SWE80005 Enterprise Development [Java EE 7 – NetBeans 8.2 + GlassFish Server 4.1.2 fix]

Pass Task 2.1 Programming with Java Database Connectivity [JDBC]

Time Frame: **Weeks 1 – 2**
Demo Needed: **Week 2 Lab**
Submission Due: **Week 2 Friday 23:59:59**

Overview
In this task, you are required to program data access object that can access the content of a database table via Java Database Connectivity, JDBC. You are also required to demonstrate your work is of good quality.

Purpose	To demonstrate your ability to develop quality application that connects to database
Tasks	1. Learn to write an application that can access the data stored in a database table via JDBC 2. Extend the application developed in 1 above so that it can now handle all the actual CRUD operations 3. Program a client application to test the program developed in 2 above 4. Prepare your test cases and test your application thoroughly by using various database contents (e.g. different records) 5. Demo your work 6. Answer questions related to the design of the application
Pre-req Task[1]	Pass Task 1.1
Follow-up Task[2]	Pass Task 3.1
Suggested Time	1 hour if you know the stuff well 4 – 5 hours if you need to read the concepts and know how to establish database connections
Resources	Lecture 02 Database Connectivity; JDBC
Feedback	Ask your tutor for feedback
Next task	Pass Task 3.1

Pass Task 2.1 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_02_Database_Connectivity

Task 2. Add the following methods in the MyEmsEmployeeDB.java class in Lab_02_Database_Connectivity that supports the CRUD operations of the EMS_EMPLOYEE table in Java DB using JDBC

1. "`boolean createRecord(EmsEmployee emsEmployee)`" – accepts an EmsEmployee object and checks whether the actual record exists in the database. If the record does not exist, it will create a new record in EMS_EMPLOYEE database with the information in the EmsEmployee object and return `true`. Otherwise, it returns `false` (and does not create the record).
2. "`EmsEmployee getRecord(String empid)`" – accepts a String object whose value is the employee id of a record to be searched. If the record can be found, it returns an EmsEmployee

[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.

object that stores the information of the actual database record. Otherwise, it returns a "`null`" object.

3. "`boolean updateRecord(EmsEmployee emsEmployee)`" – accepts an EmsEmployee 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 EmsEmployee object and return `true`. Otherwise, it returns `false` without doing anything.
4. "`boolean deleteRecord(String empid)`" – accepts a String object whose value is the employee id of a record to be deleted. If the record can be found, it sets the "Active" field in the corresponding database record to `false` and returns `true`. Otherwise, it returns `false`.

Task 3. Develop a client program that requests the methods you developed in Task 2 as a test harness.

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

Task 4. Write your test cases (including the database content and input values) and test your work thoroughly. Remember to collect the screen dump (e.g. "Console" in NetBeans or your "JUnit" test results)

Hint: What is the minimum number of test cases that you need to demonstrate your methods work **comprehensively**?

Task 5. [Demo Task] Perform a demo of Task 4 in front of your tutor in Lab in Week 2. Make sure your tutor has dropped down your name and student id for record purposes.

Note: Without the demo, your Portfolio Task will not be marked as "completed" and you will not get any "feedback" from the tutor. In case, you get stuck in the lab task, contact the tutor immediately to get help.

Task 6. Answer the following questions:

In this lab, all classes developed by you sit in one machine, but let us assume the following:

The client programs (SetUpMyEmployee.java and any other programs that you developed for requesting the CRUD operations of EmsEmployee DB table) sit on one machine (Machine A), MyEmsEmployeeDB sits on another machine (Machine B), and the Java DB sits on a third machine (Machine C).

6.1. What is the role of the EmsEmployee class? DAO / DTO / both? Justify your answer.
[Hint: In case, the "EmsEmployee" class is not a DAO. Then, it must be "MyEmsEmployeeDB" taking the role of a DAO. Remember what constitutes a DAO: Is it just the name? or the services it provides?]

6.2. In Machine A, if a program (e.g. SetUpMyEmployee) calls "`emsEmployee.setName("ABC")`", will this change the corresponding value in the database server? Why or Why not?

6.3. In Machine B, if MyEmsEmployeeDB calls "`emsEmployee.setName("ABC")`", will this change the corresponding value in the database server? Why or Why not?

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.)

Compulsory Demonstration in Lab
You need to demonstrate your work in the lab so that the tutor can keep a record to mark your submission. You should be able to do this and explain your code when asked in the lab session.