sellable)	Item record inserted into DB with status = "Active"	Item record created successfully and visible in inventory list	Pass
TC-INV-002	Add Item (missing Name)	Validation error "Item Name is required"	System displayed validation error message	Pass
TC-INV-003	Update Item Stock Quantity	Item record updated in DB	Quantity updated in database and reflected in inventory	Pass
TC-INV-004	Delete Item by ID	Item removed from DB	Item still visible in inventory due to soft delete bug	Fail

Table 8 - Table of Test Case 5

Test Case ID:	TC-INV-005
Testing Function:	CRUD operations for Equipment management
Test Priority:	High
Module Name:	Inventory management – Equipment Tracking
Dependencies:	Database(InventoryItem, Supplier tables)

Test steps:

1. Open Add Equipment form, enter valid details → verify equipment record created.
2. Add Equipment with missing Purchase Date → verify validation error.
3. Update Equipment maintenance date → verify DB record updated.
4. Search items by category → verify filtered results displayed.
5. Generate Equipment report → verify report contains correct data.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-INV-005	Add Equipment (Treadmill, Type: equipment, Purchase Date: 2025-01-15)	Equipment record inserted with maintenance tracking enabled	Equipment successfully added with maintenance schedule	Pass
TC-INV-006	Add Equipment (missing Purchase Date)	Validation error "Purchase Date is required for equipment"	System displayed validation error message	Pass
TC-INV-007	Update Equipment Maintenance Date	Equipment maintenance date record updated in DB	Maintenance date update	Pass

TC-INV-008	Search Items by Category name Equipment	Filtered list showing only gym equipment	Search returned correct filtered results	pass
------------	---	--	--	------

Table 9 - Table of Test Case 6

Test Case ID:	TC-INV-009
Testing Function:	Stock Management & Low Stock Alerts
Test Priority:	High
Module Name:	Inventory management – Stock control
Dependencies:	Database (InventoryItem, InventoryTransaction tables), Email Service

Test steps:

1. Update sellable item stock below minimum threshold → verify low stock alert triggered.
2. Process sale transaction → verify stock automatically decremented.
3. Add stock via purchase transaction → verify stock incremented.
4. Generate stock report → verify accurate stock levels displayed.
5. Test email notification to managers → verify emails sent successfully.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-INV-009	Update Protein Powder stock to 5 (below minimum 10)	Low stock alert email sent to all managers	Email notifications sent successfully to 4 managers	Pass
TC-INV-010	Process sale of 3 Protein Powder units	Stock decremented by 3, transaction recorded	Stock updated correctly, transaction logged	Pass
TC-INV-011	Add purchase of 50 Protein Powder units	Stock incremented by 50, purchase transaction recorded	Stock updated successfully	Pass

TC-INV-012	Generate Stock Report	Report showing all the stock levels in the inventory right now	Report generated with accurate low stock items	pass
------------	-----------------------	--	--	------

Table 10 - Table of Test Case 7

Test Case ID:	TC-INV- 013
Testing Function:	Shopping Cart & Member Orders
Test Priority:	Medium
Module Name:	Inventory Management - Member Shopping
Dependencies:	Database (Cart, InventoryItem, Member tables)

Test steps:

1. Member adds item to cart → verify cart record created/updated.
2. Update cart item quantity → verify cart updated in DB.
3. Remove item from cart → verify item removed from cart.
4. Process cart checkout → verify inventory stock decremented.
5. View cart contents → verify accurate item list and totals.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-INV-013	Member adds Whey Protein (Qty: 2) to cart	Cart record created/updated with item and quantity	Cart successfully updated for member	Pass
TC-INV-014	Update cart quantity from 2 to 5	Cart quantity updated to 5 in database	Quantity updated successfully	Pass

TC-INV-015	Remove item from cart	Item removed from cart record	Item successfully removed from cart	Pass
TC-INV-016	Process cart checkout	Inventory stock decremented, order created	Stock updated, order processed successfully	pass

Table 11 - Table of Test Case 8

Test Case ID:	TC-INV- 017
Testing Function:	Supplier Management & Purchase Orders
Test Priority:	Medium
Module Name:	Inventory Management - Supplier Operations
Dependencies:	Database (Supplier, InventoryItem tables)

Test steps:

1. Add new supplier with valid details → verify supplier record created.
2. Add supplier with duplicate email → verify validation error.
3. Update supplier contact information → verify DB record updated.
4. View items by supplier → verify filtered supplier inventory.
5. Generate supplier report → verify accurate supplier data.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-INV-017	Add Supplier (NutriSupply, email: info@nutrisupply.com)	Supplier record inserted into database	Supplier successfully added and available for selection	Pass
TC-INV-018	Add Supplier with duplicate email	Validation error "Email already exists"	System displayed duplicate email error	Pass

TC-INV-019	Update Supplier phone number	Supplier contact updated in DB	Phone number updated successfully	Pass
TC-INV-020	Filter items by supplier "NutriSupply"	Display all items from selected supplier	Correct supplier items displayed	pass

Table 12 - Table of Test Case 9

5.3 K B P Kavisika | IT 23 5804 80

Test Case ID: P001
Testing Function: Processing payments through PayHere gateway
Test Priority: High
Module Name: Payment Management
Dependencies: User must be logged in, PayHere gateway integration
<p>Test Steps:</p> <ol style="list-style-type: none"> 1. Navigate to the membership/booking payment page select any. 2. Select payment method and enter card details. 3. Submit payment request to PayHere 4. Verify payment status update 5. Transaction receipt will be sent to the given email.

Test ID	Test Input	Expected Output	Actual Output	Result(pass/fail)
P01	Amount:500 Type: Membership Method: Card	Payment record created with status 'completed'	Payment created successfully	Pass
P02	Amount: -100 Type: Membership Method: Card	Error: Amount must be positive	System displays validation error	Fail
P03	Invalid user(unregistered) Amount:5000 Type: Membership Method: Card	Error: Invalid user Id	Payment failed	Fail

Test Case ID: R001
Testing Function: Processing refund requests
Test Priority: Medium
Module Name: Refund Management
Dependencies: Completed payment record, refund policy
<p>Test Steps:</p> <ol style="list-style-type: none"> 1. Enter refund amount and reason 2. Process refund through PayHere 3. Update payment and create refund record

Test ID	Test Input	Expected Output	Actual Output	Result(pass/fail)
R01	Payment.Amount:5000 Refund.Amount:5000 Reason: User cancellation	Full refund processed, status='completed'	Refund completed successfully	Pass
R02	Payment.Amount:5000 Refund.Amount:2500 Reason: Partial refund	Partial refund processed, refund Amount=2500	Partial refund completed	Pass
R03	Payment.Amount:5000 Refund.Amount:6000 Reason: Over refund	Error: Refund amount exceeds payments amount	System declined over refund	Fail
R04	Already refunded payment Refund.Amount:6000	Error: Payment already refunded	Error displayed correctly	Fail

Table 13 - Table of Test Case 10

Test Case ID: B001
Testing Function: Bank transfer slip upload and verification
Test Priority: Medium
Module Name: Bank Transfer Management
Dependencies: User account, bank transfer record

Test Steps:
<ol style="list-style-type: none"> 1. Manager reviews the transfer details 2. Manager approves or declines the transfer 3. System updates payment status

Test ID	Test Input	Expected Output	Actual Output	Result(pass/fail)
B01	Valid receipt image Amount= 5000 Valid bank details	Transfer marked as 'pending', manager notification sent	Refund completed successfully	Pass
B02	Invalid receipt image Amount= 5000 Valid bank details	Error: Please upload valid image file	Partial refund completed	Fail
B03	Valid receipt image Amount mismatch Uploaded= 5000 Expected = 6000 Valid bank details	Error: Managers declined the payment	Payment gets declined, Status 'failed'	Fail
B04	Valid receipt image Amount= 5000 Valid bank details Managers approve	Error: Payment status updated 'completed'	Payment gets approve, Status 'completed'	Pass
B05	Valid receipt image Amount= 5000 Valid bank details Manager declined	Error: Payment status updated 'failed'	Payment gets declined, Status 'failed'	Fail

Table 14 - Table of Test Case 11

Test Case ID: I001
Testing Function: Automatic invoice generation after payment completion
Test Priority: High
Module Name: Invoice Management
Dependencies: Completed payment record
Test Steps:

- | |
|---|
| <ol style="list-style-type: none"> 1. Verify automatic invoice creation 2. Check invoice details accuracy 3. Verify email notification |
|---|

Test ID	Test Input	Expected Output	Actual Output	Result(pass/fail)
I01	Completed payment Amount: 5000 Type: membership User Id: valid	Invoice created with matching amount and user details	Invoice generated correctly	Pass
I02	Completed payment Amount: 2500 Type: booking User Id: valid	Invoice created with matching amount and user details	Invoice shows booking details	Pass
I03	Failed payment Type: membership User Id: valid	No invoice generated	No invoice created for failed payment	Fail
I04	Completed payment Missing user data	Invoice creation fails gracefully	No invoice created for missing user data payment	Fail

Table 15 - Table of Test Case 12

5.4 B Dhayabari | IT 23 7414 78

Test Case ID:	TC-BKG-001
Testing Function:	CRUD operations for Bookings
Test Priority:	High
Module Name:	Booking management
Dependencies:	Database (Booking, Class, Trainer, Facility tables)

Test steps:

- 1.Add Booking → verify record inserted.
- 2.Add Booking with 0 seats → verify validation error.
- 3.Add Booking with past date → verify validation error.
- 4.Update Booking date/time/trainer/facility → verify record updated.
- 5.Cancel Booking → verify booking removed or marked cancelled.
- 6.Search/Filter Booking → verify filtered results displayed.
- 7.Generate Booking report → verify report contains correct data.

Table 16 - Table of Test Case 13

Test Case ID	Input	Expected Output	Actual Output	Status
TC-BKG-001	Add booking (Yoga, John Doe, 05-10-2025, 5.00pm, Fee 500)	Booking record inserted into DB with status = "Active"	Booking created successfully	Pass
TC-BKG-002	Add booking (0 seats)	Validation error: "No seats available"	System displayed validation error	Pass
TC-BKG-003	Add booking (past date)	Validation error: "Invalid booking date"	System displayed validation error	Pass
TC-BKG-004	Update booking date/time/facility	Booking record updated in DB	Booking updated successfully and reflected in calendar	Pass
TC-BKG-005	Cancel booking	Booking removed or marked cancelled	Booking cancelled successfully	Pass
TC-BKG-006	Filter bookings by trainer/class/facility	Filtered list displayed	Correct filtered bookings displayed	Pass

TC-BKG-007	Generate daily/weekly/monthly report	Report shows correct booking data	Report generated correctly	Pass
------------	--------------------------------------	-----------------------------------	----------------------------	------

Test Case ID:	TC-CLS-001
Testing Function:	CRUD operations for Classes
Test Priority:	High
Module Name:	Class management
Dependencies:	Database (Class table)

Test Steps:

8. Add Class → verify record inserted.
9. Add Class with missing required field → verify validation error.
10. Update Class details → verify record updated in DB.
11. Delete Class → verify record removed.
12. Search/Filter Class → verify filtered results displayed.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-CLS-001	Add class (Yoga, 60 min, Fee 5000)	Class record inserted into DB with status = "Active"	Class record created successfully and visible in list	Fail
TC-CLS-002	Add class with empty Name	Validation error: "Class Name is required"	System displayed validation error	Pass
TC-CLS-003	Update class duration/fee	Class record updated in DB	Updated successfully and reflected in list	Pass
TC-CLS-004	Delete class	Class removed from DB	Class removed from list	Pass

TC-CLS-005	Search class by Name	Filtered list displayed	Correct filtered classes displayed	Fail
------------	----------------------	-------------------------	------------------------------------	------

Table 17 - Table of Test Case 14

Test Case ID:	TC-TRN-001
Testing Function:	CRUD operations for Trainers
Test Priority:	High
Module Name:	Trainer Management
Dependencies:	Database (Trainer table)

Test Steps:

1. Add Trainer → verify record inserted.
2. Add Trainer with missing required field → verify validation error.
3. Update Trainer specialization/status → verify record updated.
4. Delete Trainer → verify record removed.
5. Search/Filter Trainer → verify filtered results displayed.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-TRN-001	Add Trainer (John Doe, Yoga,1, Active)	Trainer record inserted in DB with status = "Active"	Trainer created successfully	Pass
TC-TRN-002	Add Trainer without Name	Validation error: "Trainer Name is required"	System displayed validation error	Pass
TC-TRN-003	Update Trainer specialization/status	Trainer record updated in DB	Updated successfully and reflected in list	Fail
TC-TRN-004	Delete Trainer	Trainer removed from DB	Trainer removed from list	Pass

TC-TRN-005	Search Trainer by specialization	Filtered list displayed	Correct filtered trainers displayed	Fail
------------	----------------------------------	-------------------------	-------------------------------------	------

Table 18 - Table of Test Case 15

Test Case ID:	TC-FCL-001
Testing Function:	CRUD operations for Facilities
Test Priority:	High
Module Name:	Facility Management
Dependencies:	Database (Facility table)

Test Steps:

- 1.Add Facility → verify record inserted.
- 2.Add Facility with missing required field → verify validation error.
- 3.Update Facility details → verify record updated.
- 4.Delete Facility → verify record removed.
- 5.Search/Filter Facility → verify filtered results displayed.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-FCL-001	Add Facility (Room A, Equipment: Mats, Status Active)	Facility record inserted in DB with status = "Active"	Facility created successfully	Pass
TC-FCL-002	Add Facility without Name	Validation error: "Facility Name is required"	System displayed validation error	Pass
TC-FCL-003	Update Facility equipment/status	Facility record updated in DB	Updated successfully and reflected in list	Fail

TC-FCL-004	Delete Facility	Facility removed from DB	Facility removed from list	Pass
TC-FCL-005	Search Facility by equipment	Filtered list displayed	Correct filtered facilities displayed	Fail

Table 19 - Table of Test Case 16

5.5 O.A.I. De Silva | IT 23 7536 62

Test Case ID:	TC-TRK-001
Testing Function:	CRUD operations for progress
Test Priority:	High
Module Name:	Fitness Tracking – Nutrition Management
Dependencies:	Database (Nutrition, User tables)

Test steps:

1. Open Add Nutrition form, enter valid details → verify record inserted into DB.
2. Add Nutrition with missing required field (Name empty) → verify validation error shown.
3. Update nutrition calories value → verify DB record updated.
4. Delete Nutrition by ID → verify record removed.
5. Search Nutrition by food type → verify accurate results displayed.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-TRK-001	Add Nutrition (valid details: Chicken Breast, Protein: 31g, Carbs: 0g, Fat: 3.6g, Calories: 165)	Nutrition record inserted with status = “Active”	Record successfully created and visible in nutrition list	pass

TC-TRK-002	Add Nutrition (missing name)	Validation error “Food Name is required”	Validation message displayed	Pass
TC-TRK-003	Update Nutrition Calories	Record updated in DB	Calories updated successfully	Pass
TC-TRK-004	Delete Nutrition by ID	Record removed from DB	Record still visible due to soft delete issue	Pass
TC-TRK-005	Search by type “Protein Source”	Filtered nutrition list displayed	Filter returned accurate list	Fail

Table 20 - Table of Test Case 17

Test Case ID:	TC-TRK-006
Testing Function:	CRUD operations for Workout Management
Test Priority:	High
Module Name:	Fitness Tracking – Workout Management
Dependencies:	Database (Workout, User tables)

Test steps:

1. Open Add Workout form, enter valid details → verify workout record created.
2. Add Workout with missing duration → verify validation error.
3. Update workout intensity → verify record updated in DB.
4. Delete workout by ID → verify record removed.
5. Generate Workout summary report → verify correct data displayed.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-TRK-006	Add Workout (Push Ups, Duration: 30 mins, Intensity: Medium, CaloriesBurned: 200)	Workout record inserted	Workout successfully added	Pass

TC-TRK-007	Add Workout (missing Duration)	Validation error “Duration is required”	Validation message displayed	Pass
TC-TRK-008	Update Workout Intensity	Intensity updated successfully	Value updated in DB	Pass
TC-TRK-009	Delete Workout by ID	Record deleted from DB	Deleted successfully	Pass
TC-TRK-010	Generate Workout Report	Report generated with all recent workouts	Report generated with all recent workouts	Pass

Table 21 - Table of Test Case 18

Test Case ID:	TC- TRK-011
Testing Function:	Fitness Tracking – Goal Management
Test Priority:	High
Module Name:	Fitness Tracking – Goal Management
Dependencies:	Database (Goal, Progress tables)

Test steps:

6. Open Add Goal form, enter valid goal → verify record inserted.
7. Add Goal with missing target value → verify validation error.
8. Update goal status → verify DB record updated.
9. Delete goal by ID → verify record removed.
10. Search goal by type → verify correct results.

Test Case ID	Input	Expected Output	Actual Output	Status

TC-TRK-011	Add Goal (Type: Weight Loss, Target: 60kg, StartDate: 2025-02-01, EndDate: 2025-05-01)	Goal record created successfully	Record created and visible in goal list	Pass
TC-TRK-012	Add Goal (missing TargetValue)	Validation error “Target value is required”	Validation shown properly	Pass
TC-TRK-013	Update Goal Status → Completed	Status updated successfully in DB	Reflected correctly in progress dashboard	Pass
TC-TRK-014	Delete Goal by ID	Record removed	Record deleted successfully	Pass
TC-TRK-015	Search Goal by Type “Muscle Gain”	Filtered goal list displayed	Correct filtered data shown	Pass

Table 22 - Table of Test Case 19

Test Case ID:	TC-TRK-016
Testing Function:	CRUD operations for Progress Tracking
Test Priority:	High
Module Name:	Fitness Tracking – Progress & Achievement Tracking
Dependencies:	Database (Progress, Goal, Workout, Nutrition tables)

Test steps:

6. Add progress entry with linked goal and workout → verify record inserted.
7. Add Progress with missing date → verify validation error.
8. Update achievement rate → verify DB record updated.
9. View Progress summary → verify combined data from all linked entities.
10. Generate progress report → verify data accuracy.

Test Case ID	Input	Expected Output	Actual Output	Status
TC-TRK-016	Add Progress (Goal: Weight Loss, Date: 2025-02-05, AchievementRate: 80%)	Progress record linked to goal successfully	Progress saved successfully	Pass
TC-TRK-017	Add Progress (missing Date)	Validation error “Date is required”	Error displayed	Pass
TC-TRK-018	Update Achievement Rate	Record updated in DB	Updated correctly	Pass
TC-TRK-019	View Progress Summary	Dashboard shows combined goal, nutrition & workout info	Displayed correctly	Pass
TC-TRK-020	Generate Progress Report	Accurate report generated	Report generated correctly	Pass

Table 23 - Table of Test Case 20

Chapter 6. Evaluation and Conclusion

6.1 Evaluation

The proposed ReadSpace Library Management System comprehensively addresses the essential functionalities required to deliver a modern, efficient, and integrated library experience. It combines user management, book management, and inventory control, financial tracking, and reservations for tables and PCs, ensuring both operational excellence and enhanced member satisfaction. Each module plays a significant role in achieving smooth functionality and laying the groundwork for scalability.

User management lies at the core of ReadSpace, enabling streamlined registration, profile maintenance, and membership tracking. This ensures smooth onboarding of members, monitoring of borrowing history, and tailored access to services. By distinguishing between casual visitors and registered members, the system enhances personalization, engagement, and long-term user retention.

Book management provides robust tools for cataloging, borrowing, returning, and reserving books. Automated processes minimize delays, reduce human error, and improve accessibility for users. The ability to check availability in real time and manage holds or overdue penalties strengthens efficiency and builds user trust.

Inventory management extends beyond books and other resources. By tracking stock levels and replenishment cycles, the system prevents shortages, eliminates manual errors, and ensures that resources are accessible when required. This also supports strategic planning for resource expansion in line with user demand.

Finance management ensures transparency and accountability by tracking fees, fines, and revenue streams. Automated billing for overdue returns, membership renewals, and reservations reduces administrative workload and improves financial reporting. The centralized dashboard offers valuable insights for budgeting, forecasting, and strategic decision-making.

Reservations for tables and PCs add a user-focused feature to ReadSpace. Members can seamlessly book study areas or computer stations in advance, ensuring efficient space utilization and minimizing conflicts. Administrators benefit from automated scheduling, usage analytics, and cancellation management, which improve resource allocation and enhance satisfaction.

6.2 Conclusion

The main goal of the Library Management and Reservation System has been achieved, which is to convert traditional library operations into a completely automated, effective, and user-centered digital platform. The system satisfies all the established goals and provides a comprehensive solution to optimize library administration and improve user experience by integrating the five main functional areas of finance management, user management, table and PC reservations, inventory management, and book management.

The project successfully solves the issues that traditional library systems confront, including inefficient resource allocation, restricted accessibility, and manual record-keeping. The User Management module makes it easier to register and maintain member data, guaranteeing well-organized storage and convenient access. Users can find and access reading materials more easily thanks to the Book Management module, which also improves the efficiency of classifying, borrowing, and returning items.

By providing a practical digital method for staff and students to reserve study areas and computer resources, the Table and PC Reservation feature lessens conflict and encourages equitable resource use. Better decisions about stock management and procurement are supported by the Inventory Management module, which offers precise tracking and upkeep of library assets. In addition to offering analytical insights for planning and budgeting, the Finance Management module guarantees the transparent management of all financial activities, including membership fees, fines, and other payments.

The system fulfills its goals of increasing user satisfaction, reducing human workloads, and improving operational accuracy by integrating these elements. Additionally, the platform supports accessibility, data integrity, and transparency, all of which are essential components of contemporary digital library systems.

In order to sum up, the Library Management and Reservation System has achieved all project goals and achieved its mission of developing an intelligent, dependable, and long-lasting solution for contemporary library operations. It not only streamlines routine administrative duties but also offers a scalable framework for upcoming additions and integrations. In the end, the project shows how technology may be used effectively to enhance library services in terms of organization, learning, and creativity.

References

- [1] "Next.js," Next.js Documentation. Vercel Inc. , 2024. [Online]. Available: <https://nextjs.org/>.
- [2] I. Sommerville, Software Engineering (10 th Edition), Pearson Education, 2016.
- [3] K. Coomunity, "Koha - Open source ILS," Koha community, [Online]. Available: <https://koha-community.org/>. [Accessed October 2025].
- [4] M. A. a. K. Mahmood, "Effects of IT Competencies on performance of Library automation: A case of Koha system in Pakistan," *The Electronic Library (published by Emerald Publishing)*, Vols. Volume 30, No. 1, pp. 59-74, 2012.
- [5] L. Ltd, "Library Automation Solutions," Libsys Ltd, [Online]. Available: <https://www.libsys.co.in/>. [Accessed October 2025].
- [6] A. A. a. S. Kumar, "Modern Web Development Using MEAN Stack: MongoDB, Express, AngularJS, and NodeJS," *International Journal of Computer Application*, vol. 113, pp. 17-21, 2015.
- [7] M. Grinberg, Full-Stack React Projects: Modern web development using React 16, Node.js, Express, and MongoDB, Birmingham, UK: Packt Publishing, 2020.
- [8] Microsoft, "Visual Studio code - Code Editing, Redefined," Microsoft, 2015. [Online]. Available: <https://code.visualstudio.com>. [Accessed October 2025].
- [9] I. GitHub, "GitHub: Where the world builds software," GitHub, Inc, (Owned by Microsoft), 2008. [Online]. Available: <https://github.com>. [Accessed October 2025].
- [10] I. Postman, "Postman API Platform," Postman, Inc., 2012. [Online]. Available: <https://www.postman.com>. [Accessed October 2025].
- [11] Atlassain, "Jira Software - Project Management for Agile Teams," Atlassian pvt Ltd, 2002. [Online]. Available: <https://www.atlassian.com/software/jira>. [Accessed October 2025].

Appendix A – Work Done by each Member

Member ID and Name	Work Done
IT 23 5621 10 H R G N Karunathilaka	User Management, Membership Management and Attendance and Access Control Components
IT 23 5418 70 N M B M L B Nawaratne	Inventory Control and Store Components
IT 23 5804 80 K B P Kavisika	Payment Management and PayHere Integration Components
IT 23 7536 62 O A 1 De Silva	User Progress Tracking Components
IT 23 7414 78 B Dhayabari	Booking and Reservation Management components

Table 24 - Appendix A – Work Done by each Member

Appendix B – Contribution to Final Report

Member ID and Name	Contribution
IT 23 5621 10 H R G N Karunathilaka	<ul style="list-style-type: none"> • Background, problem and motivation and literature review • Chapter 4 - User Management, Membership Management and Attendance and Access Control design and development content and diagrams • Test cases with results
IT 23 5418 70 N M B M L B Nawaratne	<ul style="list-style-type: none"> • Chapter 4 - Inventory Control and Store design and development content and diagrams • Aim and objectives, solution overview and structure of the report • Test cases with results
IT 23 5804 80 K B P Kavisika	<ul style="list-style-type: none"> • Chapter 4 - Payment Management and PayHere Integration content and diagrams • Methodology and references • Test cases with results
IT 23 7536 62 O A l De Silva	<ul style="list-style-type: none"> • Chapter 4 - User Progress Tracking Components design and development content and diagrams • Requirements • Test cases with results
IT 23 7414 78 B Dhayabari	<ul style="list-style-type: none"> • Chapter 4 - Booking and Reservation Management design and development content and diagrams • Evaluation and conclusion • Test cases with results

sTable 25 - Appendix B – Contribution to Final Report