

Project Title	Gym Membership Management System
Author	Karthikeyna M
Documented Date	14-11-2024

Gym Membership Management System

SRS Document

1. Introduction

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Project Scope

1.5 References

2. Overall Description

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 Assumptions and Dependencies

3. System Features

3.1 Functional Requirements

3.2 Usecase Diagram

4. External Interface Requirements

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communications Interfaces

5. Non-functional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

1. Introduction

1.1 Purpose

The purpose of this document is to outline the requirements for a Gym Membership Management System that allows administrators to manage gym memberships, track attendance, and maintain records for members, including payment and membership plans.

1.2 Document Conventions

This document follows standard conventions for SRS documents, utilizing IEEE formatting. Key requirements are numbered for easy reference.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Developers implementing the system
- Project managers overseeing the project
- Stakeholders and gym management team members

1.4 Project Scope

The system will manage membership data, attendance records, and payment information for a gym. The application will feature a user-friendly interface, data persistence using JDBC, and support for multiple membership plans with inheritance for different membership types.

1.5 References

- IEEE Software Requirements Specification Template
- SQL Database Documentation
- Java Documentation for JDBC and Collections

2. Overall Description

2.1 Product Perspective

This application is a standalone system that automates membership management for gyms, reducing manual record-keeping. The system integrates with a MySQL database for data storage and retrieval and will have a desktop or web-based interface.

2.2 Product Features

- **Membership management:** Add, update, and remove members.
- **Attendance tracking:** Record member attendance and calculate attendance records.
- **Payment processing:** Manage membership payments and generate payment receipts.
- **Membership plans:** Different types of memberships (e.g., Basic, Premium) with unique features and pricing.
- **File handling:** Save and retrieve data related to members, attendance, and payments.
- **Reporting:** Generate attendance and payment reports.

2.3 User Classes and Characteristics

- **Admin:** Manages memberships, attendance, and payments.
- **Member:** Views their own membership details, attendance, and payment history.

2.4 Operating Environment

- **Software:** Java JDK, MySQL, JDBC
- **Hardware:** Compatible with desktop or laptop with at least 4GB RAM and 1GHz processor.
- **Operating System:** Cross-platform support for Windows, macOS, and Linux.

2.5 Design and Implementation Constraints

- System must use Java for core functionalities and JDBC for database integration.
- MySQL database schema as outlined in the provided SQL script.
- Compliance with gym data security policies.

2.6 Assumptions and Dependencies

- Database server is properly configured and accessible.
- Java Runtime Environment (JRE) is installed on all machines running the application.

3. System Features

3.1 Functional Requirements

3.1.1 Membership Management

- **FR1.1:** The system shall allow admins to add new members.
- **FR1.2:** The system shall allow admins to update member information.
- **FR1.3:** The system shall allow admins to delete members.

3.1.2 Membership Plans

- **FR2.1:** The system shall support different membership types (e.g., Basic, Premium) with unique pricing.
- **FR2.2:** Each membership type shall inherit basic membership attributes and may override features as needed.

3.1.3 Attendance Tracking

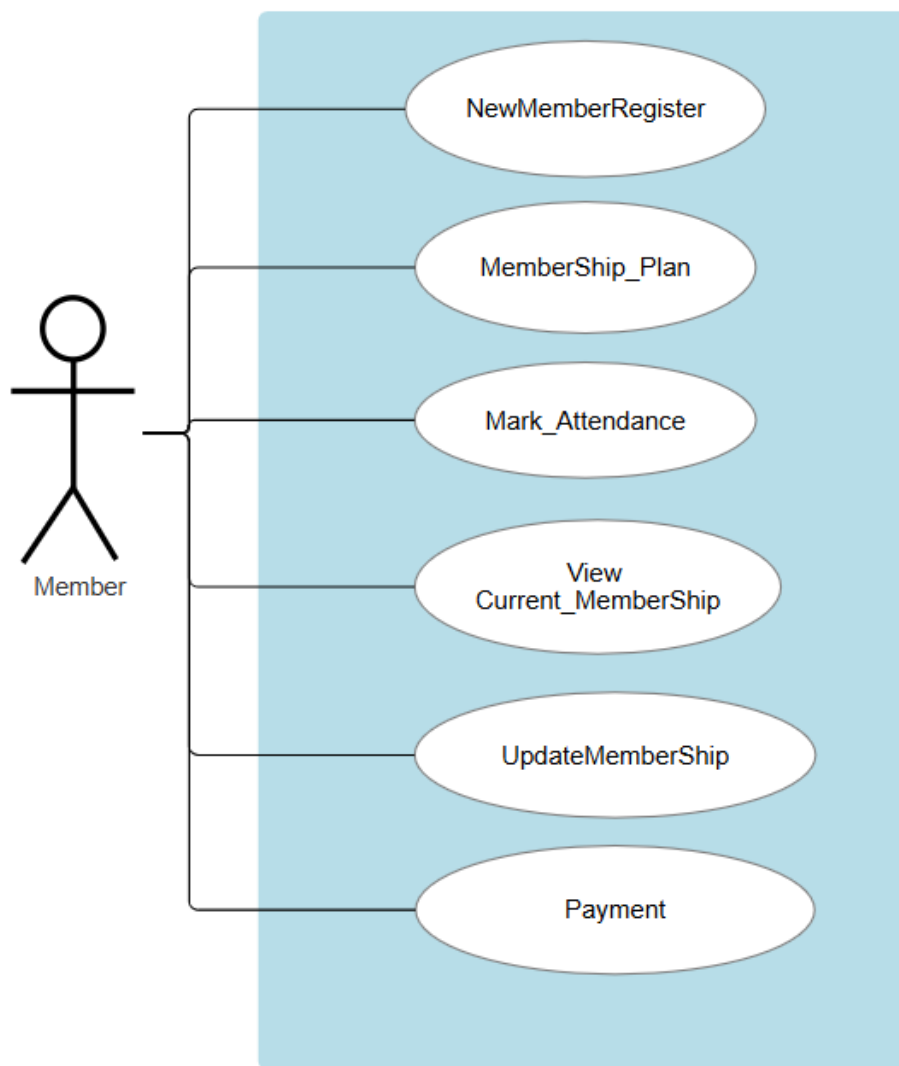
- **FR3.1:** The system shall record check-in and check-out times.
- **FR3.2:** Attendance records shall be linked to individual members.

3.1.4 Payment Management

- **FR4.1:** The system shall record payment details for each membership.
- **FR4.2:** Payment records shall include payment date, amount, method, and status.

3.2 Use Case Diagram

A use case diagram will be included to illustrate interactions between the user classes and the system functionalities, detailing actions like "Add Member," "Track Attendance," "Process Payment," and "View Membership."



4. External Interface Requirements

4.1 User Interfaces

The system will feature a GUI where admins can add members, track attendance, and manage payments. Members will have access to view their membership details and attendance history.

4.2 Hardware Interfaces

No direct hardware interfaces other than typical I/O devices such as keyboard and mouse.

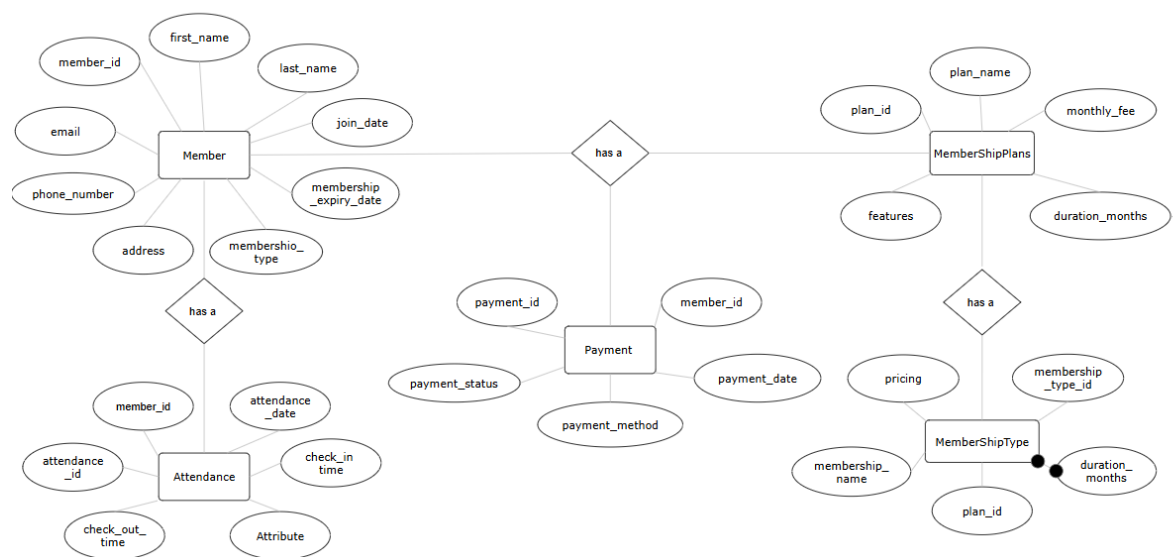
4.3 Software Interfaces

- **Database:** MySQL for data storage.
- **Java JDBC:** For database interactions.

4.4 Communications Interfaces

No network-based communication is required; the application is expected to run locally on individual machines.

4.5 Entites Relation Diagram



5. Non-functional Requirements

5.1 Performance Requirements

- The system should handle up to 1,000 active members without significant performance degradation.
- Database queries should execute within 2 seconds on average.

5.2 Safety Requirements

- Regular backups of the database are required to prevent data loss.

- Access to member information should be restricted to authorized personnel.

5.3 Security Requirements

- Data should be encrypted when stored.
- System access requires authentication for all users.

5.4 Software Quality Attributes

- **Usability:** Interface must be intuitive and accessible to non-technical users.
- **Maintainability:** Codebase should follow Java best practices to allow easy updates and bug fixes.
- **Scalability:** System should be scalable to support additional functionalities in the future.
- **Reliability:** System must accurately manage data and minimize errors in data handling.