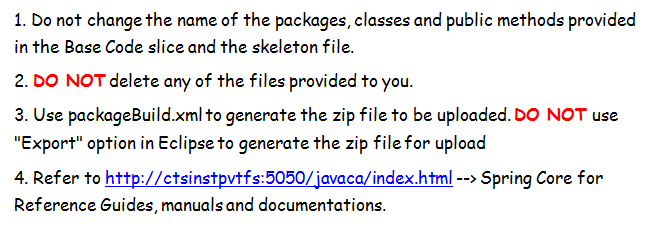
**Instructions:**



Movie World is a large movie theatre network. Recently they have decided to start online ticket reservation. They approached you to develop a system which they can use for the online reservation purpose. The system should contain the below services.

**Service1:** Accepts the movie code, ticket class, number of seats, date, show time from the customer and reserves the seats for the customer.

**Service 2:** <Explanation of Service 2 will be present here>

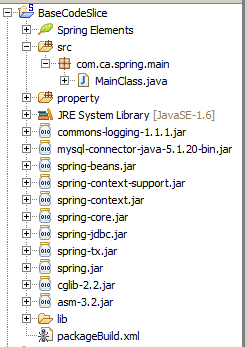
**Service 3:** <Explanation of Service 3 will be present here>

**Skeleton Code for Development:**

Use the below skeleton code for development.Use the "User Manual for SDE" document available in the URL provided in the ***Instructions*** Section for importing the base code slice and skeleton code.

<BaseCodeSlice.zip will be provided for development>

The Base code slice will have the following structure.



<CaseStudySkeleton.zip will be provided. Sample is below>



After the coding is completed, the zip file to be uploaded should be created as per the instructions in the "User Manual for SDE" document.

**DDL/DML for creating tables and inserting data:**

Please run the below.sql file before you start the case study. Refer to the “User Manual for Mysql” in the URL provided in the ***Instructions*** Section.

<Casestudy.sql file will be provided with required DDL and Master Data DMLs>

**Overview of Service 1:**

This service is used by the system to reserve seats for a customer. This service is going to become a part of different other modules in the system so have been asked to develop the Service as an independent module so that it can be plugged into other modules easily.

**Business Rules& Validations:**

This service accepts the movie code, ticket class, number of seats, date, and show time from the customer and reserves seats for him. This service needs to return the total charge payable by the customer.

The ticket charges for each ticket class should be stored in external property file. The various ticket classes available are : ECONOMY, DELUXE, BALCONY

*Total ticket charge = Number of seats reserved \* Ticket Charge for the class (From property file)*

The reservation details needs to be saved in the **RESERVATION\_DETAILS** table and the number of seats available needs to be updated in the **TICKET\_AVAILABILITY** table.

**Note:** The population of reservation details into **RESERVATION\_DETAILS** tableand updating the seat availability in **TICKET\_AVAILABILITY table** should happen in a single transaction. In case any of the operations fails the transaction should be rolled back automatically.

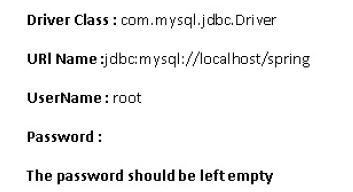
**Technical Specifications:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Method Name** | **Input** | **Output** | **Exception** |
| MovieWorldTicketManager | reserveSeats | String movieCode,String ticketClass,String numberOfSeats,String date,String time | int ticketCharge | MovieWorldException  This Exception to be caught and thrown back to Main class |
| MovieWorldTicketFacade | reserveSeats | String movieCode,String ticketClass,String numberOfSeats,String date,String time | int ticketCharge | MovieWorldException  This Exception to be caught and thrown back to MovieWorldTicketManager class |
| MovieWorldTicketBO | reserveSeats | TicketVO ticketVO | int ticketCharge | MovieWorldException  This Exception to be caught and thrown back to MovieWorldTicketFacade class |
| MovieWorldTicketDAO | saveReservationDetails | TicketVO ticketVO | void | MovieWorldException  This Exception to be caught and thrown back to MovieWorldTicketBO class |
| updateAvailableSeats | TicketVO ticketVO | void | MovieWorldException  This Exception to be caught and thrown back to MovieWorldTicketBO class |
| service1.xml | Contains all the xml configurations related to Service 1 |  |  |  |

**Limitations and Constraints**

1. The classes MovieWorldTicketManager ,MovieWorldTicketFacade and MovieWorldTicketBO should be defined as spring beans in spring configuration xml.
2. All the beans should be wired using Constructor based dependency injection.
3. Use only Spring APIs for loading the ticket class types from property files(Do not use java ResourceBundle Api)
4. The datasource needed for JDBC operation needs to be configured in spring xml file and needs to be injected to the appropriate DAOs.
5. Use Spring JDBC API for the DML operations (Should not use java JDBC API). The SQL connection details are given in the previous service.
6. Should use Spring Transaction Management APIs only for managing transactions. The transaction should be managed in the MovieWorldTicketBO
7. Use ONLY service1.xml for all configurations related to service 1

**SQL Connection Details:**



**Sequence Diagram**

**<Sequence Diagram will be provided here>**

**Other services**

This is just a sample case study for reference. Your actual case study for assessment will have more number of services to be implemented. Each of the services needs to be configured in the respective spring configuration files mentioned in the service (service1.xml, service2.xml etc).