**1. Project Title**

**Gym Management System**

**2. Objective**

The objective of this project is to develop a Gym Management System that provides functionalities to manage members, trainers, membership plans, payments, and personal details. The system will operate through a command-line interface, using Java and Hibernate ORM for data persistence.

**3. Scope of the Project**

The Gym Management System will assist gym administrators in:

* Storing and retrieving member, trainer, and membership plan details.
* Managing payment information.
* Utilizing Hibernate to handle database interactions efficiently.

The system operates through commands for storing, showing, and retrieving information from the database.

**4. Features of the System**

1. **Member Management**:
   * Add, update, and retrieve member information.
   * Track member subscriptions and status.
2. **Trainer Management**:
   * Add, update, and retrieve trainer details.
3. **Membership Plan**:
   * Define and manage different membership plans (e.g., monthly, yearly).
4. **Payment Management**:
   * Record payments for member subscriptions.
5. **CRUD Operations**:
   * Store, retrieve, and display data using command-line-based interaction.

**5. System Requirements**

**5.1 Hardware Requirements:**

* **Processor**: Intel Core i5 or higher
* **RAM**: 4GB or higher
* **Hard Disk**: 500GB
* **Operating System**: Windows/Linux/Mac

**5.2 Software Requirements:**

* **Programming Language**: Java 8 or higher
* **Database**: MySQL
* **Framework**: Hibernate ORM (Object Relational Mapping)
* **IDE**: IntelliJ IDEA/Eclipse
* **Build Tool**: Maven (for dependency management)

**6. Technology Stack**

* **Frontend**: Command-line interface
* **Backend**: Java (with Hibernate ORM for persistence)
* **Database**: MySQL
* **Tools**:
  + Hibernate for database interaction.
  + Maven for dependency management.
  + MySQL for the relational database.

**7. System Architecture**

The system follows a **two-tier architecture**:

1. **Command-Line Interface (CLI)**: Allows gym administrators to interact with the system through commands such as storing, retrieving, and displaying data.
2. **Data Access Layer**: The Hibernate ORM framework is used to interact with the MySQL database for storing and retrieving data.

**8. Project Structure**

The project structure is organized as follows:

com.gym

│

├── entity

│ ├── Member.java

│ ├── MembershipPlan.java

│ ├── Payment.java

│ ├── Person.java

│ └── Trainer.java

│

├── util

│ └── HibernateUtil.java

│

├── StoreMain.java

├── ShowMain.java

└── RetrieveMain.java

**Description of Key Components:**

* **Entity Package**:
  + **Member**: Stores member details such as name, phone, email, and membership plan.
  + **MembershipPlan**: Defines various gym subscription plans (e.g., monthly, yearly).
  + **Payment**: Manages payment details for member subscriptions.
  + **Person**: Superclass for common attributes shared by members and trainers.
  + **Trainer**: Stores trainer information including specialty.
* **Util Package**:
  + **HibernateUtil**: A utility class responsible for managing Hibernate sessions and connections to the database.
* **Main Classes**:
  + **StoreMain**: Handles the insertion of new data (members, trainers, etc.) into the database.
  + **ShowMain**: Displays existing data from the database.
  + **RetrieveMain**: Retrieves specific data from the database, such as members or trainers by ID.

**9. Database Design**

**Tables:**

1. **Member Table**:
   * Stores details of gym members, linked to their membership plan and payments.
2. **Trainer Table**:
   * Stores information about trainers.
3. **Membership Plan Table**:
   * Stores available subscription plans.
4. **Payment Table**:
   * Tracks payments made by members.

**Example Table Structure:**

**Member Table**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| memberId | BIGINT | Primary Key (Auto) |
| name | VARCHAR(50) | Member's Name |
| phone | VARCHAR(15) | Member's Phone Number |
| email | VARCHAR(50) | Member's Email |
| planId | BIGINT | Foreign Key to MembershipPlan |

**Trainer Table**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| trainerId | BIGINT | Primary Key (Auto) |
| name | VARCHAR(50) | Trainer's Name |
| specialty | VARCHAR(50) | Area of Expertise |
|  |  |  |

**10. Inheritance Strategy**

The project uses **Joined Table Inheritance** for entities with common attributes. Specifically:

* **Person**: A superclass shared by Member and Trainer.
  + Member and Trainer tables inherit attributes such as name and email from Person.

**11. Command-Line Operations**

1. **Storing Data**:
   * Use the StoreMain class to insert new members, trainers, and payment details into the system.
2. **Showing Data**:
   * Use the ShowMain class to list all members, trainers, or plans in the system.
3. **Retrieving Data**:
   * Use the RetrieveMain class to search and display specific members or trainers by their ID.

**12. Conclusion**

This Gym Management System simplifies the management of members, trainers, and payments using a simple and effective command-line interface. It leverages Java and Hibernate for reliable data management, making it suitable for small to medium-sized gyms looking to automate their processes. This project also demonstrates core Java and Hibernate skills.