COS30041 Creating Secure and Scalable Software [Java EE]

Pass Task 4.1 Stateless Business Logic

Time Frame: Weeks 4 – 5
Suggested to start and complete in Week 4
Submission Due: Week 5, Friday 6:30pm

Overview

In this task, you are required to program business layer objects that can handle business logics on the server using related technologies. You are also required to demonstrate your work is of good quality.

Purpose	To demonstrate your ability to use relevant technologies to develop quality application that handles the business logics on the server
Tasks	1. Learn to develop an enterprise application that uses stateless session bean to perform the ORM of an entity class 2. Extend the stateless session bean so that it can handle all CRUD operations 3. Extend the enterprise application client program developed in 1 above to request the actual CRUD operations provided by the stateless session bean 4. Answer questions related to the design of the project
Pre-req Task[1]	Pass Task 3.1
Follow-up Task[2]	Pass Task 5.1
Suggested Time	2 hours if you know the stuff well 6 – 8 hours if you need to read the concepts and learn how to program class on the Business Logic Layer **10+** hours even if you “know” what to do and just make ONE simple mistake during the programming
Resources	Lecture 04a Business Logics 1 Lecture 04b EJB Lecture 04c Stateless Session Bean Java EE – Stateless Session EJBs
Feedback	Ask your tutor for feedback
Next task	Pass Task 5.1

Pass Task 4.1 Submission Details and Assessment Criteria

You must create your own document (pdf) in **portrait** mode[3], which you will upload to Doubtfire, with the following details:

- Your name and student id
- Your tutor’s name
- Your own responses to the tasks according to the corresponding instructions (see below)

[1] You need to complete the pre-requisite task before doing this task.
[2] You need to complete this task in order to do the follow-up task because the follow-up task depends on your answer in this one.
[3] Landscape mode pdf does not work properly in Doubtfire.

Tasks and Instructions

Assumption: **You have implemented the Entity Class, Myuser, in Pass Task 3.1**

Task 1. Complete Lab_04_SLSB_MyuserFacade_DTO

Task 2. Programming [Assume you have completed Task 1 above]
Add the following methods in the "MyuserFacade.java" class (Lab_04_SLSB_MyuserFacade_DTO) that supports the remaining CRUD operations of MYUSER

1. "`private MyuserDTO myDAO2DTO(Myuser myuser)`" – accepts a Myuser object and returns a MyuserDTO object
2. "`MyuserDTO getRecord(String userId)`" – accepts a String object whose value is the userId of a record to be searched. If the record can be found, it returns a MyuserDTO object that stores the information of the actual database record. Otherwise, it returns a "`null`" object.
   Hint: It may call the "myDAO2DTO" method.
3. "`boolean updateRecord(MyuserDTO myuserDTO)`" – accepts a MyuserDTO object and checks whether the actual record exists in the database. If it does, it will update the information of the record with the current information stored in the MyuserDTO object and return `true`. Otherwise, it returns `false` without doing anything.
   Hint: Use the EntityManager's instance method called `merge()`
4. "`boolean deleteRecord(String userId)`" – accepts a String object whose value is the userId of a record to be deleted. If the record can be found, it removes the record in the database and return `true`. Otherwise, it returns `false`.
   Hint: Use the EntityManager's instance method called `remove()`
5. "`ArrayList<MyuserDTO> getRecordsByAddress(String address)`" – accepts a String object whose value is the address of a user. It then searches the MYUSER table and returns all records having the same address as an "`ArrayList<MyuserDTO>`". Otherwise, it returns a "`null`" object.
   Hint: The following code segment may help. But, you need to read the Java EE API on how to use them properly. For your other tasks in the future.

```
javax.persistence.Query query;
query = em.createNamedQuery("Myuser.findByAddress").setParameter("address", address);
~~Array~~List<Myuser> daoList = ~~(ArrayList<Myuser>)~~ query.getResultList();
```

Note: When reading the Java EE API, you will be amazed by how many different ways you can achieve the same thing. However, due to time limitation, please stick to the current one. You are strongly encouraged to explore other possibilities on your own time.

Task 3. Programming [Assume you have completed Task 2 above]
Modify the MyuserAppClient.java class so that it acts as a test harness of the methods you implemented in Task 2 above.

Note: Although this is a test harness, there is no need to use JUnit Tests.

Task 4. Answer the following questions:

4.1. Which class is responsible for doing all the ORM work? MyuserFacade / Myuser? Justify your answer.
4.2. Explain (in your own words) the concept of bean instance pooling in the context of stateless session bean using "MyuserFacade" as an example. Also explain how it can achieve scalability.

Submission Task
Once completed, you need to submit a pdf file that contains all your work (e.g. selected code segments – show me the key stuff and some screen dumps of your testing)

Demonstration

You may be asked to demonstrate your assignment in the lab. You should be able to do this and explain your code when asked in the lab session.