

Phase 3: Countdown Sales Service (Web service + iOS programming)

1 Introduction

The main goal of the third phase of the project is to create and propose geolocated offers to buyers. The other goals of this phase are to use concepts seen so far such as the creation and the use of a web service as well as the development of an iOS application. As described in Figure 1, you will have to create a REST web service located on your Java server, to create an iOS application and to use a remote publish/subscribe called *Adjago* in your Java server (via a REST API) and in your iOS application (via a SDK).

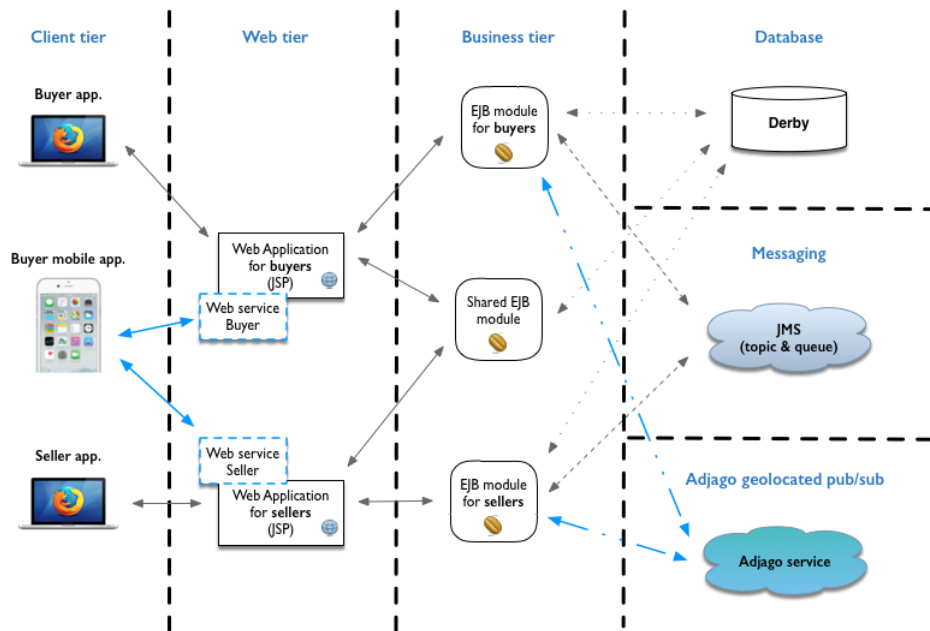


Figure 1: A suggested architecture for Phase 3

In what follows, we give some information about the web service you must create on your Java server as well as the iOS application. We also indicate again the expected functionalities of the iOS application because we will introduce the concept of geolocated offer. Several other functionalities must remain the same you implemented in Phase 1 and 2.

Adjago documentation:

- Adjago website with demo: <https://dev.adjago.io>,
- Adjago API doc.: <https://dev.adjago.io/documentation/api>,
- Adjago SDK doc.: <https://dev.adjago.io/documentation/sdk>.

2 Update of the basic functionalities (standard and geolocated offers)

For this phase, we will consider two types of offer: the standard offers, which are not geolocated, and the the geolocated offers, which have three additional attributes (i.e., latitude, longitude and range) to those of a standard offer described in the document presenting the overview of the project (http://doplab.unil.ch/wp-content/uploads/2016/12/EDA17_ProjectPresentation.pdf). Consequently, you will have to adapt the code you created during Phase 2 to take into consideration this new specification.

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

- Java Web Application for sellers :
 - A seller can log or register into the web application with her username and her password.
 - The seller, logged into the application, can create a new standard offer or a new geolocated offer on her dashboard. A standard offer is sent to the topic (JMS), while a geolocated offer is only sent to Adjago by using the REST API. 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 (standard or geolocated) as well as the current stock of each offer.
 - Standard transaction (1): When the application of the seller receives a transaction, which is a buyer request containing the reference of a standard 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 the 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.

- Geolocated transaction (2): When the application of the seller receives a geolocated transaction, which is a buyer request containing the reference of a geolocated offer for which the quantity is equal to 1, via the seller web service, 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 JSON response to the buyer.
- Java Web Application for buyers (same as the documentation for Phase 1 and 2) :
 - A buyer can log or register into the web application with her username and her password.
 - All standard 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 standard transaction. Then, this 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.
- iOS application for buyers (for sake of simplicity, the iOS application will be only created for buyers) :
 - A buyer can log or register into the iOS application with her username and her password via the Java web-service you created (Buyer side).
 - All geolocated offers are displayed in the dashboard of the buyer as soon as they are proposed by the sellers and the buyer are in the range of the location of the offer (the standard offers may not be displayed!).
 - The buyer can select a geolocated offer and send the purchase request via the Java web-service you created (seller side). According to Adjago, a geolocated offer is unique, the quantity is obviously 1.
 - The buyer must also see the history of all geolocated offers that were accepted.

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).

As depicted in Figure 1, the use of the database is required for Phase 3.

3 Tips

For sake of simplicity, you will use a file containing a sequence of locations to simulate the movements of a user owning the iOS device. This file will have to be specified in the running scheme of the project. These locations will be directly taken into account when you will start the iOS simulator and enable the location manager of the application.

4 Information regarding the architecture

As illustrated in Figure 1, the architecture contains four new modules: an iOS application for buyers, a buyer web service, a seller web service and a remote publish/subscribe service called *Adjago*. We suggest you to create two web services, one to process login and registration of iOS buyers (buyer side web service) and another one to process geolocated transactions of iOS buyers (seller side web service). As mentioned previously, the iOS application will be only created for buyers. The iOS application for buyers will interact with the *Adjago* service by using the SDK to subscribe and receive geolocated offers. The Java server will interact with *Adjago* by using the REST API to publish geolocated offers.

5 Presentations and deliverables

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