Year 2 Semester 2 2025

IT Project

GYM Management System

(ZFit)

Information Technology Project

IT2080

Group ID:

ITP25_B2_W225

Group Details:

Campus : Malabe

Group Number : ITP25_B2_W225

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

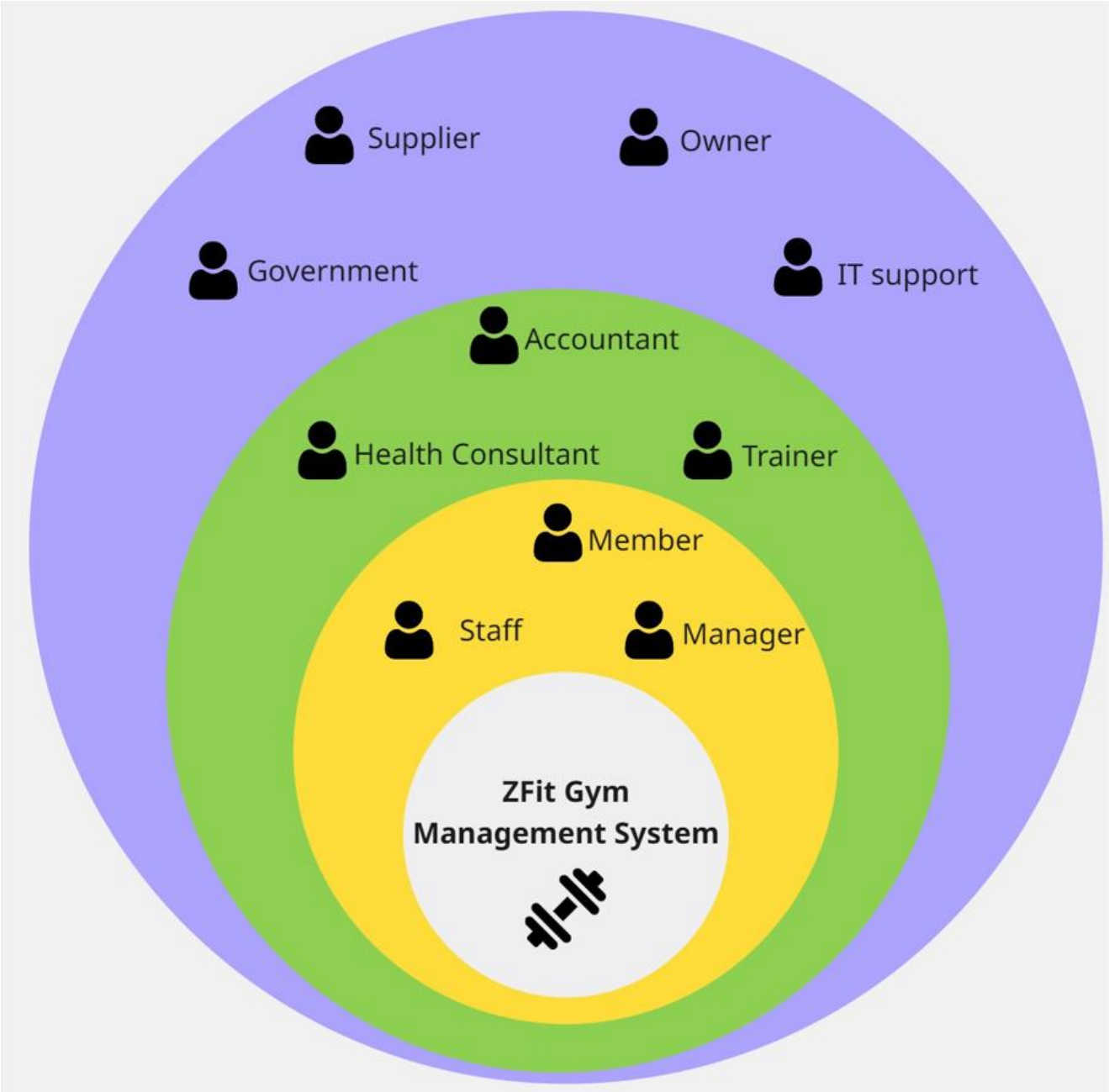
Activity 2

Stakeholders

- Member
- Staff
- Manager
- Trainer
- Accountant
- Health Consultant
- IT Support
- Owner
- Supplier
- Government

The Stakeholders of the ZFit gym Management System serve a wide range of purposes. Primary stakeholders include members, staff, managers and trainers who use the system for fitness and daily activities. Secondary stakeholders include gym owners, gym managers, health consultants, IT support, warehouse management, suppliers and government agencies. These users rely on the system for operational and regulatory management.

Onion Diagram



Functional Requirements

- Members should be able to register using their information.
- The members should be able to login using their credentials, view and manage their profile.
- The members should be able to browse membership options and subscribe to memberships.
- The members should be able to view their membership status and renew their membership.
- The system should create a QR code for each member, store it and display it to the member for attendance and access control.
- Members should log entering using their username and password (ID). After logging they will be able to view their profiles including details such as attendance history, personal information and assigned trainer information.
- Members / trainers can record their work out details. Like exercise names, sets, repetitions, used weight, any additional notes will be included in this. This information should be saved in the database their given ID. They can access it for future purposes.
- Members can set fitness goals such as “Lose 5kg in 3 months” or “Bench press 80kg.” Progress update can be added manually (e.g., “current weight: 56kg”) and this can save in the database for tracking.
- Members and trainers should be able to see attendance history, workout history and progress of each overtime. This information can be displayed in charts or tables for better understanding.
- Members can view available gym facilities, classes, and trainer schedules without logging in.
- Members can register and log in using their email and password.
- Logged in members can book gym sessions, classes, or equipment by selecting a date, time, and resource.
- Members can cancel or reschedule their existing bookings from their profile.

- The system displays confirmation messages after successful booking, cancellation, or rescheduling.
- Logged in members can view their full booking history, including past and upcoming bookings.
- Members and administrators can view payments history.
- Receipt must be generated after each payment.
- Payments can be filtered by date, service, status.
- Staff can also set goals for members.
- The staff and managers should be able to create member accounts, view and manage their profile.
- The managers and staff should be able to see if the members membership subscription has expired.
- The system allows managers and staff members to view inventory details.
- Manager users can view, edit, or cancel any booking in the system.
- Manager users can view basic booking reports such as total bookings, cancellations, and most booked facilities.
- Managers able to view, edit and manage invoices.
- The managers should be able to able to create membership offerings and customize them.
- Manager should be able to search members and generate member reports.
- The system allows managers to add new inventory items.
- The system allows managers to update inventory details.
- The system allows managers to remove inventory items.
- The system allows managers to generate inventory reports.
- The system generates low stock and maintenance alerts.
- An Manager should assign a trainer to a member. This link should be saved in the database so both the trainers and members will be able to see who they are working with.
- The system should generate sample reports like, Member attendance of the last month, Common workout activities, Summary of the Goal Progress. Reports that can be shown on dashboard or exported as any documentation type (PDF, Excel).

- The system should remind the members about any upcoming news, updates, missing sessions, or renewals. Mobile apps or web can used to appear the notifications or that can be sent as emails.
- Each invoice must include ; Unique invoice ID, Member details (name, ID), List of service/ products charged, Total amount (with tax, discounts), Due date.
- Support the following payments methods; Cash, Card payments, Bank Transfers.
- Should integrate with payment service providers such as PayHere.
- Support for single payments and online automatic (recuring) payments.
- The system checks existing bookings to prevent double booking of the same resource and time slot.

Non-Functional Requirement

Security

- JWT with expiring tokens to authentication system for secure login and session management.
- Each password should be hashed before storing it in the database.
- Role based access control to ensure only the right people with permission can access restricted services.

Performance

- Major functions of the website should response within 2 seconds.
- System should run without any issues under 1000 concurrent users.

Usability

- Web app should be responsive and be able to use on any screen size like tablets, laptops and mobile phones.
- Should work on any operating system like Windows, Linux and MacOS.
- Should be user friendly enough to use the platform for even people with less technical knowledge.

Maintainability

- Typescript for better type safety and maintainable code base.
- Descriptive comments throughout the code for better understanding of the code.
- Comprehensive documentation.

Analyzing Non-Functional Requirements based on each role

Member

- **Security –** Personal data should be secure.
- **Performance –** Quick payments and subscription to memberships through the portal.
- **Usability –** Easy booking and rescheduling.

Staff

- **Security –** Access to restricted functions using role based access system .
- **Performance –** Quick payments and subscription to memberships through the portal.

Manager

- **Security –** Reports and summaries should be only accessible for managers.
- **Performance –** Should be able to process the reports withing 5 seconds.
- **Usability –** Easy to review data.

Technical Requirement

Frontend

- NextJs and React are used for building dynamic and responsive user interfaces with server-side rendering support and SEO optimization. NextJs built in file based routing will be used in the frontend.
- Typescript which is a JavaScript variety with type safety will be used for reliable and maintainable code base.
- Tailwind CSS for styling the user interfaces.
- Redux Toolkit for advanced state management.

Backend

- NodeJS runtime for running JavaScript in the server.
- ExpressJS as the backend NodeJS framework to build reliable backend.
- REST APIs for communicating with frontend and third-party services.
- JWT authentication system for a stateless and secure role based access control authentication system.
- Bcrypt for hashing passwords.
- Typescript which is a JavaScript variety with type safety will be used for reliable and maintainable code base.

Database

- MongoDB as the No-SQL database for storing data.
- Mongoose ORM for schema validation and querying.

Use Case Diagram

Use Case Description

Use Case ID	01	
Name	Reservation and booking	
Summary	Member Book/reserves the Classes or Sessions	
Priority	01	
Pre-condition	Member must Log in to the System	
Post-condition	Booking is recorded	
Primary actor	Member	
Trigger	Member decides to reserve a gym session/Class	
Main scenario	**Step**	**Action**
	1.	Member Visit the Reservation page after the log in
	2.	Member selects the Session, class or trainer
	3.	System displays the Available time slots and asks to enter the details.
	4.	Member selects preferred time slot
	5.	Member giving the Booking details
	6.	System validates availability
	7.	System reserves reservation.
	8.	System sends confirmation notification
Extensions	**Step**	**Branching Action**
	1a	System notifies Member that the entered E-mail is not valid.
	4a	System notifies Member if the slot is already full.
	4b	System notifies Member if the selected trainer is unavailable

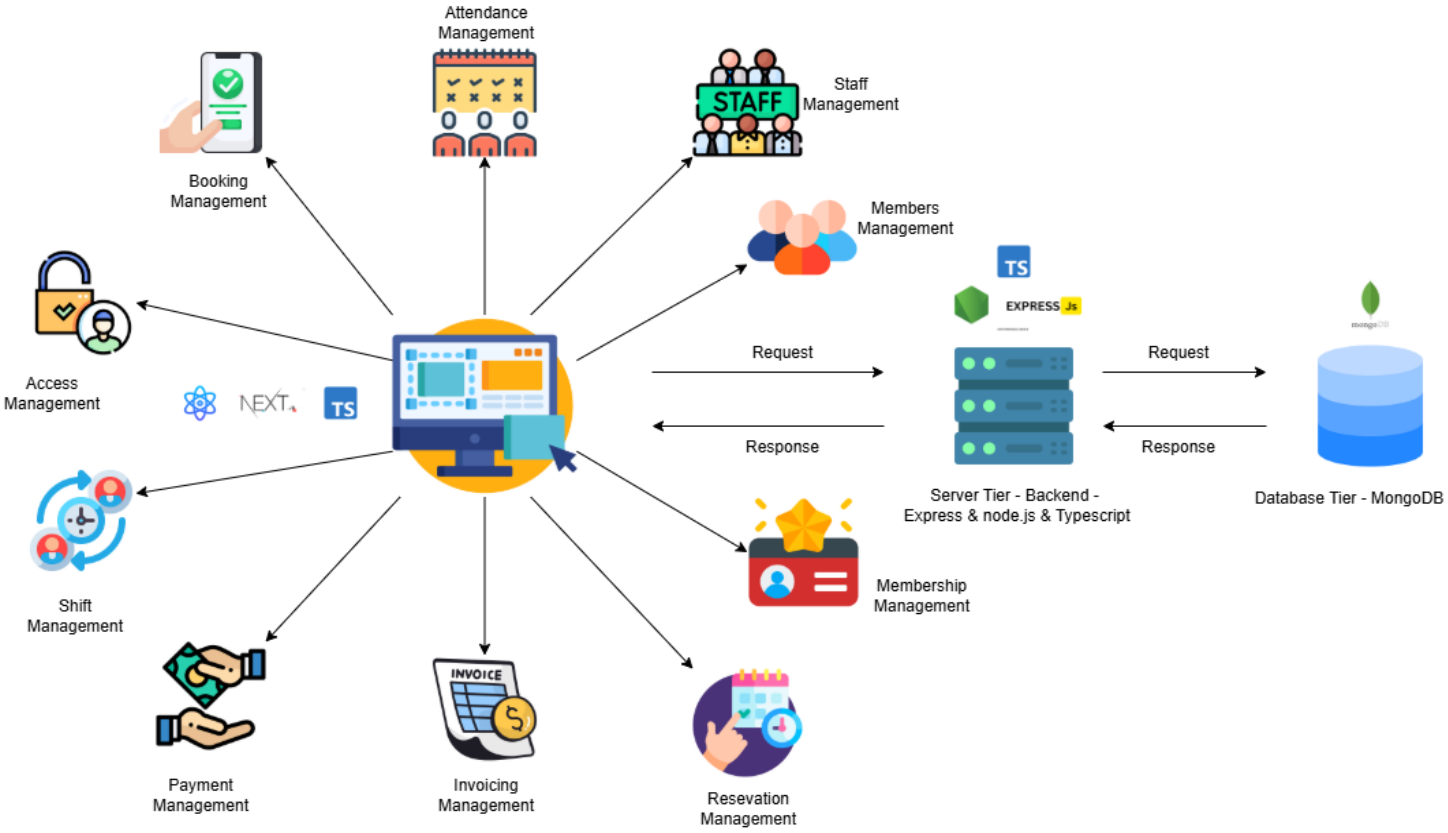
Use Case ID	02	
Name	Membership management	
Summary	Staff Register new member and add their profile	
Priority	01	
Pre-condition	Staff must Log in to the System	
Post-condition	Member details are updated in the database	
Primary actor	Staff	
Trigger	A new member joins or an existing member update details	
Main scenario	**Step**	**Action**
	1.	Staff selects “member management” option
	2.	Staff Enter member’s details
	3.	System validates entered data
	4.	Staff submits details
	5.	System stores member profile
	6.	System generates a membership ID
	7.	System sends confirmation to member
Extensions	**Step**	**Branching Action**
	3a	System notifies if email/phone already exists.
	3b	System notifies if membership type is invalid
	6a	System notifies if membership ID generation fails.

Use Case ID	03	
Name	Payment Management	
Summary	Member purchases goods (equipment, Supplements) and make payment	
Priority	01	
Pre-condition	Members must be logged in and select goods for purchase.	
Post-condition	Payment is recorded and receipt is generated	
Primary actor	Member	
Trigger	Member decides to purchase goods	
Main scenario	**Step**	**Action**
	1.	Member selects good from the inventory store
	2.	System displays product details
	3.	Member adds items to cart and proceeds to checkout
	4.	Member selects payment method
	5.	Members enter payment details
	6.	System validates the details
	7.	System processes payment
	8.	System updates inventory to reflect the sale
	9.	system records payment and generates receipt.
	10.	System sends confirmation to member
Extensions	**Step**	**Branching Action**
	6a	Payment details are incorrect. System prompt to enter payment details again.

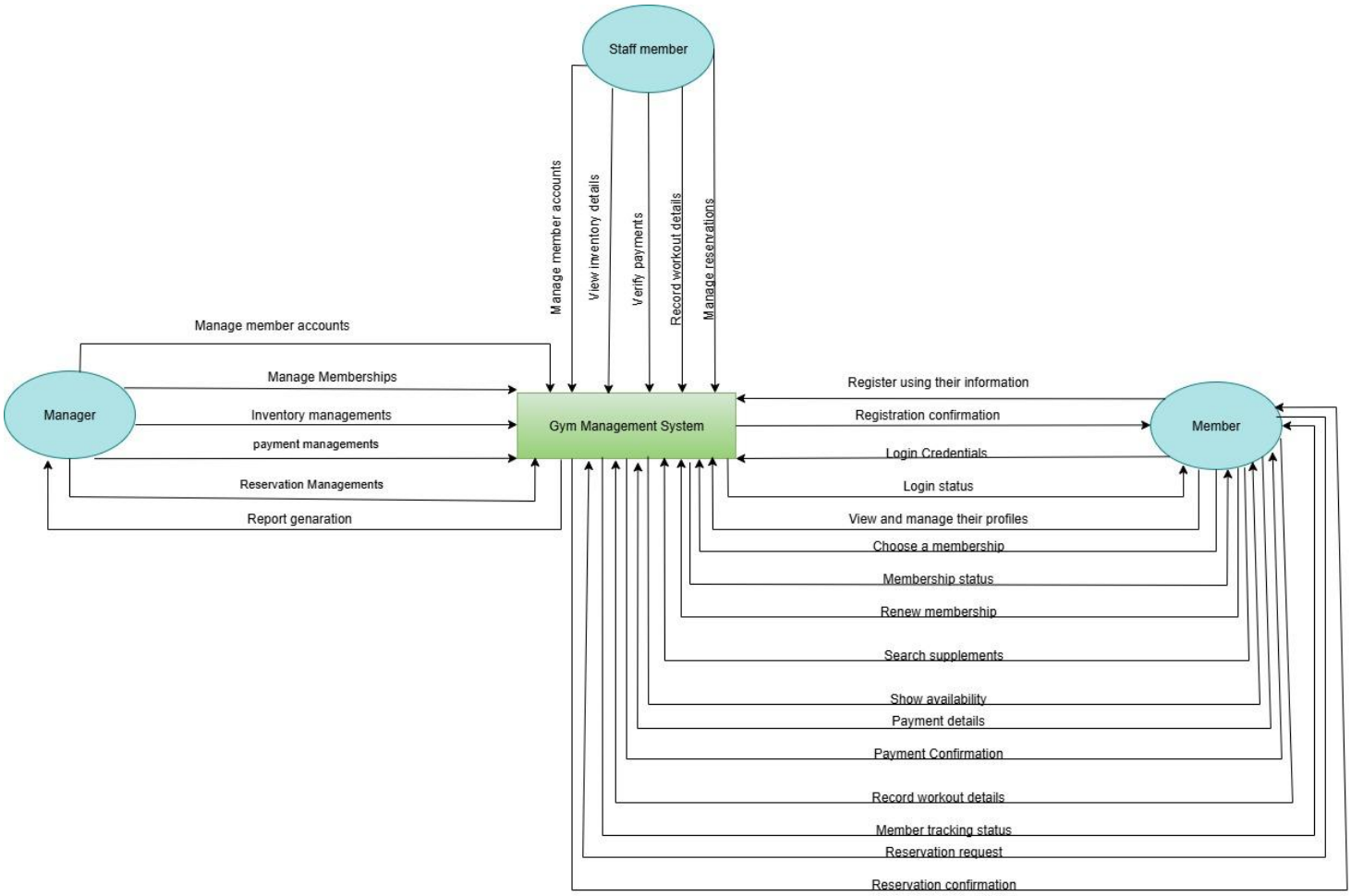
Use Case ID	04	
Name	Inventory Control	
Summary	Managers (and authorized staff) use the system to add, update, remove, and monitor gym equipment and supplement stock. The system issues alerts for low stock levels or equipment maintenance needs, ensuring smooth-operation.	
Priority	4	
Pre-condition	The manager or staff is logged into the system. The inventory module is accessible only to authorized roles.	
Post-condition	Inventory details are updated in the database. Alerts are generated for low stock or scheduled maintenance.	
Primary actor	Manager	
Trigger	New stock purchased, equipment added/removed, or item condition updated.	
Main scenario	**Step**	**Action**
	1.	The manager logs into the Gym Management System.
	2.	Navigates to the Inventory Control module.
	3.	Selects Add New Item, Update Item, or Remove Item.
	4.	The manager enters item details (name, type: equipment/supplement, quantity, condition, purchase date, and expiry date if applicable).
	5.	The system validates the entered details.
	6.	Item details are stored/updated in the inventory database.
	7	The system checks stock thresholds and generates low stock/maintenance alerts if necessary.
	8	Managers can generate and view inventory reports.
Extensions	**Step**	**Branching Action**
	3a	Invalid input (e.g., negative quantity) ; System rejects and prompts correct input.
	3b	Duplicate item detected; System suggests updating instead of creating.
	6a	If stock is below the threshold, the system generates a low stock alert.
	6b	If an item is flagged as faulty/expired, the system moves it to “inactive” stock.

Use Case ID	05	
Name	Attendance and Process Tracking	
Summary	The member checks in using a QR code, the system logs attendance, and the trainer updates workout logs for progress tracking.	
Priority	05	
Pre-condition	The member has active membership The trainer has access rights to update progress	
Post-condition	Payment is recorded and receipt is generated	
Primary actor	Member	
Trigger	Member scans QR code at gyms entrance	
Main scenario	**Step**	**Action**
	1.	The member logs into the system.
	2.	The system validates membership and QR codes.
	3.	Attendance is logged in the system.
	4.	The trainer logs work out details (sets, reps, weights).
	5.	The system updates member progress history.
	6.	The member views progress via the dashboard.
Extensions	**Step**	**Branching Action**
	2a	Invalid/expired membership; access denied
	3a	Duplicate check-in attempt; system blocks

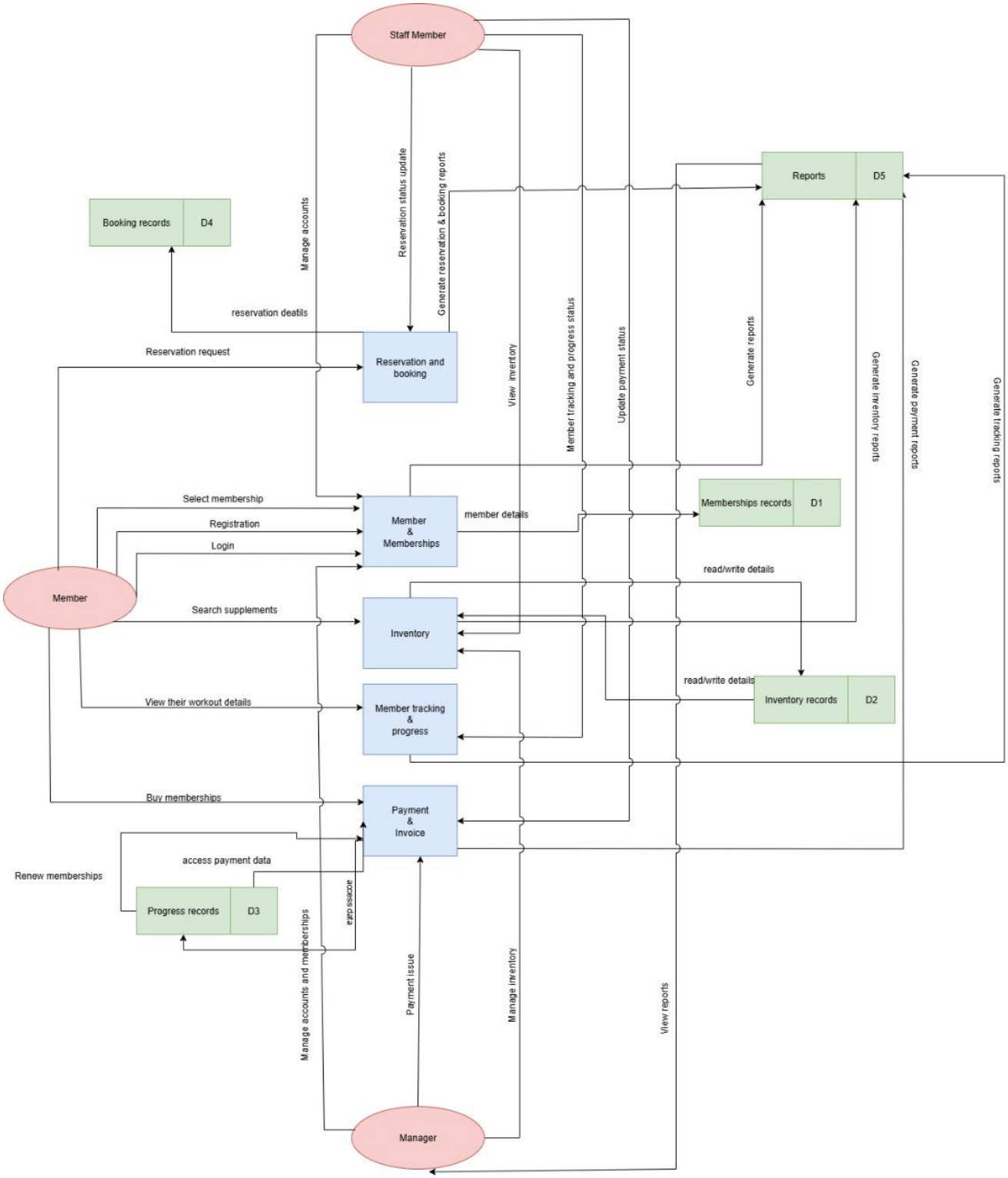
System Diagram



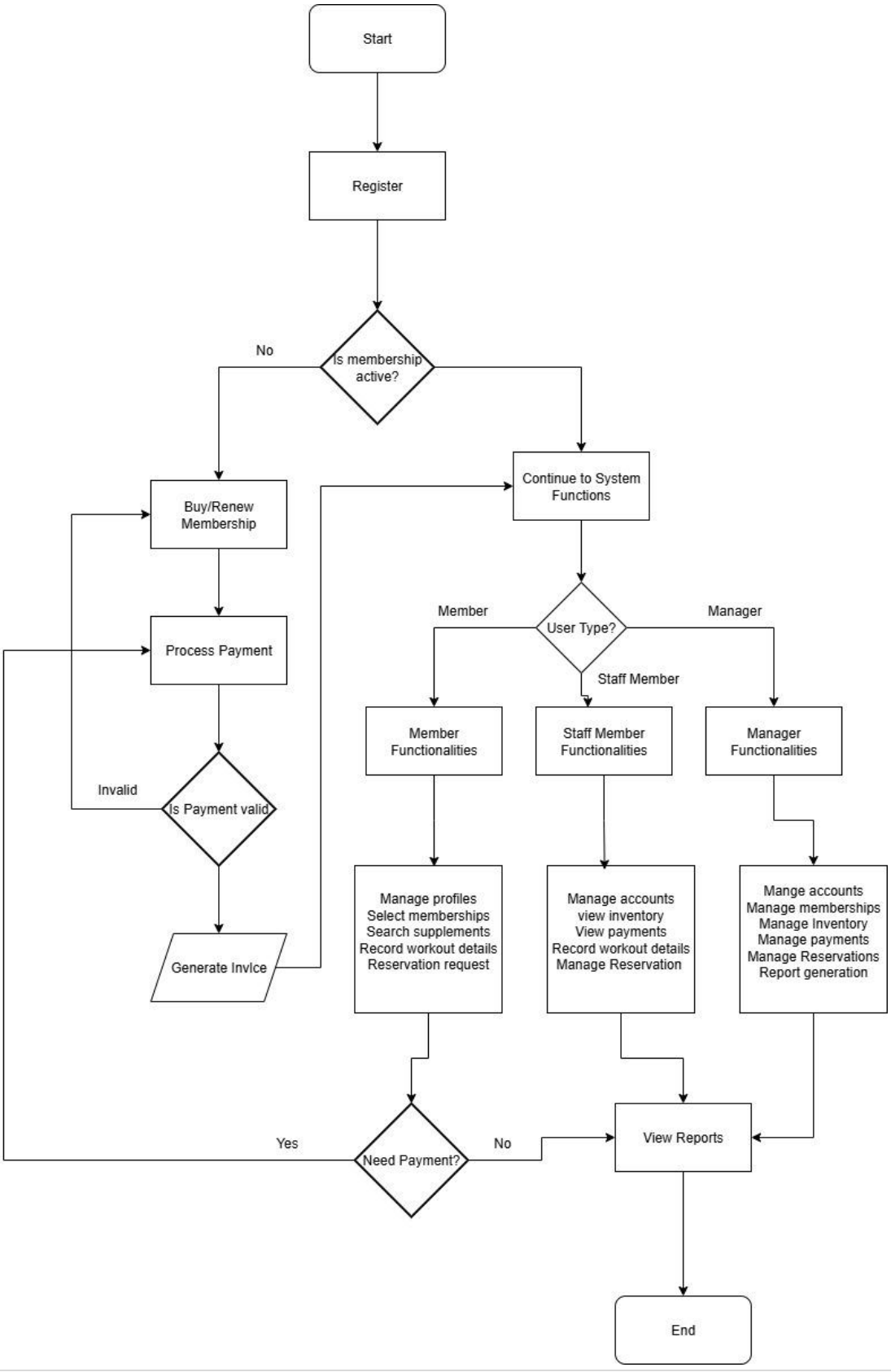
Data Flow Diagram



Data Flow Diagram



Flow chart


Start
Register
Is membership active?
No
Buy/Renew Membership
Process Payment
Is Payment valid
Invalid
Generate Invice
Continue to System Functions
User Type?
Member
Staff Member
Manager
Member Functionalities
Staff Member Functionalities
Manager Functionalities
Manage profiles
Select memberships
Search supplements
Record workout details
Reservation request
Manage accounts
view inventory
View payments
Record workout details
Manage Reservation
Mange accounts
Manage memberships
Manage Inventory
Manage payments
Manage Reservations
Report generation
Need Payment?
Yes
No
View Reports
End

Project Development Plan

Weekly Planning for Project Development

Week	Phase	Task
01	Project initialization and requirements gathering.	• Stakeholder interviews. • Observation of current process. • User requirement assessment.
02	Requirement Engineering	• User stories and Use case creation. • Identifying core business functions by refining user requirements. • Dividing responsibilities among members. • Choosing a suitable tech stack. • Project proposal creation.
03 - 04	High Level System Architecture Design and Development Environment setup	• Wireframing and UI mockups creation. • Database schema creation and refining. • Creating user workflows. • Creating system diagrams. • Setting up a GitHub repository.
05 - 08	Development Sprint 1 (Backend Development)	• Setting up backend project. • Creating API endpoints for business functions. • Database creation and connecting the database to the application. • Start developing backend. • Applying business logic to the backend.
09 - 10	Development Sprint 2 (Frontend Development and Integration) and Progress	• Progress report generation. • Developing the frontend. • Integrating API endpoints to the frontend. • Developing user interfaces and styling.
11 - 12	Testing and Optimization	• Unit testing and integration testing.

		• Testing and optimizing the user increasing user experience. • Test application security. • Final report preparation.
13 - 14	Delivery	• Delivering the web application. • Final presentation.

Work Distribution among the members

- User, Member and Membership Management –H R G N Karunathilaka
- Inventory Control - N M B M L B Nawaratne
- Payment and Invoice Management - K B P Kavisika
- Member Tracking and Progress Management - O A l De Silva
- Reservation and Booking Management - B Dhayabari