CS 454/SCS 491: Software Engineering for Distributed Systems

Assignment 2: A functioning online home-made dishes operation platform

Objective:

The objective of this assignment is to familiarize ourselves with the concepts provided by **EJBs, RabbitMQ, and microservices**.

Requirements:

Functional Requirements:

Admins in the system should have the following abilities:

- Creation of dishes seller companies representative accounts.

  a. Given a range of company unique names

  b. Password for each company is auto generated

- Listing of customer accounts
- Listing of dishes preparation companies representative accounts.

As a dishes' seller representative, you should be able to:

1. Login into the system using the generated credentials as sent by the admin
2. View your currently offered dishes for sale.
3. View previously sold dishes, including information about the customers who bought each product and the shipping company.
4. Add new dishes and their corresponding amounts and prices.
5. Update existing dishes info.

As a customer, you should be able to

1. Register as a new customer through the system.
2. Login into the system using the credentials used during registration.
3. View current and past dishes orders.
4. Make new orders, where an order would consist of several dishes with varied amounts.
5. Orders processing should be confirmed back to customers.

Use RabbitMQ to apply the following requirements:

- When an order is issued by a customer, the system should confirm whether there is enough stock of that order.

- If an order is confirmed to be available in stock, the order should be completed, and the payment for that order should proceed. A minimum charge should apply when proceeding with processing orders. If the order is confirmed, the order should be confirmed back to the customer.
- If the minimum charge is not met, orders should be rejected, and cancelled automatically. All previous actions should be rolled back.

Additional features for teams of 3

- As an admin, you be able to:
    a. Create Shipping companies. Companies should have geographic coverage including specific regions (i.e., a customer cannot request shipping from a company that doesn't cover his geographic location).
    b. See a listing of shipping companies
- As a customer, you should be able to select a shipping location, and while making an order, your shipping should be confirmed.
- When an order is made, the shipping availability should be confirmed before proceeding with the payment. The shipping fees should be incorporated in the total order charge.
- In case of order rejection, all shipping actions should be rolled back.
- Using RabbitMQ, customers should be notified of any updates regarding their order (in case of shipping confirmation, payment confirmation, or order cancellation).

Bonus [2 marks]:

- Admins should be notified in case of payment failure In case of only "PaymentFailed" events on payments exchange (Hint: Requires use of Rabbit MQ direct exchange feature)
- Add a "log" exchange that all services use to log events with different severities (Info, Warning, Error) named as the following "ServiceName_Severity" ie. Inventory_Error, Order_Info. Admins should be notified **only** of error logs. (Hint: Requires use of RabbitMQ topics)

Technical Requirements

1. **Using Microservices:**
    - The above system should be designed to follow the microservice architectural style, and to include at least 3 services, while supporting the same functional requirements.
    - Each service should be implemented as its own project. This means that it has its own codebase and its own DB. If you have S1 Service and S2 Service then S1 shouldn't be

able to get any information from the DB of S2, but instead should request it from the S2 Service through REST calls.

- Your submission should have:
  - At least 3 complete services.
  - One of the services should be developed using EJBs while applying two EJB types (check **point 2** below for more details).
  - The remaining services can be developed in Java only, OR you can choose to develop only one of the services in a different programming language (or using a different framework like Spring).

2. **Using EJBs:**
   - You are required to use any two of these 4 different bean types to fulfill some of the above functional requirements.
     - Stateless
     - Stateful
     - Singleton
     - Message Driven
3. **General considerations:**
   - Your submission should have a functioning UI. It can be a separate UI that calls the APIs exposed by your services.
   - Your interface should be a web-based interface using any technology of your choice to simulate a functioning online learning platform with different users as per the above-mentioned functional requirements (i.e., we should be able to perform all the functionalities using such web-based interface).
   - Your service should be exposed as REST APIs, and you should expose your beans using REST to fulfill the web service REST API as appropriate.

Deliverables:

The source code for your developed application.

Rules of Submission:

1. No email submissions will be accepted.
2. There are no late submissions.
3. The assignment is in groups of 2 ~3 with the same TA. In case you're forming a group of 3, then you **MUST** implement the additional features.
4. If more than 3 team members submit the assignment, all team members will get zero.
5. Cheating is not tolerated and will be given negative grades
6. You should submit your assignment as **ONE zip file** with the below naming convention: **DS_Assign2_GroupNumber_ID1_ID2** (example: DS_Assign2_S1_20116001_20116002)

7. You should submit your **source code along with a <u>document</u>** explaining any decisions or assumptions that you made. This document should also include any steps needed so that the TA can properly run your code. It is your duty to write down the exact steps needed to run the source code properly.
8. You SHOULD NOT copy any code from the internet or from your colleagues. It will be detected and considered as a cheating case.
9. Submission of the assignment will be on Google Classroom through a Google form link that will be shared later.
10. Deadline for the submission is **<u>Thursday 15th of May 2025.</u>** A separate submission link will be shared through the Google classroom.