Cairo University
Faculty of Computers and Artificial Intelligence

Assignment Description 2023/2024

Version 1.0

Introduction

- In this assignment you will design and implement a non-trivial software system. You will practice the concepts you learned during the course.
- In each phase you are required to deliver the following deliverables **through Google classroom as one zip file**, named with your student IDs as follows: **Assignment2_SE_StudentID1_StudentID2_StudentID3_StudentID4**. Following such naming is a MUST. If you don't follow that naming convention, your submission will NOT be graded.
- The project must:
    1. Be developed in the Java Programming Language.
    2. Use IntelliJ IDE for Java EE (not SE) or VS Code with springboot https://code.visualstudio.com/docs/java/java-spring-boot.
    3. Use Spring boot (you can use spring initializer https://start.spring.io/)
    4. Application Server: Tomcat
    5. Postman for testing the API.
    6. For the data storage, you **must use in-memory database** (built in technology or you build it using lists). Hence, the data storage must not require any additional setup.
    7. Run properly on the TA's machine. One way that can help you confirm that your project would run properly on the TA's machine, is to take all your submission from one machine, and try to run it on a completely different machine using the provided IntelliJ version and Java version. **It is your responsibility to make sure that your submission will work properly on the TA's machine.**
- The deliverables are:
    1. An SDS (Software Design Document) using the attached template, where such document would contain the following items:
        - **One** class diagram
        - Sequence diagrams for the **most complex** scenarios. The submitted sequence diagrams should be **1 x the size of the team**, where each team member would be responsible **for submitting one sequence diagram**.
        - Link for Git repository for the developed source code project. Your Git repository should be private and it is your responsibility to add your TA to your project Git repository to be able to assess your proper usage of Git within your project development.
        - A table that clearly explains the exact mapping between every single requirement and its corresponding web service API operation. If you do not provide such table with clear details for **every single requirement**, the TA will NOT be able to mark your project. The TA needs to know the exact web service calls to test the project.
        - A Postman collection for the designed webservice.

2. Zipped copy of the source code project. Note that your submitted source code project should work properly on any TA's machine that has IntelliJ IDE version ideaIU-2023.2.4 and Java JDK version 11.0.16.1.

- **All deliverables MUST be included in your submitted zip file through Google classroom as per the google form to be posted.**

Assignment Logistics

1. Students will be divided into groups; each group consists of 3-4 members. Groups can be from the same lab or from different labs.
2. If a team is formed out of more than 4 members, their submission will be rejected, and they would get a zero for the phase.
3. Questions about the project should only be asked through the following form link to ensure consistent answers to all students: https://docs.google.com/forms/d/e/1FAIpQLSe0zKu3rf1mbYdSzaxbint-FWI-lmAGA3xettHuUOx5wWTfDw/viewform
4. Academic honesty is assumed. All work submitted must be original and written by your team (Not copied from students, the net, outside sources). Plagiarism will be penalized.
    - Soon, you will be our colleague and we will be proud of you.
    - Professional conduct and practice are essential in your career.

Project Phases:

Phase	Deliverables	Deadline	Mark
Assignment submission on **the course's Google classroom**	**Design and implement the assignment requirements (mentioned below)** Submit all the required deliverables (mentioned above)	Dec. 31st, 2023 • Late submission is not allowed • No email submissions will be accepted.	4 marks + 2 bonus marks = 6 marks

Assignment overview

The assignment will NOT be an entire software system, but rather a component within the system that fulfills a need, provides a feature, that serves a greater PURPOSE within the WHOLE purpose of the software. And typically, we developers, do not implement the whole system as a whole, but we (in teams) get assigned to parts of the system, and these parts when collectively work together, make up the system/software.

So, we are going to simulate this real situation in this assignment scope, we (as teams) will not develop a whole software, but rather a module within this software. This module would be the "Orders and Notifications Management" module. Such module should allow making online purchase for orders, as well as managing message notifications based on various actions taken during the various orders' purchase relevant operations.

Requirements

The "Orders and Notifications Management" module should provide the following features:

1. A list of all the products currently available for purchase should be displayed. Products are defined by a serial number, a name, a vendor, a category, and a price. Furthermore, a count of the remaining parts from each category should be available.

2. A customer should be able to create an account and put a specific balance using that account. Such balance would be utilized during future purchasing operations.

3. A customer can place a simple order, where such an order would include a single product or several products. Furthermore, a customer can make a compound order, where that order can include various orders headed to near-by locations, in addition to his own products, to lessen the shipping fees. For instance, a customer can make a compound order for himself and two of his friends, who all live in different locations in El-Dokki.

4. You should be able to list all the details of both simple and compound orders.

5. To place a compound order, the customer needs to pick his products, and his friends' orders as well. Once the order is placed, the balance of each order is deducted from its corresponding customer.

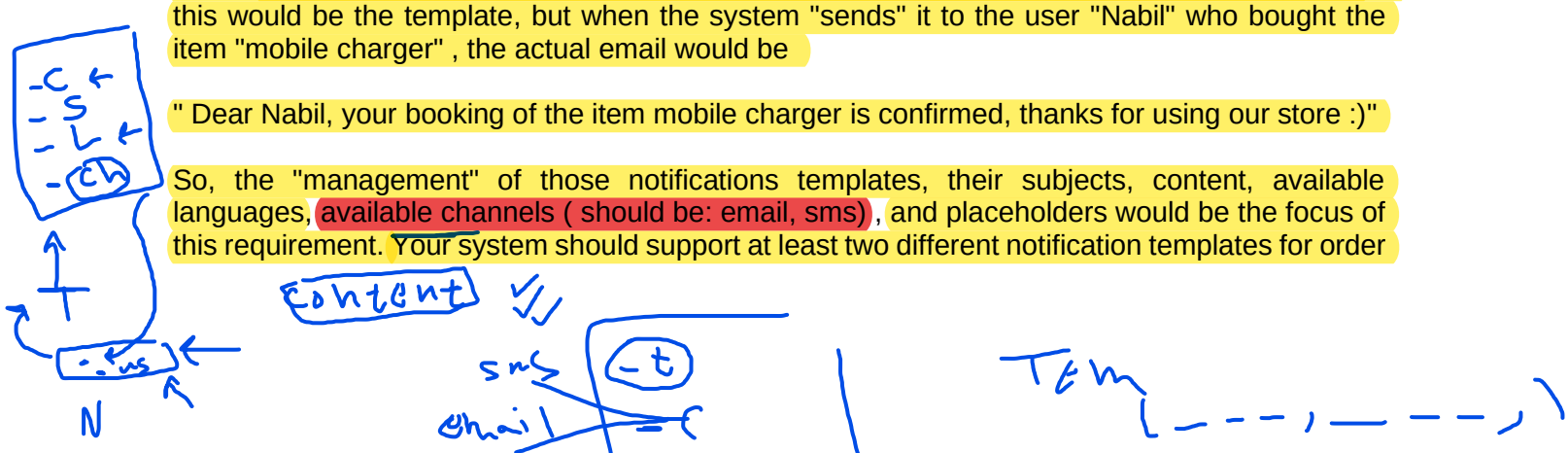
6. After placing the order, the user can ship the order. For simple orders, when an order is shipped, its shipping fees are deducted from its customer's account. For compound orders, when an order is shipped, its shipping fees are deducted from all the customers who participated in the compound order.

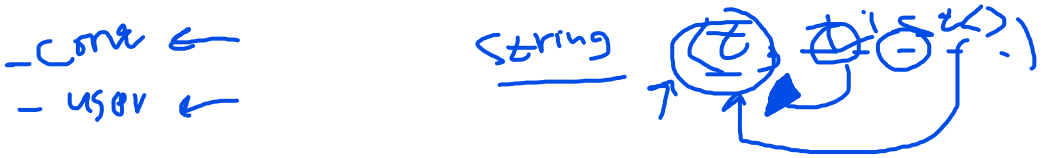
7. Notifications creation for various operations. The system manages those different notification "templates", and the languages these templates can be sent in, and of course the "placeholders" within the content of these templates to be replaced with "actual values".

   ex: " Dear {x} , your booking of the {y} is confirmed. thanks for using our store :) " this would be the template, but when the system "sends" it to the user "Nabil" who bought the item "mobile charger" , the actual email would be

   " Dear Nabil, your booking of the item mobile charger is confirmed, thanks for using our store :)"

   So, the "management" of those notifications templates, their subjects, content, available languages, available channels (should be: email, sms) , and placeholders would be the focus of this requirement. Your system should support at least two different notification templates for order





placement, and order shipment. Each of those templates should have different text, a different number of placeholders, where such templates types should be settled prior to launching the system.

8. For created notifications, you should implement a "notifications Queue", where you insert "notifications" that ARE TO BE SENT. You should be able to list the current content of that Queue.

Bonus part (Only the part highlighted in yellow is bonus):

9. Customers can cancel an order placement, or cancel only its shipping within a pre-configured automated duration. Such cancellation should update appropriately within the system.

10. After a configured time, messages are removed from the queue automatically to simulate that they were actually sent. Accordingly, the system should provide some live statistics to the overall software. The target of these statistics is to provide a vision about the notifications that are sent successfully:
    - The most notified email-address/phone-number.
    - The most sent notification template.

By now you should be able to develop web services and APIs. **In this assignment you are required to develop a RESTful API for this software.** Hence, all the requirements need to be exposed as a RESTful API.

For example, to be able to achieve a requirement for login, you would need to provide the following API functions within your implementation:

GET /user/check
A service to check if the user exists or not. This service returns all user info if exists
Input: email and password.

Note: The web service API should be structured in terms of Resources and actions. So you should have a section for each resource and each resource should contain the corresponding actions.

Evaluation Criteria

1. Properly working functionality as per the phase requirements.
2. Quality of project configuration (i.e. actual realistic usage of Git throughout the phase development by all team members. Such usage should reflect that team members actually used Github throughout the phase)
3. Consistency between the various submitted system models.
4. Consistency between the submitted system models, and the working product.
5. Quality of the design in terms of its usage of appropriate design patterns, and SOLID principles as needed.
6. Quality of the web service API design