**Software Design Specification**

**Project Title:** Gym Management System

**Version:** 1.0

**Project Code:** [To be assigned]

**Supervisor:** [To be added]

**Co-Supervisor:** [To be added]

**Project Team:**

* Muhammad Yousuf Rehan (22K-4457)
* Azmeer Un Nabi (22K-4235)
* Ateeb Azam (22K-4205)

**Submission Date:** [To be added]

**Document Sign-Off**

| **Version** | **Sign-off Authority** | **Project Role** | **Signature** | **Sign-off Date** |
| --- | --- | --- | --- | --- |
| 1.0 |  |  |  |  |

**Document Information**

| **Category** | **Information** |
| --- | --- |
| Customer | FAST-NU |
| Project | Gym Management System |
| Document | Software Design Specification |
| Version | 1.0 |
| Status | Draft |
| Authors | Muhammad Yousuf Rehan, Azmeer Un Nabi, Ateeb Azam |
| Approvers | [To be added] |
| Issue Date | [To be added] |
| Location | [To be added] |
| Distribution | Advisor, Project Coordinator's Office |

**Definition of Terms, Acronyms and Abbreviations**

| **Term** | **Description** |
| --- | --- |
| GUI | Graphical User Interface |
| DBMS | Database Management System |
| CRUD | Create, Read, Update, Delete |
| UML | Unified Modeling Language |
| MVC | Model View Controller |

**Table of Contents**

1. Introduction
2. Design Considerations
3. System Architecture
4. Design Strategy
5. Detailed System Design
6. References
7. Appendices

**1. Introduction**

**1.1 Purpose of Document**

This document outlines the design architecture for the Gym Management System. It serves as a reference for developers, evaluators, and maintainers. Object-oriented design methodology is used.

**1.2 Intended Audience**

* Course Instructor
* Project Supervisor
* Evaluators
* Peer Developers

**1.3 Document Convention**

* Font: Times New Roman
* Font Size: Headings 14pt, Body 12pt

**1.4 Project Overview**

The Gym Management System is a desktop application designed for gyms to manage their daily operations such as member registration, trainer assignments, attendance, and payment management.

**1.5 Scope**

**Included:**

* Admin login and CRUD operations
* Trainer and member management
* Payment tracking
* Package and equipment records
* Attendance recording

**Excluded:**

* Integration with third-party APIs
* Real-time sensor inputs
* Mobile app version

**2. Design Considerations**

**2.1 Assumptions and Dependencies**

* Admin will operate the system through desktop interface.
* System runs on Windows OS.
* Backend database is MySQL.
* Java is used for development.

**2.2 Risks and Volatile Areas**

* Changes in gym policy affecting membership structure.
* Potential upgrade to web-based or mobile-based system.
* Limited validation of data on frontend.

**3. System Architecture**

**3.1 System Level Architecture**

The system follows a modular approach:

* **UI Layer:** Java Swing forms
* **Business Logic Layer:** Java classes and methods
* **Database Layer:** MySQL

Communication between UI and DB through DAO classes.

**3.2 Software Architecture**

* **User Interface Layer:** Forms for login, dashboard, and records
* **Middle Layer:** Handles validation, computation, and method calls
* **Data Layer:** Interacts with MySQL for CRUD operations

**4. Design Strategy**

* Use of OOP principles for scalability
* Code reuse through modular classes
* MySQL normalization to avoid redundancy
* Future migration option to web-based version

**Test Cases**

**Test Case 1: Admin Login**

* **Test Case ID:** TC001
* **Input:** Valid username and password
* **Expected Output:** Admin dashboard displayed
* **Result:** Pass/Fail

**Test Case 2: Invalid Login**

* **Test Case ID:** TC002
* **Input:** Incorrect username or password
* **Expected Output:** Error message shown
* **Result:** Pass/Fail

**Test Case 3: Add Trainer**

* **Test Case ID:** TC003
* **Input:** Trainer details (name, phone, salary, shift)
* **Expected Output:** Trainer added successfully
* **Result:** Pass/Fail

**Test Case 4: View Trainer List**

* **Test Case ID:** TC004
* **Input:** Navigate to trainer list
* **Expected Output:** List of all trainers displayed
* **Result:** Pass/Fail

**Test Case 5: Add Member**

* **Test Case ID:** TC005
* **Input:** Member details (name, age, trainer, package)
* **Expected Output:** Member added successfully
* **Result:** Pass/Fail

**Test Case 6: Record Attendance**

* **Test Case ID:** TC006
* **Input:** Select member and date
* **Expected Output:** Attendance recorded
* **Result:** Pass/Fail

**Test Case 7: Generate Payment Report**

* **Test Case ID:** TC007
* **Input:** Select time range
* **Expected Output:** List of payments generated
* **Result:** Pass/Fail

**Test Case 8: Add Equipment**

* **Test Case ID:** TC008
* **Input:** Equipment name, quantity, purchase date
* **Expected Output:** Equipment added to inventory
* **Result:** Pass/Fail

**Test Case 9: Update Package**

* **Test Case ID:** TC009
* **Input:** Modify existing package details
* **Expected Output:** Package updated in DB
* **Result:** Pass/Fail

**Test Case 10: Logout**

* **Test Case ID:** TC010
* **Input:** Click logout button
* **Expected Output:** Return to login screen
* **Result:** Pass/Fail

**5. Detailed System Design**

**5.1 Database Design**

**5.1.1 ER Diagram**

(Entity-Relationship diagram showing Admin, Trainer, Member, Package, Payment, Attendance, and Equipment relationships.)

**5.1.2 Data Dictionary**

**Table: admin**

| **Column Name** | **Description** | **Type** | **Length** | **Nullable** | **Default** | **Key Type** |
| --- | --- | --- | --- | --- | --- | --- |
| id | Admin ID | INT | 11 | No | - | PK |
| username | Login ID | TEXT | - | No | - |  |
| password | Password | TEXT | - | No | - |  |

**Table: trainer**

| **Column Name** | **Description** | **Type** | **Length** | **Nullable** | **Default** | **Key Type** |
| --- | --- | --- | --- | --- | --- | --- |
| id | Trainer ID | INT | 11 | No | - | PK |
| name | Full Name | TEXT | - | No | - |  |
| phone | Contact Number | VARCHAR | 11 | Yes | - |  |
| salary | Monthly Salary | INT | 11 | Yes | - |  |
| shift | Work Shift | TEXT | - | Yes | - |  |

(...similar for member, attendance, payment, equipment, and package...)

**5.2 Application Design**

**5.2.1 Sequence Diagrams**

**Login Flow:**

* User enters credentials → Submit → Check DB → Success → Dashboard

**Add Trainer:**

* Admin → AddTrainerForm → Enter details → Save → DAO Insert → DB Update

**5.2.2 State Diagrams**

**Admin State Diagram:**

* Logged Out → Login → Dashboard → (Manage: Trainer | Member | Payment | Logout)

**Member Registration State:**

* Idle → New Member → Form Filled → Save → Success or Error → Idle

**6. References**

* Java Oracle Docs: <https://docs.oracle.com/en/java/>
* MySQL Documentation: <https://dev.mysql.com/doc/>
* UML Diagrams: <https://www.uml-diagrams.org/>

**7. Appendices**

* GUI screenshots
* Test cases
* Sample SQL script
* Installation guide

**End of Document**