<Fitness Club Website/Membership Management System>

**Design Document**

**Version 1.0**



Group Id: <S230276DC0 >

Supervisor Name :< Abdullah Qamar >

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 29-08-2023 | 1.0 | It's become clear that the health and wellness industry helps people have a better life. Fitness centers and gyms only focus on exercise, but clubs offer a complete approach to health and wellness for everyone. Clubs need to keep their members happy while also getting new ones. That's why a Gym/Fitness Club Membership Management System is needed. This system will help gym and fitness clubs manage their members, attendance, fitness training, and profiles. It will make things easier and more convenient for club members. You have to develop a Gym/Fitness Club Membership Management System. This system will be designed to help gym and fitness clubs manage their memberships. | Bc190402690 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Introduction of Design Document](#One)
2. [Entity Relationship Diagram (ERD)](#ERD)
3. [Sequence Diagrams](#Six)
4. [Architecture Design Diagram](#Seven)
5. [Class Diagram](#class)
6. [Database Design](#databasedesign)
7. [Interface Design](#interfacedesign)
8. [Test Cases](#testcases)

1. **Introduction of Design Document**

Detailed Design Document Overview:

1. Title Page:

* Content:
  + Project Name: "Gym/Fitness Club Membership Management System"
  + Document Title: "Software Design Document"
  + Your Name and Contact Information
  + Date
* Purpose: This page serves as the cover of the document, providing essential project information and contact details.

2. Table of Contents:

* Content:
  + Lists all sections and subsections with their respective page numbers.
* Purpose: A quick reference for readers to navigate through the document efficiently.

3. Introduction:

* Content:
  + Briefly introduces the project, emphasizing its significance and the need for the Gym/Fitness Club Membership Management System.
  + Describes the objectives, goals, and key stakeholders involved.
* Purpose: Sets the context for the document, ensuring that readers understand the project's importance and what it aims to achieve.

4. System Architecture:

* Content:
  + Provides an in-depth overview of the system's architecture, including:
    - Detailed descriptions of major components and modules.
    - Technologies and frameworks used (PHP Laravel 9 for backend, React for frontend, MySQL for the database).
    - Integration methods (SMTP for emails, Trello for SMS).
* Purpose: Helps stakeholders and development teams understand how the system is structured and the technologies used to build it.

5. Functional Requirements:

* Content:
  + A comprehensive breakdown of each functional requirement specified in your project requirements, including:
    - Membership Sign-up and Renewal
    - Fee Management
    - Personalized Member Dashboard
    - RFID or Biometric Integration
    - Membership Analytics
    - Member Communication
    - Automated Notifications
    - Fitness Training Routines
    - Customer Diet Plans
* Purpose: Details what the system should do, ensuring alignment with project goals and user needs.

6. System Components:

* Content:
  + Identifies and describes major system components and modules (e.g., User Management, Membership Management, Attendance Tracking).
* Purpose: Organizes the development process, allowing for efficient division of work among teams or developers.

7. Data Model:

* Content:
  + Explains the database structure using MySQL, possibly including entity-relationship diagrams (ERDs).
* Purpose: Ensures data consistency and integrity, guiding the database design and implementation.

8. User Interfaces:

* Content:
  + Describes the user interface design using React, potentially including wireframes or mockups.
* Purpose: Provides a visual representation of the software, guiding the UI/UX design process.

9. Security Considerations:

* Content:
  + Addresses security concerns and outlines measures to protect the system (authentication, authorization, data encryption, etc.).
* Purpose: Ensures the safety of user data and system functionality, protecting against potential threats.

**10. Performance and Scalability:** - Content: - Discusses how the system will perform under various conditions and how it can be scaled to handle increased loads. - Purpose: Optimizes system performance and prepares it for growth, ensuring it can handle more users and data.

**11. Testing and Quality Assurance:** - Content: - Defines the testing approach (unit testing, integration testing, user acceptance testing), test cases, and tools. - Purpose: Ensures that the software is thoroughly tested, reducing the likelihood of bugs and improving quality.

**12. Deployment and DevOps**: - Content: - Outlines the deployment process, server infrastructure, and any CI/CD pipelines. - Purpose: Streamlines deployment and transitions the project from development to production efficiently.

**13. Maintenance and Support:** - Content: - Discusses plans for maintaining and supporting the software post-deployment, including bug tracking and issue resolution. - Purpose: Ensures the software remains reliable, up-to-date, and continues to provide value to users.

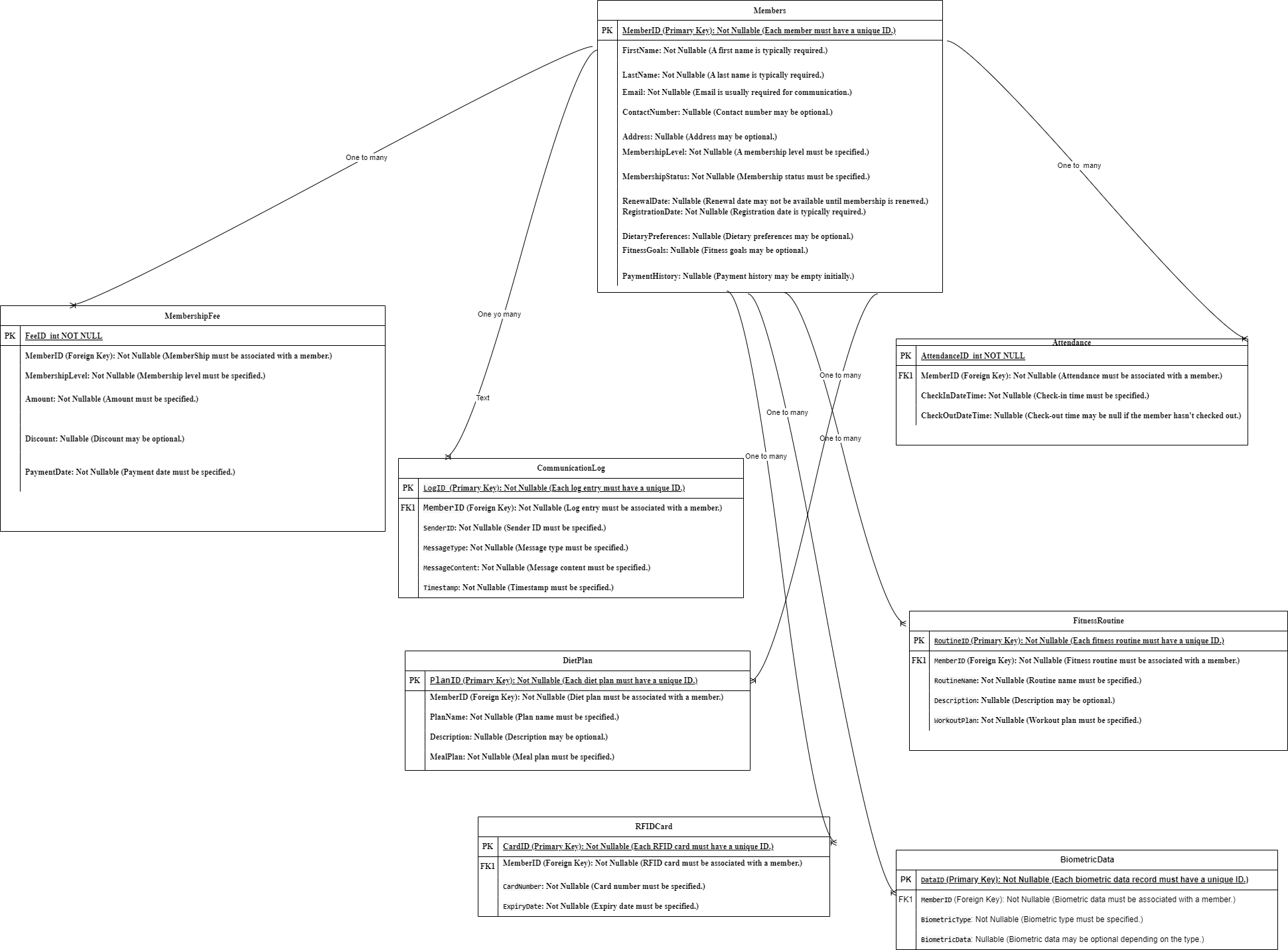
**14. Conclusion:** - Content: - Summarizes key points from the document. - Purpose: Provides a quick reference, reinforcing the document's main takeaways.

**15. References and Revision History:** - Content: - Cites external sources and tracks document revisions. - Purpose: Ensures transparency, accountability, and proper attribution of information and document changes.

Benefits and Purpose of the Design Phase in Software Development:

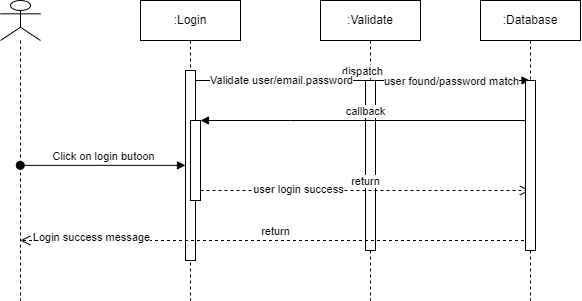
**The design phase plays a vital role in software development for several reasons:**

1. **Clarity and Alignment:** It ensures that all stakeholders have a shared understanding of the project's scope, goals, and features, reducing the potential for misunderstandings and misalignment.
2. **Risk Mitigation:** By identifying potential issues and vulnerabilities early in the project, the design phase helps in risk assessment and mitigation planning, reducing the likelihood of costly problems during development.
3. **Efficient Development:** A well-thought-out design serves as a blueprint for development, making it easier for developers to write code, design components, and collaborate effectively. This leads to more efficient development and reduced rework.
4. **Quality Assurance:** The design phase establishes testing requirements and strategies, helping ensure that the software is thoroughly tested, resulting in a higher-quality product with fewer defects.
5. **Cost and Time Savings:** Addressing architectural and design considerations upfront can save time and resources by avoiding costly changes and adjustments later in the development process.
6. **Client and Stakeholder Confidence:** A well-documented design phase instills confidence in clients and stakeholders that the project is well-planned and on track to meet their expectations, fostering trust and collaboration.
7. **Entity Relationship Diagram (ERD)**

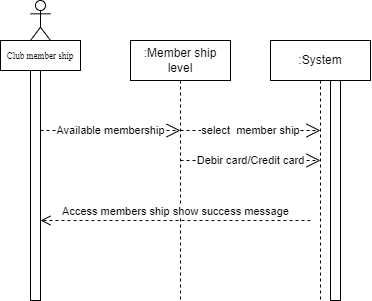


1. Sequence Diagrams

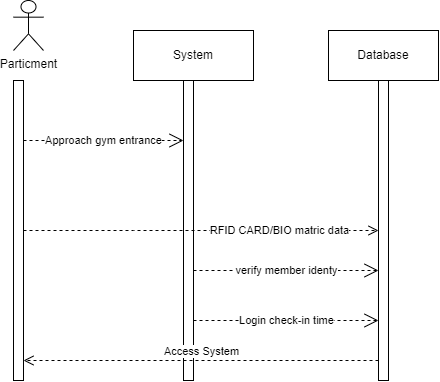
Member Login



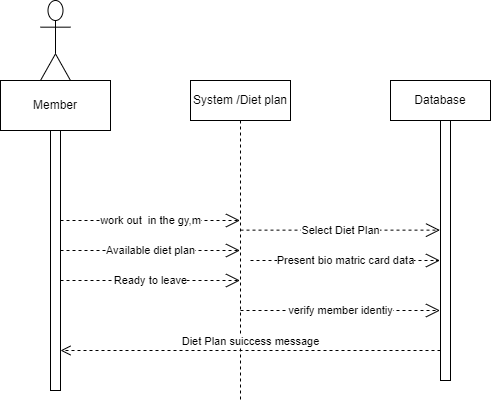
Membership Fee



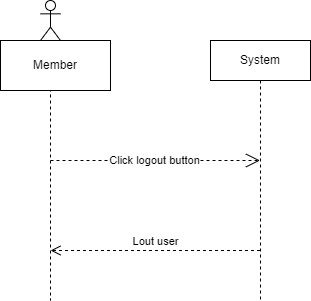
Verify user and Access System



Diet Plan



Logout



1. Architecture Design Diagram

**Web Application Architecture Components**

Typically web-based application architecture comprises 3 core components:

1) **Web Browser**: The browser or the client-side component or the front-end component is the key component that interacts with the user, receives the input and manages the presentation logic while controlling user interactions with the application. User inputs are validated as well, if required.

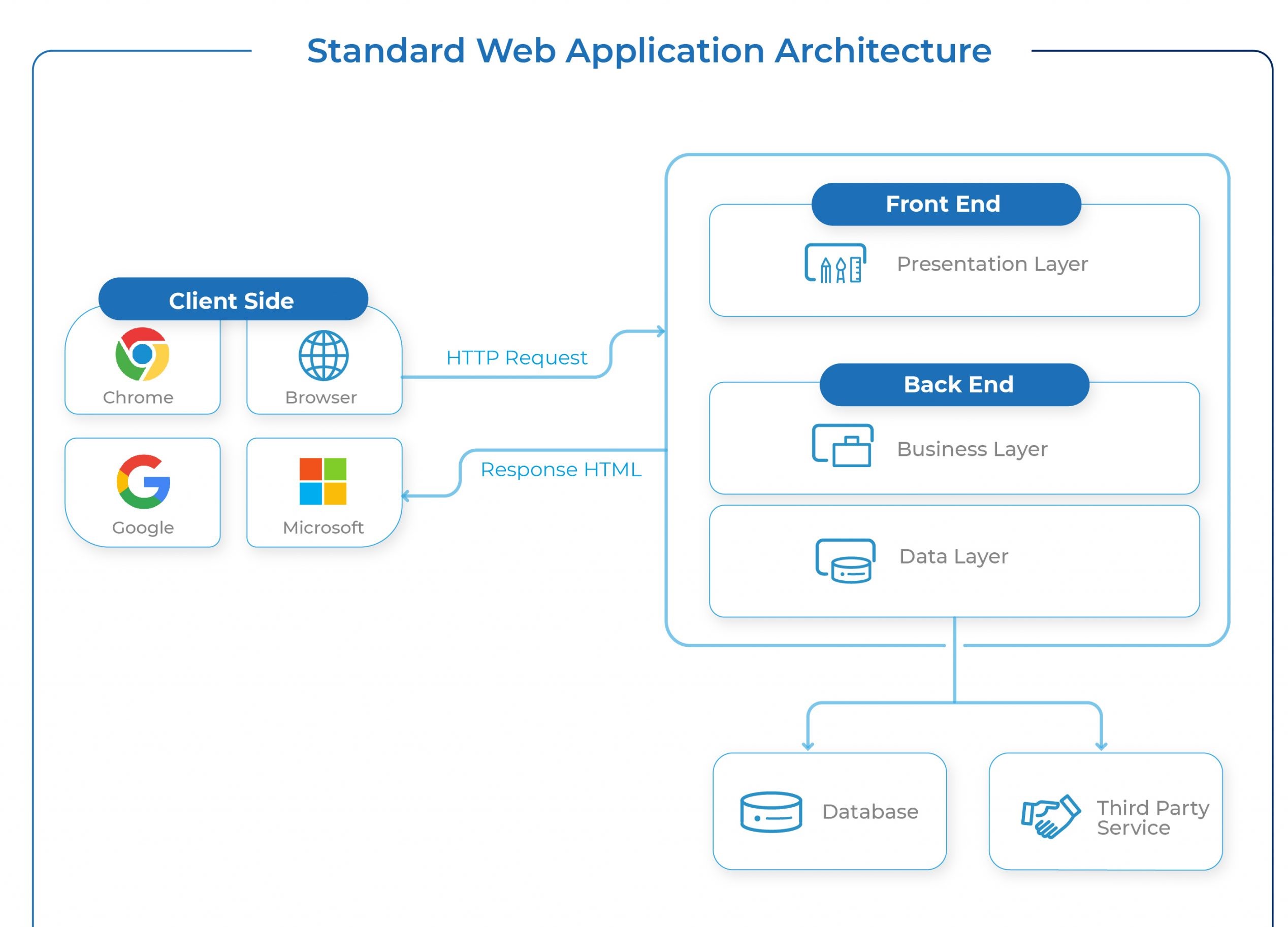
2) **Web Server:**The web server also known as the backend component or the server-side component handles the business logic and processes the user requests by routing the requests to the right component and managing the entire application operations. It can run and oversee requests from a wide variety of clients.

3) **Database Server**: The database server provides the required data for the application. It handles data-related tasks. In a multi-tiered architecture, database servers can manage business logic with the help of stored procedures.

**Benefits of Tiered Architecture:**

* **Separation of Concerns:** Tiered architecture separates user interface, business logic, and data storage, making the system more modular and maintainable.
* **Scalability:** Individual tiers can be scaled independently to accommodate increased user load or data volume.
* **Security:** Security measures can be implemented at each tier to protect sensitive data and ensure secure user access.
* **Flexibility:** Allows for the potential integration of additional services or components in the future without major system restructuring.
* **Maintainability:** Easier maintenance and updates as changes in one tier do not necessarily affect the others.

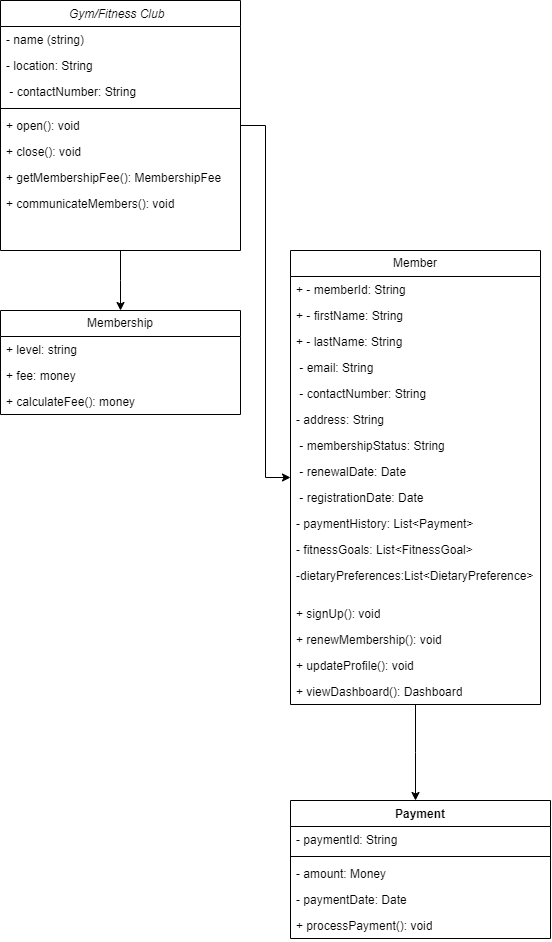
**Diagram**



1. Class Diagram

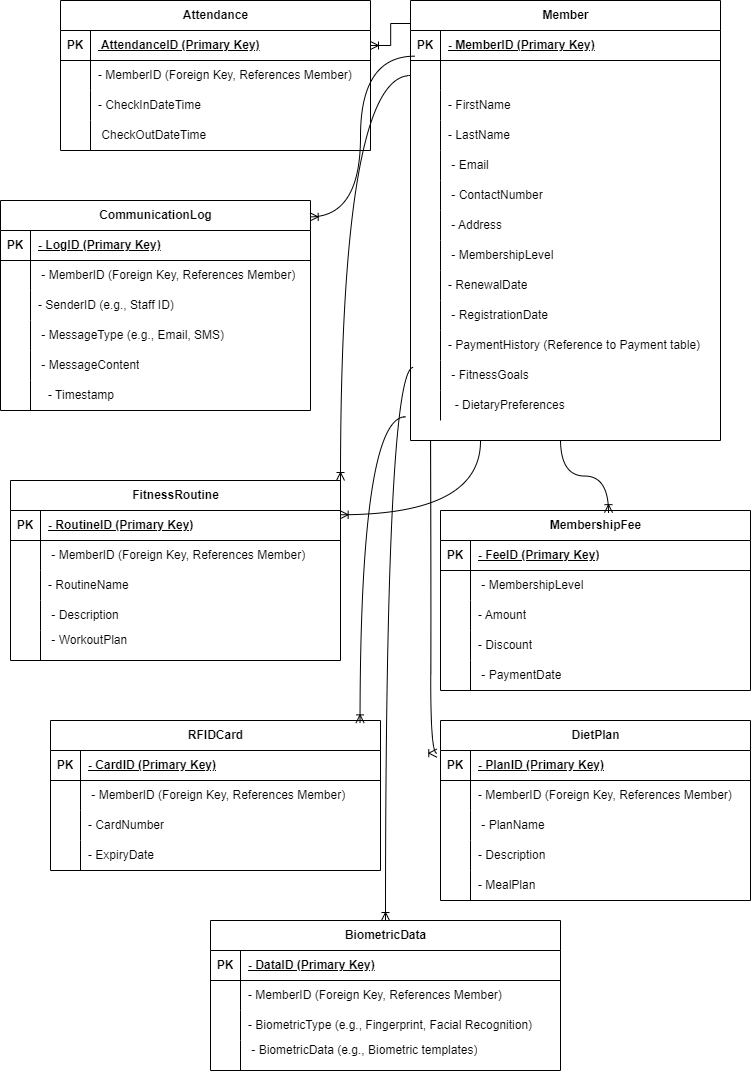
Class diagrams are a valuable tool for designing, documenting, and communicating the structure of a software system. They contribute to better software quality, maintainability, and collaboration among development teams and stakeholders.

Diagram



1. Database Design

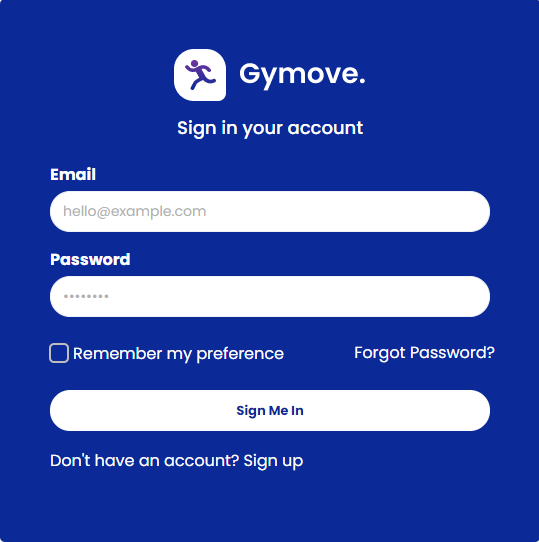
Database design is a critical aspect of software development that impacts data quality, system performance, and overall system reliability. A well-designed database system can lead to more efficient, secure, and adaptable applications, ultimately enhancing the value of the software solution.



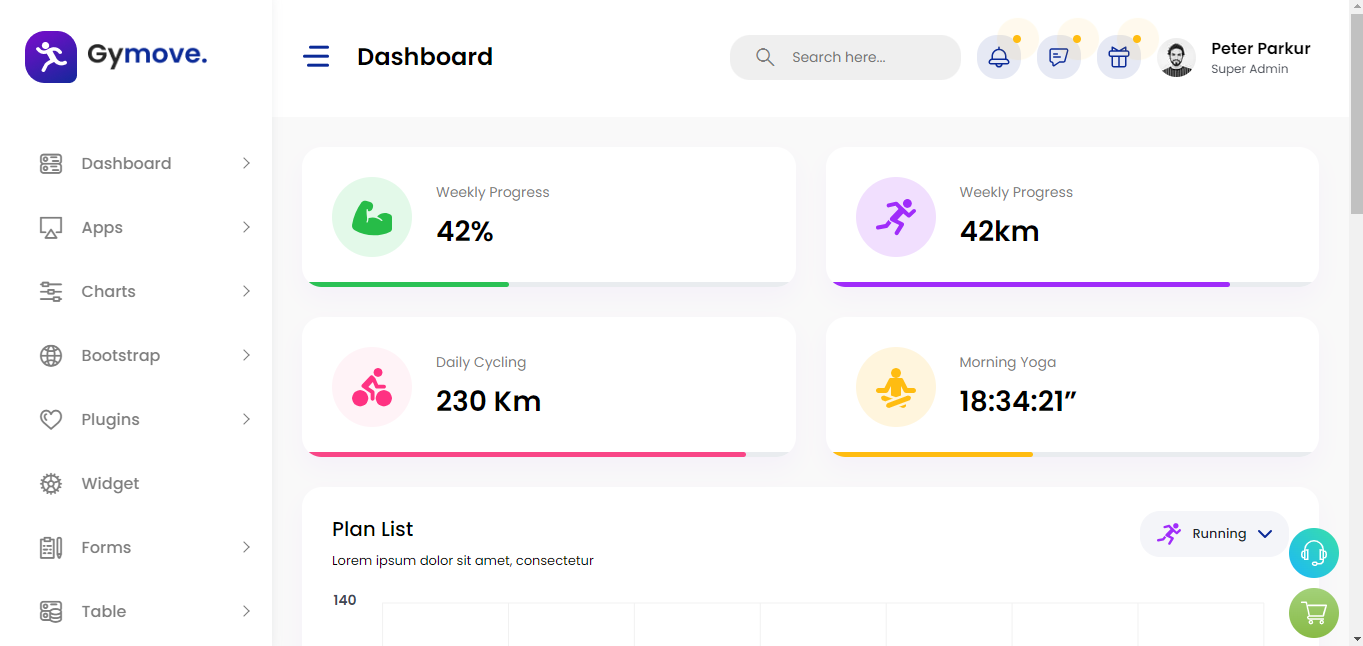
1. Interface Design

The GUI serves as a critical interface between users and software applications, facilitating user interaction, enhancing usability, and ultimately determining the overall user experience. An effective GUI design is essential for making software more user-friendly, efficient, and appealing to a wide range of users.

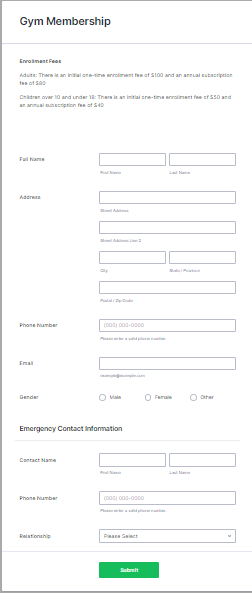
Screenshot:



Screenshot:



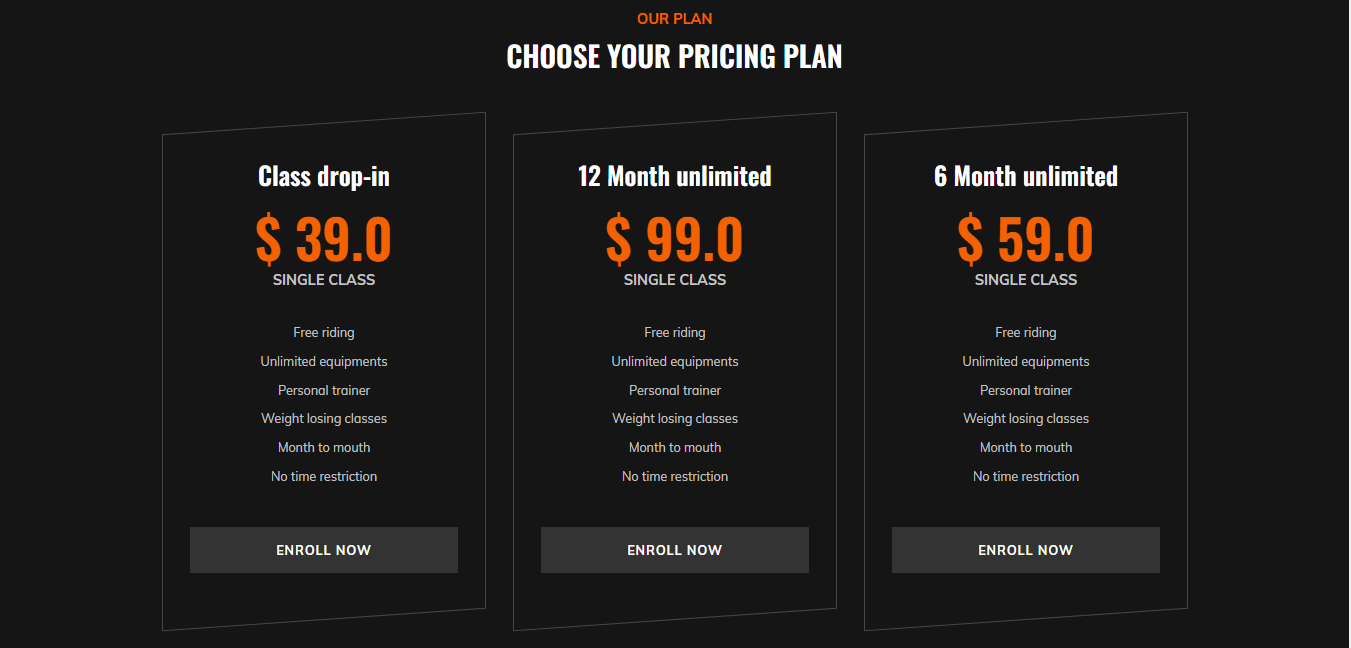
Screenshot



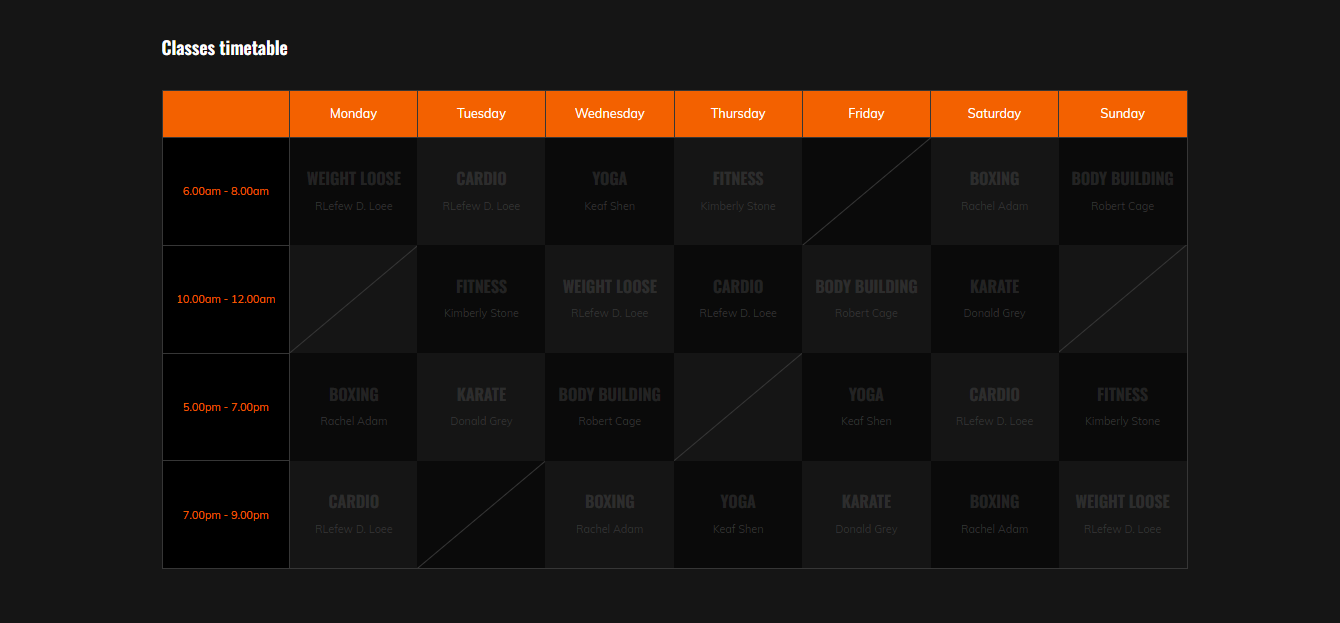
Screenshot



Screenshot



Screenshot



1. Test Cases

|  |  |  |
| --- | --- | --- |
| **Title: Test new member sign-up with a basic membership** | |  |
| **Test case ID** | TC \_1 | |
| **Pre-Conditions** | User is on the membership sign-up page. | |
| **Actions** | * Select membership level: Basic. * Provide valid information. * Submit payment. | |
| **Expected Result** | Member successfully signs up, and membership status is "Active." Payment history is updated. | |
| **Authority** | Admin, | |
| **Tested By** | BC190402690 | |
| **Result**  **(Pass/Fail)** | Pass | |

|  |  |  |
| --- | --- | --- |
| **Title: Test membership renewal for an existing member** | |  |
| **Test case ID** | TC \_2 | |
| **Pre-Conditions** | User is logged in as an existing member. | |
| **Actions** | * Navigate to membership renewal. * Select renewal option. * Submit payment. | |
| **Expected Result** | Member's membership is successfully renewed, and renewal date is updated.  Payment history is updated. | |
| **Authority** | Admin, | |
| **Tested By** | BC190402690 | |
| **Result**  **(Pass/Fail)** | Pass | |

|  |  |  |
| --- | --- | --- |
| **Title: Test member login and access to the dashboard** | |  |
| **Test case ID** | TC \_3 | |
| **Pre-Conditions** | Test member login and access to the dashboard | |
| **Actions** | * Enter valid credentials. * Click "Login." | |
| **Expected Result** | Member is redirected to their personalized dashboard. | |
| **Authority** | Admin, | |
| **Tested By** | BC190402690 | |
| **Result**  **(Pass/Fail)** | Pass | |

|  |  |  |
| --- | --- | --- |
| **Title: Test updating member's personal information** | |  |
| **Test case ID** | TC \_4 | |
| **Pre-Conditions** | Member is logged in and on the dashboard. | |
| **Actions** | * Navigate to profile settings. * Update contact number and address * Save changes. | |
| **Expected Result** | Member's contact information is successfully  updated. | |
| **Authority** | Admin, | |
| **Tested By** | BC190402690 | |
| **Result**  **(Pass/Fail)** | Pass | |

|  |  |  |
| --- | --- | --- |
| **Title: Test member check-in using an RFID card** | |  |
| **Test case ID** | TC \_5 | |
| **Pre-Conditions** | Test member check-in using an RFID card | |
| **Actions** | * Approach the gym entrance. * Present the RFID card. * Verify successful check-in. | |
| **Expected Result** | Member is allowed access, and check-in time is recorded. | |
| **Authority** | Admin, | |
| **Tested By** | BC190402690 | |
| **Result**  **(Pass/Fail)** | Pass | |

|  |  |  |
| --- | --- | --- |
| **Title: Test member check-out using an RFID card** | |  |
| **Test case ID** | TC \_6 | |
| **Pre-Conditions** | Member has already checked in. | |
| **Actions** | * Approach the gym exit. * . Present the RFID card again. * Verify successful check-out. | |
| **Expected Result** | Member is allowed to exit, and check-out  time is recorded. | |
| **Authority** | Admin, | |
| **Tested By** | BC190402690 | |
| **Result**  **(Pass/Fail)** | Pass | |

|  |  |  |
| --- | --- | --- |
| **Title: Test Membership Analytics** | |  |
| **Test case ID** | TC \_7 | |
| **Pre-Conditions** | System is operational and data is available | |
| **Actions** | * Navigate to Membership Analytics. * Select the desired analytics report. * View the report and data. | |
| **Expected Result** | Analytics report displays relevant data accurately. | |
| **Authority** | Admin or Authorized User | |
| **Tested By** | BC190402690 | |
| **Result**  **(Pass/Fail)** | Pass | |

|  |  |  |
| --- | --- | --- |
| **Title: Test Member Communication** | |  |
| **Test case ID** | TC \_8 | |
| **Pre-Conditions** | User has the privilege to send messages. | |
| **Actions** | * Compose a message to a member. * Send the message via email or SMS. | |
| **Expected Result** | The member receives the message via the  selected communication channel. | |
| **Authority** | Admin or Authorized User | |
| **Tested By** | BC190402690 | |
| **Result**  **(Pass/Fail)** | Pass | |

|  |  |  |
| --- | --- | --- |
| **Title: Test Automated Notifications** | |  |
| **Test case ID** | TC \_9 | |
| **Pre-Conditions** | The system has members with overdue status. | |
| **Actions** | * Trigger automated notifications for * members with overdue status. | |
| **Expected Result** | Members with overdue status receive  automated notifications encouraging them  to return. | |
| **Authority** | Admin or Authorized User | |
| **Tested By** | BC190402690 | |
| **Result**  **(Pass/Fail)** | Pass | |