Faculty of Engineering and Technology
Computer Science Department
COMP4382-SP.TOP: WEB SERVICES TECHNOLOGY

18-May-2024

Project #2

Hotel Management System

Summary

The Hotel Management System is designed to streamline operations for both hotel employees and guests. It offers functionalities for the customers, i.e., searching the available rooms, reservations, customer check-ins and check-outs, and invoice generation. Employees can manage room availability, maintain customer profiles, and oversee housekeeping schedules. You need to design and implement a set of RESTful APIs based on Spring Boot to enable the development of the required system frontend (web and mobile interfaces). Students need to work in groups of two only. In this project, you should implement the following:

1. Backend implementation that covers the basic implementation of the following details.
    a. Customer management functionality in order to enable users to register, login, and manage user profiles (view, update, and change passwords).
    b. Employee management functionality, in order to enable the admin to manage the hotel employees and staff.
    c. Search functionality, enabling searching for reservations by customer name or ID and date, search customer info, and available room, with showing its details, price, facilities, capacity, size, and features. These functionalities are available for user roles (admin, customer), but with different details.
    d. Reservation functionality, in order to enable the customer to book room(s), modify, and cancel a reservation. reservation cancellation request needs admin approval.
    e. Room management functionality includes managing room types, availability, and status by the admin users.
    f. Check-In/Check-Out process to manage the customer arrival and departure processes by the admin users.
    g. Housekeeping management functionality, such as scheduling and tracking housekeeping tasks and employees.
    h. Billing functionality, to generate and manage invoices for customer reservations.
    i. Also, consider the role-based access control, where different functionalities are available based on user roles (admin, customer).
    j. You can use part of the database design mentioned in the following tutorial, where applicable, in order to simplify your work: https://www.youtube.com/watch?v=pYYmvaKy-yQ.

2. Create a Docker image for your application and push it into your DockerHub account. During the project discussion, you will be asked to run the application with its database using docker-compose.
3. Push your project on GitHub under the repository “project2-hotel-management-system”, make the repository private, and add ‘mohkharma’ user as a collaborator. Ensure the proper use of the Git commands when pushing the project content into the repository and file structure (no manual upload, archived source code upload).
4. Put into the repository readme.md file the following:
    a. Clear and detailed information about the project, a description of each resource, and student names.
    b. ER diagram with the use of proper notations.
    c. How to build, package, and run the application to generate JAR or Docker image. With a link to the Docker image on DockerHub (make the repository on DockerHub public).
    d. The last section in the readme file will be a section to list what you have learned from this project.
5. Document your code in detail. Use OAS 3.1.0 to document the APIs. You need to follow the Code First approach.
6. Implement the entity relations in the DAO layer.
7. Follow all the best practices and constraints that you have learned during the course.
8. Secure the APIs using JWT. Some of the APIs will be publicly accessible, and some of them will require authenticated/authorized users. Only two types of user roles are required: customer and admin. Your implementation should include signup and authenticate APIs.
9. You have to showcase API versioning usage in at least 3 APIs, each of which uses a different versioning approach.
10. Based on the generated API documentation based on OAS standards from the code, prepare a Postman collection that includes a complete testing scenario to simulate the user journey on the system interface. Then export the Postman collection and push it into the GitHub repository.
11. The application needs to initialize the database structure, populate the required initial data, and generate samples automatically, so you can enable the use of the APIs without any issues during the discussion.
12. Data validation and proper exception handling need to be implemented.

Evaluation criteria:

1. Coverage of the above requirements.
2. Implementing the best practices and constraints.
3. Overall design and understanding.

Submission:

1. Due date: June 13, 2024, 00:00. You have to reply to the assignment message on Ritaj with your GitHub repository link only. Then download the project from GitHub (.zip format) and upload it on ITC.
2. No late submissions or extensions will be accepted.
3. You might be asked to design and apply code changes during the discussion.

Good luck