

Phase 2:

Countdown Sales Service

(Business tier + Persistence)

1 Introduction

For the second phase of the project, we start presenting the expected architecture as well as the expected basic and advanced functionalities of this phase.

1.1 Description

The purpose of this phase of the project is to use concepts seen so far such as the EJB, the JPA and, if you want to go a step further, the notion of Web Socket and Server endpoint in order to automatically refresh the JSPs of your project. As described in Figure 1, you will have to implement a Business tier and connect EJBs to a database and the JMS topic and queue. In what follows, we give some information about the Business tier you must create and we also indicate again the expected functionalities of these two web applications, which are basically the same explained in Phase 1.

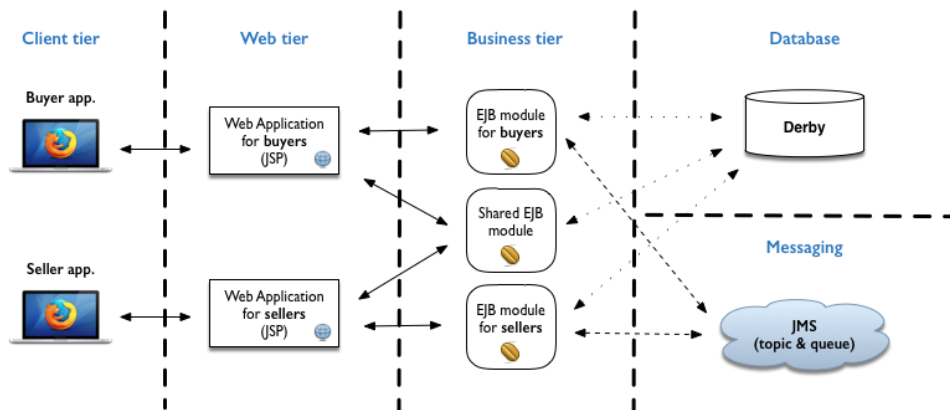


Figure 1: A suggested architecture for Phase 2

2 Information regarding the Business tier

In this Business tier, we suggest that you create three EJB modules as described in Figure 1. For example, you can firstly decide to create a *Shared EJB module* that deals with the authentication of buyer or seller with a stateless EJB and creates stateful EJBs for each connected buyer or seller to the web applications. This module can also store buyer and seller information in the database. To reach this goal, you must create JPA entities, which map objects to relational database tables.

Secondly, a specific EJB module for sellers can be created in which a singleton EJB can store all socket sessions created via the Server endpoint linked to the jsp of sellers, can receive and interpret all messages coming from the Message Driven Bean (Queue oriented). This module must also store offers created by sellers and update them according to the transactions received by buyers in the database.

Finally, a last EJB module for buyers can be created in which a singleton EJB can store all sessions created via the Server endpoint linked to the jsp of buyers, can receive and interpret all messages coming from the Message Driven Bean (Topic oriented). This module must also store the result of each transaction requested by buyers in the database.

3 Basic Functionalities (reminder)

The first phase of the project must contain the following functionalities:

- Java Web Application for sellers :
 - A seller can log into the web application with her username and her password.
 - The seller, logged into the application, can create a new limited commercial offer on her dashboard. This new offer is sent to the topic (JMS). A seller does not need to be able to modify an offer after its creation!
 - The seller can see the history of her created offers as well as the current stock of each offer.
 - When the application of the seller receives a transaction, which is a buyer request containing the reference of an offer as well as a quantity (this quantity must always be equal to 1), via the queue (JMS), the transaction will automatically be evaluated. If the transaction can be accepted according to the current stock of the offer, the application will send a positive notification to the buyer via the topic (JMS). During the evaluation of transactions, offers can be terminated if the stock reaches 0. In this case, a

notification must also be sent to all buyers via the topic (JMS) and the dashboard of the seller must be updated accordingly.

- Java Web Application for buyers :
 - A buyer can log into the web application with her username and her password.
 - All offers, received by the topic (JMS), are displayed in the dashboard of the buyer as soon as they are proposed by the sellers.
 - The buyer can select an offer as well as its quantity (this quantity must always be equal to 1). As indicated previously, this selection is called a transaction. Then, the transaction is sent to the queue (JMS).
 - When a notification is received via the topic (JMS), the dashboard of the buyer must be updated accordingly (remaining stock of the offers, if an offer is terminated or not with different colors).
 - The buyer must also see the history of all offers that were accepted by the application of sellers.

If you want to improve this phase, you can add advanced functionalities such as:

- Handling of errors (e.g., required fields in forms) by displaying appropriate messages for each type of errors.
- Customizing the web interfaces by including CSS.
- etc... (as you are free to add other improvements and features that you think might be useful but in this case please check with us your improvements propositions).

4 Presentations and deliverables

For the details about the presentation and deliverables, please check the *Project Guidelines* document available on the website of the course.