Faculty of Science

Course: CSCI 3240 Web Application Development

Instructor: Dr. Rohollah Moosavi

Hotel Management System Project

You are required to submit the .zip file of the project on Canvas. **The deadline for submission is April 4, 2024**. All team members must individually submit the file on Canvas. Additionally, you need to **present it to the instructor in person on April 4, 2024**, as per a later schedule.

If you have any questions regarding the project, please contact Hafiz Arslan during his office hours (Mondays, 3:45pm to 4:45 pm, on https://meet.google.com/igm-ibsb-gqf).

1. Project Introduction

Welcome to the Hotel Management System project. This project is designed to introduce you to the fundamentals of web application development using **Spring Boot**, **HTML**,**CSS**, **Object-Oriented Programming** and **Database**. You will develop a simple yet comprehensive system to manage hotel operations, including handling rooms, bookings, customers, and services provided.

1.1 Project General Requirements

- IDE: STS.
- Build Tool: Maven.
- Backend Framework: Spring Boot.
- Database: H2.
- ORM: Spring Data JPA.
- Frontend: Thymeleaf, HTML, CSS.
- Java Version: 8 or above.

1.2 Project Specification

Your application must include the following components:

Models

- **Customer**: To store customer details.
- **Room**: To manage room details.
- **Booking**: To keep track of room bookings.
- **ProvidedService**: To manage services offered by the hotel.

Repositories

- Interfaces for each model to extend `JpaRepository`.

Services

- Service classes for business logic related to each model.

Controllers

- Controllers to handle HTTP requests for each model.

Templates

- Thymeleaf templates for displaying and managing each model.

Tables

Your H2 database should include tables corresponding to each model. The tables are:

- `CUSTOMER`
- `ROOM`
- `BOOKING`
- `PROVIDEDSERVICE`

In Figure 1, you would see the diagram and the relationships. A brief description of these relationships is a follows

A Customer can have multiple Bookings: This relationship indicates that a single customer can book multiple rooms at different times.

A Room can be associated with multiple Bookings: This shows that a particular room can be booked multiple times (by the same or different customers) across different dates.

A Booking can include multiple ProvidedServices: This relationship captures the idea that a booking (which is essentially a stay at the hotel) can include multiple services like room cleaning, food services, etc.

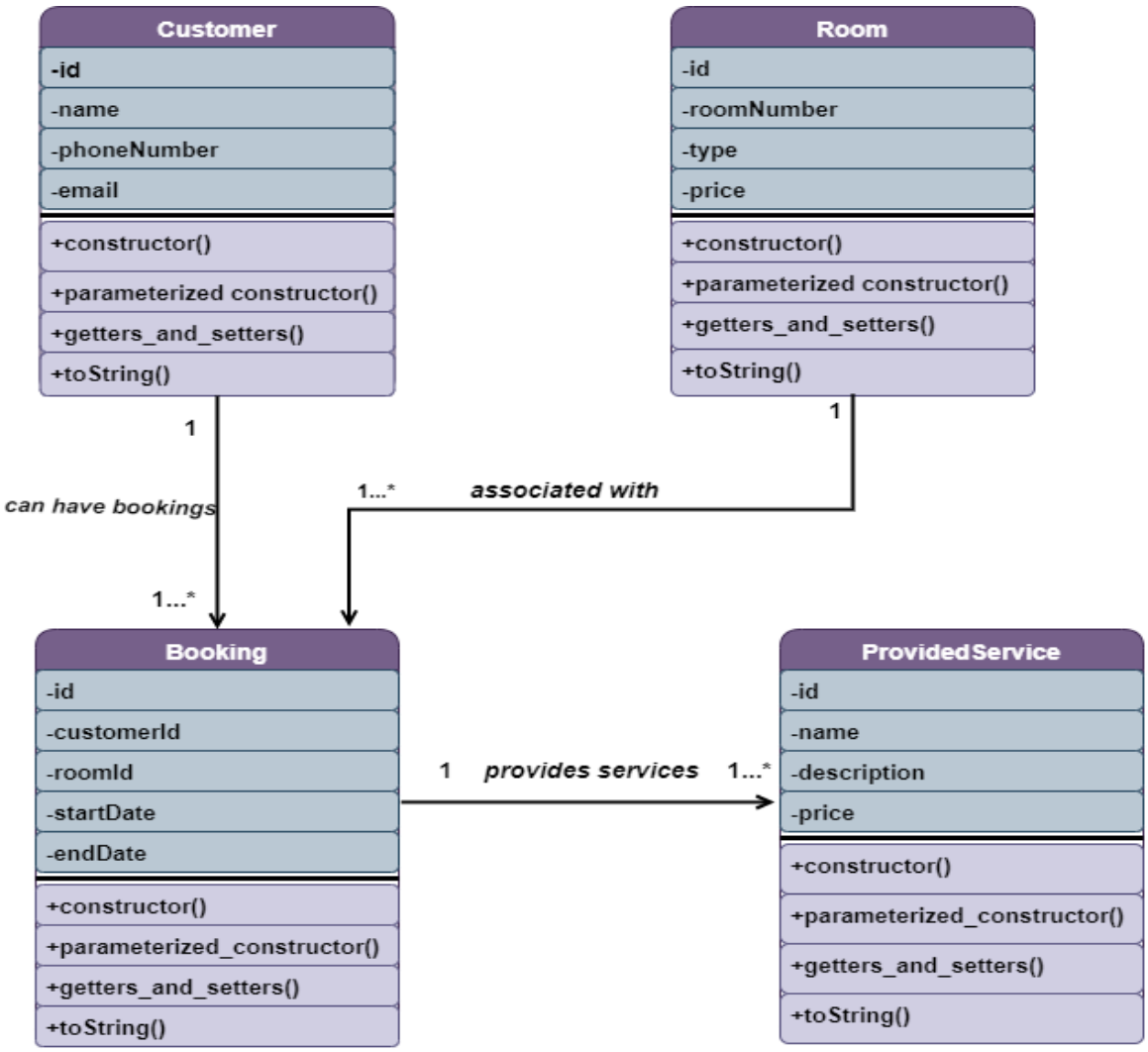

Figure 1Hotel Management Diagram

1.3 Project Submission

Submit your project on Canvas as a ZIP file containing your entire Spring Boot project. Ensure your project is well-organized and follows the specified guidelines.

2. General Guidelines while Developing the Backend Application

Packages and Files

Create the following packages and classes within your Spring Boot application:

- `**model**`: Contains your entity classes.

- `Customer.java`
- `Room.java`
- `Booking.java`
- `ProvidedService.java`

- `**repository**`: Contains your Spring Data JPA repositories.

- `CustomerRepository.java`
- `RoomRepository.java`
- `BookingRepository.java`
- `ServiceRepository.java`

- `**service**`: Contains service classes for business logic.

- `CustomerService.java`
- `RoomService.java`
- `BookingService.java`
- `ProvidedServiceService.java`

- `**controller**`: Contains web controllers to handle HTTP requests.

- `CustomerController.java`
- `RoomController.java`
- `BookingController.java`
- `ProvidedServiceController.java`

Role of Each File

- Model classes represent the entities in your application and correspond to tables in your database.
- Repository interfaces facilitate data access and manipulation.
- Service classes contain business logic and interact with repositories.
- Controller classes handle HTTP requests and responses, interacting with services and views.

3. General Guidelines while Developing the Frontend

Templates

- Create Thymeleaf templates in the `src/main/resources/templates` directory:
- `customers.html`: For listing and adding customers.
- `rooms.html`: For listing and adding rooms.
- `bookings.html`: For listing and adding bookings.
- `services.html`: For listing and adding services.

Styling

Include a `style.css` file in the `src/main/resources/static` directory to style your templates. Ensure your HTML templates link to this CSS file for consistent styling across your application.

Navigation

Each template should include navigation links to the other pages, ensuring a seamless user experience. Use Thymeleaf's `th:href` attribute to create these links.

4. Project Rubric

Total Marks Available: 34%. Below are the structured criteria for grading rubric to comprehensively evaluate your work on the Hotel Management System project. This project is a significant part of your coursework, designed to assess your proficiency in both backend and frontend web development within the context of a Spring Boot application. The rubric below outlines the criteria against which your submission will be assessed.

4.1. Backend Development (Total Marks = 18.8%)

- **Model Classes (5%):** Each entity (`Customer`, `Room`, `Booking`, `ProvidedService`) you implement correctly will contribute towards this grade. It is crucial that these models accurately represent the project's domain and are properly annotated for JPA. (1.25% per entity)
- **Repository Interfaces (4%):** Demonstrating your understanding of Spring Data JPA, each repository interface should extend `JpaRepository` and be correctly parameterized for its entity and key types. (1% per repository)
- **Service Classes (4%):** Services are the backbone of your business logic. Proper annotation, adherence to the Single Responsibility Principle, and correct utilization of repositories are key metrics here. (1% per service class)
- **Controller Classes (5.8%):** Your controllers should be well-structured, utilizing the `@Controller`, `@GetMapping`, and `@PostMapping` annotations effectively. Each controller will be evaluated on its ability to handle model attributes for forms, manage data display, and correctly implement navigation logic. (1.45% per controller)

4.2. Frontend Development with HTML and CSS (Total Marks = 10.2%)

- **HTML Templates (6%):** The structure and functionality of your Thymeleaf templates (`customers.html`, `rooms.html`, `bookings.html`, `services.html`) are essential. Correct use of Thymeleaf syntax for dynamic content rendering and form handling is expected. (1.5% per template)
- **Navigation and Usability (2%):** A seamless and intuitive navigation experience across your application is crucial. This includes logical flow and user-friendly interfaces. (2%)

- **Styling with CSS (2.2%):** Your application should not only function well but also look professional and be accessible across different devices. This will be judged on the consistency, clarity of your design, and responsiveness. (2.2%)

4.3 Code Quality and Organization (5%)

- **Readability (2.5%):** Your code should be well-formatted, use clear naming conventions, and include comments where necessary to explain complex logic.
- **Structure (2.5%):** The logical organization of your project into packages, classes, and files will be assessed. This organization should reflect best practices and facilitate easy navigation and understanding of your project structure.

4.4 Submission Requirements

You are required to submit your project as a `.zip` file through Canvas. This submission must include all source code for both backend and frontend components, along with any additional resources utilized. Please ensure your submission includes a `README.md` file detailing instructions on how to run your project and any other relevant documentation.

5. Final Notes

Partial marks may be awarded for each category based on the completeness and correctness of your implementation. I encourage you to thoroughly test your application to ensure all functionalities work as expected. This project is an opportunity to demonstrate your skills and understanding of web development principles. I look forward to reviewing your innovative solutions and understanding of the course material through this project.

Supplementary Files: Please see supplementary materials for previous labs.