**Report For Enterprise Middleware Coursework**

**Project Documentation**

**Overview**

The project implements a Travel Agent service, which consist of three commodities. The system facilitates the creation of entities for customers, hotels, booking and finally by calling other commodities from colleague’s booking services. Using object-oriented programming concepts like interface-based design and immutability, the system ensures flexibility and maintainability.

**Key System Features:**

* **Entity classes**: It created relational database.
* **Repository classes**: This class is to manage entities.
* **Rest Services class**: It is to define API and perform CRUD operations in swagger.
* **Validator**: To validate the services.

**Evaluation of my Achievement**

**Task 1**

In this step, customer details, hotel details and booking details has been created. Then, Restful service is created to perform Get, Put, Post and Delete operation in swagger. Then test case for rest service is added.

**Task 2**

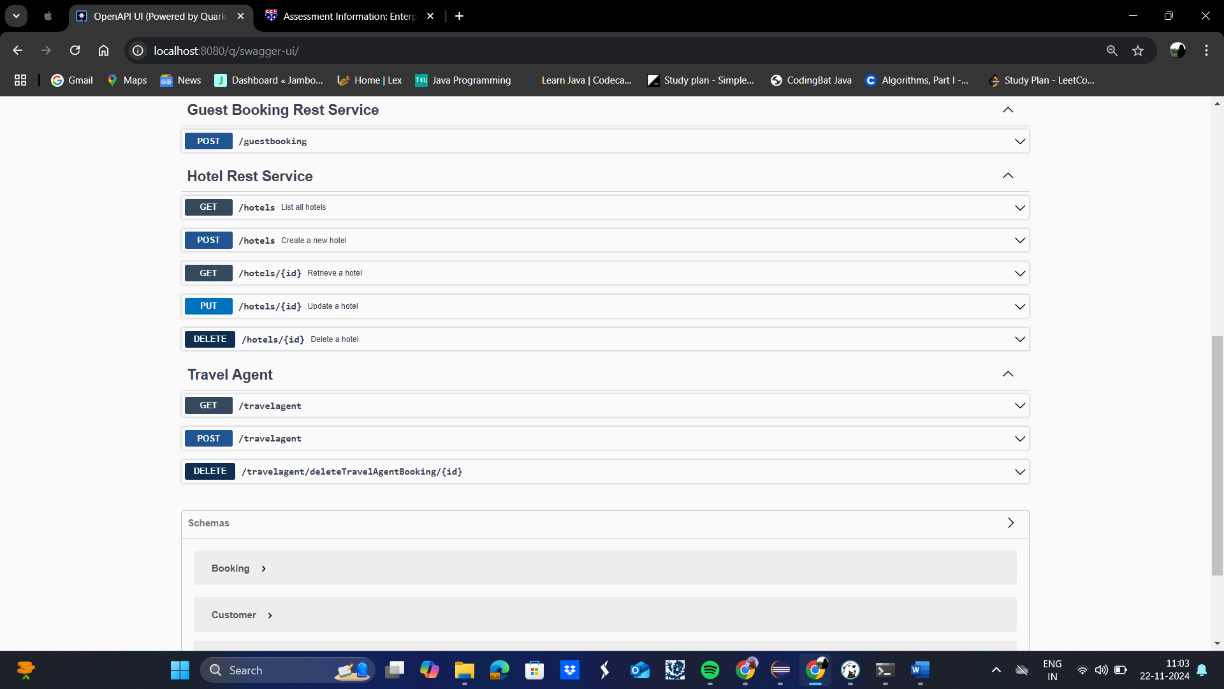
In this task, three main advancement is performed. First, deletion of customer and hotel is performed, then secondly, cascading is done, i.e., if customer or hotel is deleted then then bookings related to them is also deleted. I used one to many and many to one annotation. Finally, guest booking service is created, where customer and bookings ca be done in one transaction. And testing added.

**Task 3**

In this task, I connected other two swagger in my code using application.property to create a travel agency. After calling the swagger, created an object class for travel agent and restful service class. Then added a test class for travel agent.

**Screenshots of all outputs**

1. **Swagger page:A computer screen shot of a computer

   Description automatically generated**
2. **Terminal:A screenshot of a computer screen

   Description automatically generatedA screenshot of a computer

   Description automatically generatedA screenshot of a computer

   Description automatically generated**
3. **DBeaver:** A computer screen shot of a computer

   Description automatically generated

**Unit Tests and Testing**

Unit tests are implemented to ensure the system functions correctly in various scenarios, including normal operations, edge cases, and exceptional conditions.

**Core Components Tested:**

* **Customer testing**: Testing the crud operation for customer details.
* **Hotel and Booking**: Testing the crud operation for hotel and booking details.
* **Guest Booking**: Tests the single transaction booking.
* **Travel agent**: Test for retrieving all booking details of customer.

**Conclusion**

The system meets the project specifications, offering flexible Travel agent system. The system is designed for extensibility, allowing for future enhancements. The inclusion of thorough unit testing ensures that the system is reliable and robust.