SUPINFO Academic Dept.

SUPINFO International University

INSTITUTE OF INFORMATION TECHNOLOGY

4JVA - Enterprise Programming Project

Document content

Subject
Delivery

Version 1.0
Last update: 22/11/2015
Use: Students/Staff
Author: SAD

SOMMAIRE

1 CONTEXT

SUPINFO wants to create its own bartering service and needs you to develop it. Because it needs to be powerful and scalable you naturally choose to use Java Enterprise Edition Technologies.

This project must be done by groups, each containing 3 or 4 students maximum. Working in a bigger group will be sanctioned by penalties points.

2 SPECIFICATIONS

The first version of the website will be composed of several functionalities listed below:

- As anonymous:
  - View a short description of this service and statistics in the index page
  - Register as a new user, authenticate himself
  - Use web services
  - Search objects
  - List last added objects in SupBartering

- As a registered user:
  - Add an object
  - Delete an object
  - View and edit his profile
  - Log out

You have to use EJB 3.1 and Servlet/JSP implementing JSP Model 2 Architecture and JPA implementing good practices (DAO, Factories, Criteria and MetaModel API...).

2.1 DATA STRUCTURE

Before starting the project, draw an UML class diagram representing the JPA Entities you will need with their relationships.

This diagram will be useful for you and for the team that will develop the next version of the platform.

The class diagram must be returned in ***jpeg***, ***png*** *or* ***pdf*** format (otherwise your Teacher will hate you!).

2.2 INDEX PAGE

For anonymous the index page must show a short description of this service and statistics of use (number of users, number of objects). You are free to add others statistics.

A navigation area must be displayed on the top of all pages (login, register).
For authenticated users, the navigation area displays a logout link, a link to his profile and a link to manage his objects.

This page displays the last objects stored in the database.

A search bar based on the name, price, description or type to find objects.
Show a button for each item to view details.
Add paging buttons at the bottom of this page if more than 10 objects are returned.

2.3 REGISTER AND AUTHENTICATE

When registering, users have to give some details about them, like username first name, last name, email address, a postal code and password.
Of course, you'll have to check user input.
When the user registers him, it must log him in too.

The user can authenticate by a dedicated log in page with username and password.

2.4 MANAGE OBJECTS

An authenticated user can manager his objects.
He can add or delete an object.
To add a new object, the user have to give some details about, like a title, a description, a price, a typeof object and a picture.
The page to manage his objects displays a list of all his objects.
For each object, display a link to delete it. A confirmation is needed.

2.5 VIEW AND EDIT HIS PROFILE

Users can view a profile page to change their details (except username).

2.6 WEB SERVICE

This application provides a web service (REST) to list objects and search objects by name, price, description, type to find objects
This Api returns JSON only.

1.1 LOG OUT

This functionality must log the user out.

3 INSTRUCTIONS

- Plagiarism is forbidden.
- Make accessible his code on a public sharing platform (as GitHub) before the end of the evaluation is forbidden.

Don't abiding by these rules will result in suspension of your assessment and will be considered cheating.

4 NOTATION

Functionalities	Points
Data Structure	2
Index (objects list, paging, navigation area, …)	5
Security (register, auth, log in/out, filters)	2
Search objects on index page	5
Object details page	3
Add / remove an object	3
EJB 3.1 is used	3
View and edit profile	2
Dao factory is used	2
Criteria API & MetaModel	3
Webservice list Objects	2.5
Webservice search objects	2.5
Design	2
Code Quality & Conventions	3
TOTAL	**40/40**

5 RETURN

Return your graded exercise as a ZIP archive named as follows:
4JVA_SupBartering_Campus_IdBooster.zip.
For example: 4JVA_SupBartering_Troyes_10000.zip
Not following this convention will result in point loss.

You will send the archive **to your Teacher SUPINFO email address and a copy to 4JVA@supinfo.com** to secure your project. Send it **before the 18th December 2015 before 23:59**. After that delay, your graded exercise **will not be corrected and the mark 0 will be assigned to you**.