

USAL1J - Architecture EJB Exam

*Any document is authorised.
The indicated point values for the questions are approximate, and might change.*

Duration : 3 hours

Problem

A sport club wants an EJB application to manage its members, and the activity it proposes.

The database contains the following tables:

MEMBER <u>memberId</u> login password name address	MEMBER_ACTIVITY <u>#memberId</u> <u>#activityId</u>	Activity <u>activityId</u> label cost #teacherId	TEACHER <u>teacherId</u> name phone
--	--	---	---

(underline is used for primary keys; «#» in front of a field means it's a foreign key).

«label» is a name of an activity (like «football»); cost is the cost of an activity for a member.

Question 1 (5pts)

Write the corresponding entity classes for JPA, with the correct annotations (for ids annotations, you are free to choose the solution you prefer).

We specifically want a bi-directional link between members and activities.

You don't need to write all accessors (getter and setters), but if an accessor is more complex than

```
void setName(String name) {this.name= name;}
```

you must write it.

Question 2 (2pts)

Write the java **jpa** code to create and save in the database the activity «swimming» with a cost of 80, and teacher «Albert».

You can consider that an entity manager object is available for you to use and you don't need to create it.

Question 3 (4pts)

Write the **JPQL** queries to get the following informations:

- all activities with a cost greater than 30 ;
- activities taught by a teacher named «Albert»
- all members who have the activity «swimming»
- all members who have both the activity «swimming» and the activity «taichi».

Question 4 (conception problem, not technical) (1pt)

Suppose the rule to compute a member's fee to the club is the following: the fee is the sum of the costs of the member's activities, minus 25% if he has 3 activities or more.

- If you are asked to put this method in one of the classes you have already written in Question 1, where will you put it?
- Now, suppose that we now that the fee computation system might change at some point, maybe in a future version of the software (and, for instance, also depend on the total number of members in the club). **Explain** if the first solution is a good one if the software can change.

Question 5 (2pts)

As the member object contains the member's login and password, the web application can keep it as a session object when the user has logged. Now, using a form, the user changes the value of his address. The software modifies the address in the member object kept in the session. However, the value in the database doesn't change.

- why
- explain in a line or two what should be done so that the address is really changed in the database

Question 6 (5pts)

Consider the following use case

to manage existing users and inscribe them to activities, an administrator will:

- select users from a list of users (it might be done on many pages, so the list of selected users must be stored somewhere);
- then select an activity from the list of activities
- and validate his choice, which will add the selected activity to the activities of each of the selected user

The club managers want a technology which can be independent of the front-end system (web, stand-alone java application, interactive or batch processed...)

So, your task is to propose and write the EJB(s) objects, stateful and/or stateless, to provide the necessary methods (concentrate on the needed methods, don't add other methods simply because they might be useful).

Tips:

- you probably don't need to write a lot of code - depending on your choice, I would say you need *at most* two EJBs;
- beware that we want operations to be atomic: if the validation fails at some point, for instance for one user, it should be cancelled for all of them.

Question 7 (1pt)

In the previous question, we said we wanted the modifications to be cancelled in case of failure. Explain **why** it works in your code (the explanation can be quite short).