## COS30041 Creating Secure and Scalable Software [Java EE]

**Pass Task 3.1 Entity Persistence and ORM**

**Time Frame: Weeks 3 – 4**
>        **Suggested to start and complete in Week 3**
>        **Submission Due: Week 4, Thursday, 6:30pm**

**Overview**
In this task, you are required to program a data access object that can access the content of a database table via Entity Persistence technologies, or Java Persistence API. You are also required to demonstrate your work is of good quality.

| Purpose | To demonstrate your ability to develop quality application that involves the uses of entity persistence technologies to access the database content |
|---|---|
| Tasks | 1. Learn to develop an application that uses JPA API to manage an entity class<br>2. Extend the application so that it can handle all CRUD operations<br>3. Develop a client program that requests the actual CRUD operations provided by the entity class / object developed in 1 and 2 above as your test harness<br>4. Answer questions related to the design of the project |
| Pre-req Task[1] | Pass Task 2.1 |
| Follow-up Task[2] | Pass Task 4.1 |
| Suggested Time | 1 hour if you know the stuff well<br>4 - 5 hours if you need to read the concepts and learn how to use ORM to map to a database server |
| Resources | Lecture 03 Entity ORM<br>Java EE – JPA |
| Feedback | Ask your tutor for feedback |
| Next task | Pass Task 4.1 |

**Pass Task 3.1 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)

**Tasks and Instructions**

**Task 1.**  Complete Lab_03_Entity_Class_JPA

**Task 2.**  Programming [Assume you have completed Task 1 above]
Add the following methods in the MyuserDB.java class (Lab_03_Entity_Class_JPA) that supports the remaining CRUD operations of MYUSER using JPA
>     1. "`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.

---

[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.

2. "`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()`

3. "`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()`

**Task 3.**  Modify the MyuserApp.java class so that it acts as a test harness of the CRUD operations developed in Task 2 above

**Note**:    The client program can be a desktop application (either console or with GUI). Console is the simplest. Or, it can be a web application (but this is too much work at the moment).

**Task 4.**  Answer the following questions:
   4.1.  Which class is responsible for doing all the ORM work? MyuserDB / Myuser? Justify your answer
   4.2.  Is the role of the Myuser entity class here different from that of the "Myuser" class in the application in Lab 02. Why? Justify your answer.
   4.3.  Why Myuser entity class in this application domain is different from that in the Lab 02 (in the JDBC domain)?

**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.