

Project Title:
“Online Gym Management System”

Guide Name:
Suramya Biswas

Team Member:
Prachi Chhatrola
Divya Solanki
Pooja Amanchi

Contents

1	Project Overview	
2	Features	
3	Modules And Functionalities	
4	Technologies	
	1.	Prerequisites
5	Setup Instructions	
6	Database Configuration	
	1.	Data Dictionary
7	Project Structure	
8	System Design	
	1.	ER Diagram
	2.	Class Diagram
9	Conclusion	
10	Future Enhancement	

1. Project Overview

The Gym Management System is a web-based application designed to manage the operations of a gym efficiently. Built using Java Spring Boot, MySQL, and JSP, the system provides an easy-to-use interface for gym administrators to manage members, classes, and schedules. This application streamlines administrative tasks, improves communication between staff and members, and ensures a smooth operation of the gym's day-to-day activities.

2. Features

This website provides Features like

- Member Management
- Slot scheduling and management
- Authentication and Authorization
- User Feedback

3. Modules And Functionalities

Modules:

1. Admin

Key Functionalities:

- Manage gym items
- Manage slots
- Manage booking
- Manage users
- Manage slot reports
- Review user feedback
- View profile

2. Member

Key Functionalities:

- View profile
- View and book slots
- Cancel bookings
- Provide feedback on services and facilities

4. Technologies

Key Technologies:

- **Java Spring Boot:** Provides a robust and flexible framework for building the backend of the application, including RESTful APIs and business logic.
- **MySQL:** A reliable relational database management system used to store and manage data related to members, trainers, classes, and other gym-related information.
- **JSP (Java Server Pages):** Used for creating dynamic web pages and presenting data to the users.
- **JS (Java Script):** A versatile scripting language used to create dynamic, interactive effects and functionality on web pages.
- **CSS (Cascading Style Sheets):** A stylesheet language used to control the presentation, layout, and design of web pages.

4.1 Prerequisites

- Eclipse IDE for Enterprise Java and Web Developers - 2024-03
- JDK Version 17
- Maven Version 2.7.16
- MySQL Version 8
- Java Servlet Version 4.0.1
- Tomcat Version 9.0.65
- Git

5. Setup Instructions

Clone the repository

```
git clone https://github.com/yourusername/gym-management-  
system.git  
cd gym-management-system
```

Build Project

```
mvn clean install
```

Run the Application

```
mvn spring-boot: run
```


6. Database Configuration

The project uses MySQL database, which is configured in

src/main/resources/application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3006/GymDB
spring.datasource.username=root
spring.datasource.password=password
spring.jpa.properties.hibernate.dialect =
org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=false
```

You can access the MySQL database using a client application such as MySQL Workbench or through the command line with the following settings:

- **JDBC URL:** jdbc:mysql://localhost:3006/GymDB
- **User Name:** root
- **Password:** password

6.1 Data Dictionary

➤ feedback

Field Name	Data Type	Relational Key
f_id	bigint	Primary Key
User_name	varchar (255)	Not Null
content	longtext	Not Null

Table 6.1.1 feedback

➤ gym_book

Field Name	Data Type	Relational Key
booking_id	bigint	Primary Key
item_id	bigint	Foreign Key
slot_id	bigint	Foreign Key
username	varchar (255)	Foreign Key

Table 6.1.2 gym_book

➤ gym_item

Field Name	Data Type	Relational Key
item_id	bigint	Primary Key
item_name	varchar (255)	Not Null
total_seat	int	Not Null

Table 6.1.3 gym_item

➤ gym_user

Field Name	Data Type	Relational Key
username	varchar (255)	Primary Key
email	varchar (255)	Not Null
first_name	varchar (255)	Not Null
last_name	varchar (255)	Not Null
password	varchar (255)	Not Null
type	varchar (255)	Not Null

Table 6.1.4 gym_user

➤ slot

Field Name	Data Type	Relational Key
slot_id	bigint	Primary Key
pricing	double	Not Null
slot_time	varchar (255)	Not Null

Table 6.1.5 slot

➤ slot_item

Field Name	Data Type	Relational Key
item_id	bigint	Primary Key
slot_id	bigint	Primary Key
seat_booked	int	Not Null

Table 6.1.6 slot_item

7. Project Structure

Here are the descriptions for each of the files:

com.GymInfo.gymManagementSystem.bean

- **Feedback.java:** Represents the feedback entity with user feedback details.
- **GymBook.java:** Manages booking information for gym slots.
- **GymItem.java:** Defines the items available in the gym.
- **GymUser.java:** Contains user information for gym members.
- **Item.java:** Represents the general items and their attributes in the gym.
- **Slot.java:** Defines time slots available for gym booking.
- **SlotItem.java:** Associates gym items with specific time slots.
- **SlotItemEmbed.java:** Provides a composite key for the SlotItem entity.

com.GymInfo.gymManagementSystem.config

- **EncoderConfig.java:** Configures password encoding for security.
- **SecurityConfig.java:** Manages security configurations, including authentication and authorization.

com.GymInfo.gymManagementSystem.controller

- **GymController.java:** Handles requests related to gym operations and bookings.
- **LoginController.java:** Manages user login and authentication processes.

com.GymInfo.gymManagementSystem.dao

- **FeedbackDao.java:** Interface for CRUD operations on feedback data.
- **FeedbackDaoImpl.java:** Implementation of FeedbackDao for interacting with the database.
- **FeedbackRepository.java:** Repository interface for feedback entity with Spring Data JPA.
- **GymBookDao.java:** Interface for CRUD operations on gym booking data.
- **GymBookDaoImpl.java:** Implementation of GymBookDao for interacting with the database.
- **GymBookRepository.java:** Repository interface for gym booking entity with Spring Data JPA.
- **GymItemDao.java:** Interface for CRUD operations on gym item data.
- **GymItemDaoImpl.java:** Implementation of GymItemDao for interacting with the database.
- **GymItemRepository.java:** Repository interface for gym item entity with Spring Data JPA.
- **GymUserRepository.java:** Repository interface for gym user entity with Spring Data JPA.

- **ItemDao.java:** Interface for CRUD operations on general item data.
- **ItemDaoImpl.java:** Implementation of ItemDao for interacting with the database.
- **ItemRepository.java:** Repository interface for item entity with Spring Data JPA.
- **SlotDao.java:** Interface for CRUD operations on slot data.
- **SlotDaoImpl.java:** Implementation of SlotDao for interacting with the database.
- **SlotRepository.java:** Repository interface for slot entity with Spring Data JPA.
- **SlotItemDao.java:** Interface for CRUD operations on slot-item association data.
- **SlotItemDaoImpl.java:** Implementation of SlotItemDao for interacting with the database.
- **SlotItemRepository.java:** Repository interface for slot-item entity with Spring Data JPA.

com.GymInfo.gymManagementSystem.exception

- **DeleteUserException.java:** Custom exception thrown when a user delete log-in user.
- **SeatNotAvailableException.java:** Custom exception thrown when a gym seat is not available for booking.
- **SlotAlreadyBookedException.java:** Custom exception thrown when attempting to book an already booked slot.
- **GlobalExceptionHandler.java:** Handles all exceptions globally and provides custom error responses.

com.GymInfo.gymManagementSystem.service

- **GymItemService.java:** Provides business logic and operations related to gym items.
- **GymUserService.java:** Manages business logic and operations related to gym users.

/gymManagementSystem/src/main/webapp/WEB-INF/jsp

- **about.jsp:** Provides information about the gym and its services.
- **bookedSlotReportPage.jsp:** Displays a report of all booked slots.
- **bookReportPage.jsp:** Shows a report of bookings made by users.
- **contactUs.jsp:** Contains a form for users to contact the gym.
- **emptySlotReportPage.jsp:** Displays a report of available slots.
- **feedbackPage.jsp:** Allows users to submit feedback.
- **feedbackReportPage.jsp:** Shows a report of all user feedback.
- **gymItemEntryPage.jsp:** Provides a form for adding new gym items.
- **gymItemReportPage.jsp:** Displays a report of all gym items.
- **header1.jsp:** Contains the header section for the website.
- **header2.jsp:** Contains an alternative header section for the website.

- **index.jsp**: The main landing page of the website.
- **index1.jsp**: The landing page (Admin).
- **index2.jsp**: The landing page (User).
- **loginErrorPage.jsp**: Displays error messages for login issues.
- **loginPage.jsp**: The login page for users.
- **newUserRegistration.jsp**: Registration form for new users.
- **seatErrorPage.jsp**: Shows error messages related to seat availability.
- **slotBookedPage1.jsp**: View all booked slots (Admin).
- **slotBookedPage2.jsp**: View all booked slots (User).
- **slotBookPage1.jsp**: Page for booking a slot (Admin).
- **slotBookPage2.jsp**: Page for booking a slot (User).
- **slotEntryPage.jsp**: Provides a form for adding new slots.
- **slotErrorPage.jsp**: Displays error messages related to slot booking.
- **slotReportPage1.jsp**: Shows a report of all slots (Admin).
- **slotReportPage2.jsp**: Shows a report of all slots (User).
- **userErrorPage.jsp**: Displays error messages related to user actions.
- **userReportPage.jsp**: Shows a report of all users.

8. System Design

8.1 ER Diagram

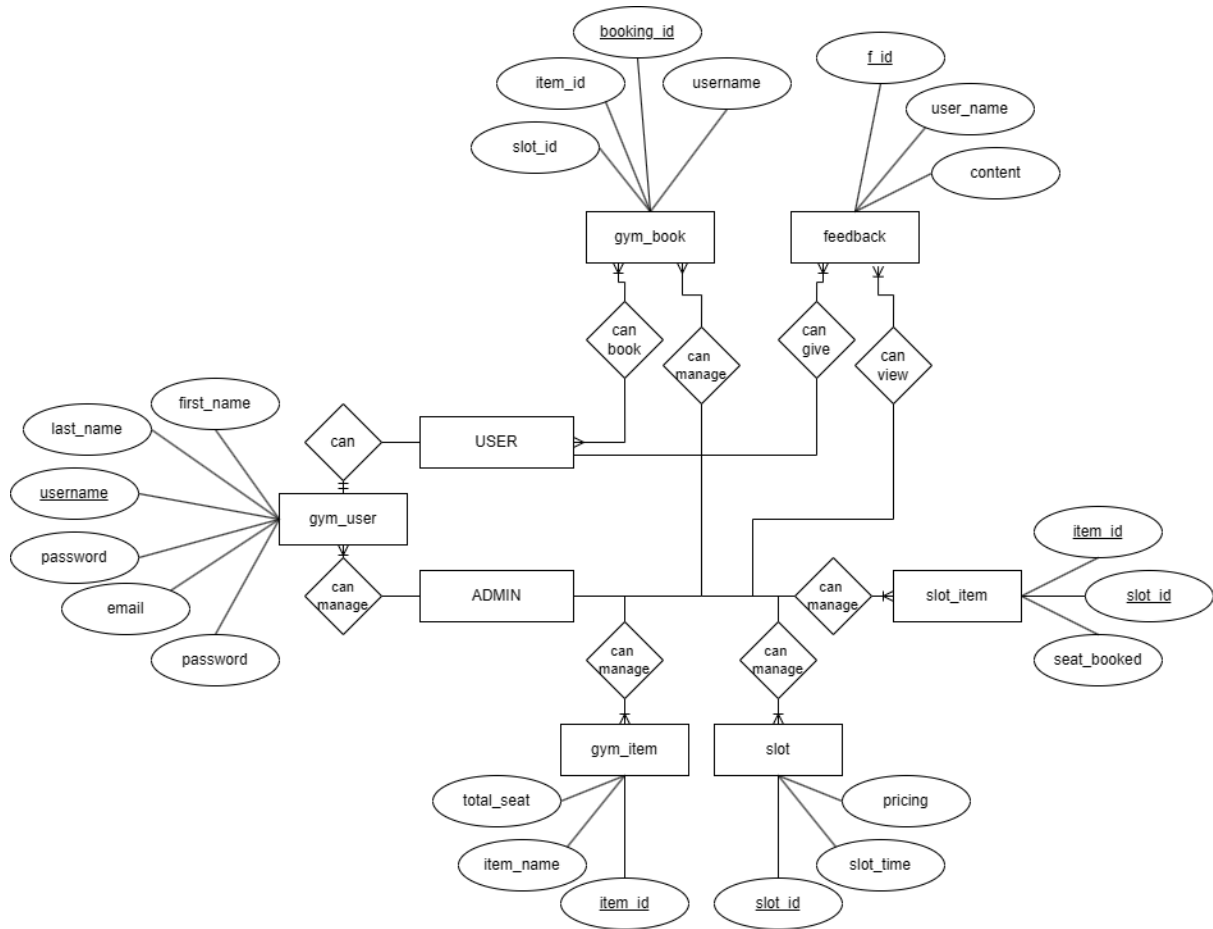


Figure 8.1 ER Diagram

8.2 Class Diagram

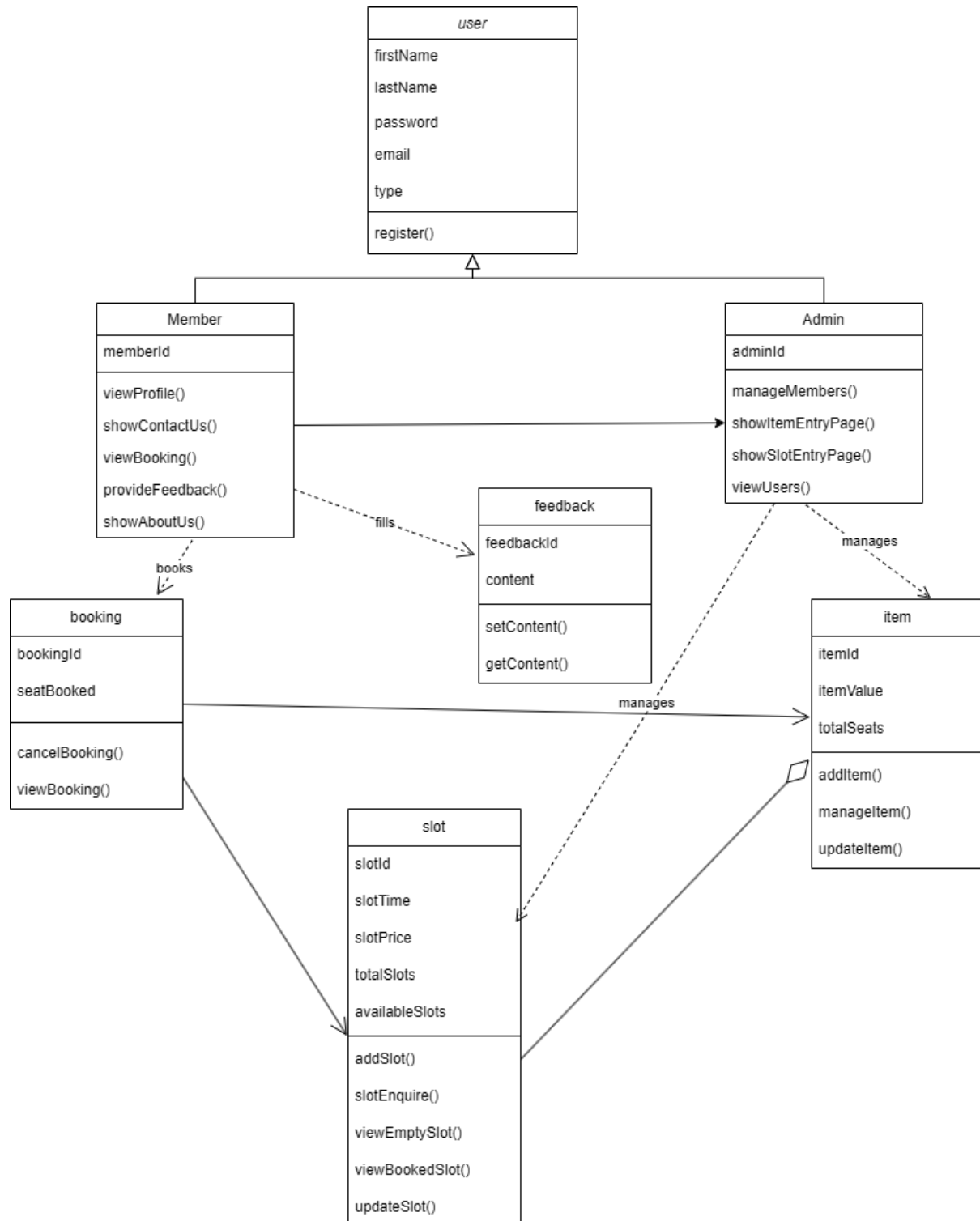


Figure 8.2 Class Diagram

9. Conclusion

The gym management system built using Spring Boot is designed to streamline and enhance the administrative and operational tasks associated with running a gym. This application features two primary modules: the Admin module and the Member module. Each module is tailored to meet the specific needs and roles within the gym's ecosystem.

In conclusion, the gym management system using Spring Boot is a comprehensive solution that addresses the key challenges faced by gym administrators and members. It enhances operational efficiency, improves member engagement, and provides a solid foundation for future growth and innovation.

10. Future Enhancement

Future Enhancement:

- Mobile Application Integration
- Advanced Analytics and Reporting
- AI-Powered Personalization
- Wearable Devices Integration
- Health and Fitness Tracking / Nutrition Tracking and Meal Planning
- Enhance Member Engagement