Java Technologies - Lab 7

[valid 2022-2023]

Contexts and Dependency Injection (CDI)

1. (2p) Create a JSF application for managing the submission of documents into a repository. Documents have a name, [at least]* one author and a single file (content).
   The application will allow the following:
   - An authentication mechanism based on username and password (implement it as you please, we'll rewrite it soon).
   - Register new users and assign them a specific role, for example *admin*, *author*, *reviewer*, etc.
   - Specify a time frame, in which registration is open for users and submissions.
   - The possibility to upload a document (for authors) and to view all uploaded documents (for admin). Each uploaded document will have a uniquely generated registration number. All submissions will be logged in a text file.
2. (3p) Use **Contexts and Dependency Injection (CDI)** for:
   - (0.5) the management of application's beans (@Inject) and transactions (@Transactional);
   - (0.5) decoupling the components using dependency injection (for example, use a producer method to generate registration numbers) (@Produces);
   - (0.5) decoupling orthogonal concerns, such as logging; (@Interceptor)
   - (0.5) decoupling bussines concerns, such as verifying the date for operations like registration and submission (@Decorator);
   - (0.5) implementing at least one event-based comunication (for instance, whenever a new document is uploaded a message is produced and all observers of this type of event will be notified) (@Observes);
   - (0.5) data validation, using Bean Validation annotations.

   The original use other CDI concepts such as alternatives, specializations, etc. will be appreciated.

   Bibliography:

   - Slides
   - Java EE Tutorial: CDI
   - Weld - Reference Implementation

- Arquillian: An integration testing framework for Java EE
- Java EE Tutorial: Bean Validation