BSc. (Hons) in Computing

Enterprise and Cloud Computing

Assessment 1, Java EE

Value: 25%

Date dues: Deliverables 1 to 4 should be completed by Wed Nov 9th; Final report and code to be uploaded to Moodle by Wednesday Nov 23rd, 5:30pm. Demos will take place during class on Wednesday Nov 23rd.

Objective: Design and implement a Java EE application.

Learning Outcomes:

- Programming with Java EE
- Implementing Entity beans, Session beans and a Message bean using a JMS server.
- Implementing static and dynamic JPQL queries using Entity and Session beans

Business Domain:

Pick a business scenario to model that has, at a minimum:

- **business transactions** involving three entities including a **product** or **service** and a **person** (customer, supplier, employee etc.);
- business logic that requires both synchronous and asynchronous communication (i.e. the solution is to include session and message beans);
- A web interface implemented using the JSF framework

Examples:

Ordering a product online, where online confirmation is given synchronously, but an e-mail confirmation is send later (asynchronous).

A system for managing contract journalists, which accepts articles from journalists and processes payments based on the number of articles submitted (could be a scheduled task run each Friday). Like above, the journalist could receive immediate confirmation that the article is uploaded, while notification to the editor that an article is awaiting review could be done asynchronously.

Marks will be allocated for including additional API's and services such as security and scheduling.

Deliverables:

1. A short description of the problem domain and business scenario you are modelling.
2. An ERD or class diagram for the business domain, illustrating the entity classes needed.
3. A description of how the business logic is to be mapped to session bean(s) and message bean(s).
4. A diagram of the architecture of the system (like the diagram at the end of lab sheet 3, showing interactions between classes).
5. Screen shots of your user interface
6. A paragraph detailing APIs and/or services you use in your implementation (including dependency injection). The purpose of this is to illustrate your understanding of the Java EE container support you are availing of.
7. A demo of the system during scheduled class time.

What to upload to moodle:

Upload both your report and a zip file containing your netbeans project files to Moodle. There is a single upload link, which contains two tabs, one for the report and one for the code.

Note: there are a number of Netbeans tutorials to create applications like this. Implementing these tutorials for your assessment is NOT acceptable. You must develop your idea from scratch.