Abstract

The Hotel Management System is a comprehensive software solution designed to automate and streamline hotel operations. It encompasses functionalities such as hotel and room management, reservation handling, amenity tracking, payment processing, and review management. Built using Spring Boot and MySQL, the system provides RESTful APIs for seamless integration and scalability.

Objectives

The primary objective of the Hotel Management System is to enhance operational efficiency in hotel environments. It aims to simplify tasks such as booking, updating room availability, managing amenities, and tracking payments and reviews. The system also supports robust data handling and ensures consistency through a relational database schema.

Technologies Used

The project utilizes the following technologies:

- Java 21
- Spring Boot Framework
- MySQL Database
- JPA (Java Persistence API)
- Swagger for API documentation
- Maven for dependency management
- Lombok for boilerplate code reduction

System Architecture

The system follows a layered architecture comprising:

- Controller Layer: Handles HTTP requests and responses.
- Service Layer: Contains business logic and service methods.
- Repository Layer: Interfaces with the database using JPA.
- Model Layer: Defines entity classes mapped to database tables.
- Exception Handling: Global exception handler for consistent error responses.

Database Design

The database is designed using a relational schema with the following key tables:
- Hotel
- RoomType
- Room
- Reservation
- Amenity
- HotelAmenity (Many-to-Many relationship)
- RoomAmenity (Many-to-Many relationship)
- Payment
- Review

Each table includes appropriate foreign key constraints to maintain referential integrity.

API Endpoints

The system exposes RESTful APIs for all major operations. Examples include:
- POST /api/hotels/post: Add a new hotel
- GET /api/hotels/all: Retrieve all hotels
- PUT /api/hotels/update/{hotelId}: Update hotel details
- DELETE /api/hotels/delete/{hotelId}: Delete a hotel
- GET /api/rooms/available/{roomTypeId}: Get available rooms by type
- POST /api/reservation/post: Create a reservation
- GET /api/payment/total-revenue: Get total revenue
- POST /api/review/post: Add a review

Implementation Details

The project is implemented using Spring Boot with JPA for ORM. Entities are annotated with @Entity and mapped to database tables. Controllers handle HTTP requests and delegate to services, which perform business logic and interact with repositories. Swagger is configured for API documentation, and exception handling is centralized using @RestControllerAdvice.

Challenges Faced

Some of the challenges faced during development included:
- Designing a scalable and normalized database schema.

- Managing many-to-many relationships between hotels/rooms and amenities.
- Implementing robust exception handling for all endpoints.
- Ensuring data consistency and validation across layers.
- Integrating Swagger for API documentation.

Final Outcome as a Learner

As a learner, this project provided valuable experience in full-stack development using Spring Boot. It enhanced understanding of RESTful API design, relational database modeling, and enterprise application architecture. The hands-on implementation of a real-world system improved problem-solving and debugging skills.

Conclusion

The Hotel Management System successfully demonstrates the application of modern software engineering practices to solve real-world problems. Its modular design, robust API structure, and comprehensive feature set make it a scalable and maintainable solution for hotel operations. This project serves as a strong foundation for future enhancements such as user authentication, reporting dashboards, and mobile integration.