Politecnico
di Milano

Software Engineering 2
JEE Laboratory

Michele Guerriero
michele.guerriero@polimi.it

Valerio Castelli
valerio1.castelli@mail.polimi.it

Request specification

- Write a basic JEE application which allows an user to just register and log in.
- You can freely chose the kind of web components to adopt between JSP and Servlets.
- You have to use Enterprise Java Beans to implement your business logic.
- Users information (as specified during the registration phase) has to be stored in an RDBMS (e.g. Derby Java DB, MySQL). To interact with the database you have to use the Java Persistence APIs and a Java Database Connectivity (JDBC) connection.

Platform selection

- You can freely chose your IDE (e.g. Eclipse, Netbeans), the application server (e.g. Glassfish, JBoss) and the DBMS (e.g. Derby Java DB, MySQL) to use.

- To reduce the installation and configuration time, we propose to use Netbean which already includes Glassfish, and Derby as the database, which, in turn, is included in Glassfish.

Requirement specification: registration process

- In order to register, a user is asked to insert a name, surname, age, email address and password.
- All the fields are mandatory.
- The user is uniquely identified by her email address.
- The registration process has to reject a registration if the specified email is already associated with an existing user in the database.

Which JEE components do we need?

Which JEE components do we need?

- Start by creating the User JPA entity to store users' information in the database
- Configure properly the persistence.xml (you might refer to the one used in the class example)
- Create a (stateless?) session bean to manage the registration logic (i.e. check registration fields and store/reject the new user)
- Create the web layer according to the technology you selected (JSP, Servlet)

Requirement specification: login process

- In order to log in a user has to provide her email address and password.
- The login process have to reject a login request if any user with the specified pair of email address and password is found in the database.
- In the cae of rejected login, the login process has additionally to notify the user about the reject cause:
  - reject because the inserted email does not exist in the database
  - reject because the inserted password does not match with the one stored with the inserted email, but the inserted email has been found in the database

References

- You might refer to the set of slides presented during the first two JEE classes
- Have a look at the example application discussed during the first two JEE classes: https://github.com/MicheleGuerriero/se2-class-example
- The JEE 7 official tutorial is a much more complete and detailed reference

Let's make the application a bit more interesting

- Once an user is able to register and to log in, the system is supposed to provide her with few functionalities
- Let's consider the PowerEnJoy system, for which we know the functional requirements…
- You are asked to implement (a simple prototype) one or more among the possible functionalities

Some of the possible functionalities to implement

- An user might insert information about a new available car (ideally an administrator, but you can assume there is a single type of user doing everything)
- An user might look at the list of all the registered cars with the associated status (free/busy)
- An user might place a reservation for a specific car
- At the end of the usage the user might finalize the reservation
- ….

How to proceed

- Each functionality need an entire process to be correctly implemented
- Remember to think about the exceptions that could raise within a given process
- Pick up a functionality, isolate it and start thinking at how to implement it
- You might start from the car insertion process, since without any car the system is useless…
- Once you implemented the car insertion functionality you might move to the reservation management!
- Keep in mind what we have seen, think at which entities do you need, which web and business components, etc.