ENSF 519 – Principles of Software Development
Fall 2019

Lab Assignment #8: Event-Handling, JDBC

Due Dates	
Project I	Submit electronically on D2L before 11:59 PM on **Monday November 18th.** **Only one** group member should submit the project on D2L but **put both names on the submission.**

This is a **Group** assignment. Students can work in groups of two. You can pair up regardless of your TA groups. Please note, students have the option of working on their own.

Important Note: Even though this is a group assignment, it is **very important** that both people in the group work on all exercises and collaborate closely together. This is a great opportunity to gain valuable pair programming (https://www.agilealliance.org/glossary/pairing/) experience and engage in collaborative design. Taking the approach of "I do this part, and you do the next" is not a good idea and simply results in a missed opportunity!

The objectives of this project are:

1. Java Graphical User Interface Event-Handling
2. Working with database through a Java program
3. MVC


The following rules apply to this project and all other assignments in future:

1. Before submitting your project reports, take a moment to make sure that you are handing in all the material that is required. If you forget to hand something in, that is your fault; you can't use `I forgot' as an excuse to hand in parts of the assignment late.
2. **20% marks** will be deducted from the assignments handed in up to **24 hours** after each due date. It means if your mark is X out of Y, you will only gain 0.8 times X. There will be no credit for assignments turned in later than 24 hours after the due dates; they will be returned unmarked.



Lab (50 mark)

Exercise - 1: Accessing a MySQL Database with Java (8 Marks)

For this exercise, you need to download `InventoryManager.java`, `Tool.java`, and `ItemsNew.txt` from D2L, then install MySQL and JDBC on your computer. Detailed instructions for MySQL are posted on D2L under "Some Help with MySQL".

What to do:

Task 1 – Create an empty database on your computer with MySQL. Then, make any necessary changes to the `InventoryManager` class to make it connect to that database and run without error. You will need to provide your username, password, and a valid path to the database to connect. If you are successful, your program output should look like this:

```
Connected to: jdbc:mysql://localhost:3306/InventoryDB

Created Table ToolTable

Filling the table with tools
Reading all tools from the table:
Tools:
1000 Knock Bits 88 12.67 8015
1001 Widgets 10 35.5 8004
1002 Grommets 20 23.45 8001
       .
       .
       .
1038 Googly Eyes 756 6.99 8001
1039 Handy Pandies 321 4.35 8017
1040 Inny Outies 219 3.45 8010

Searching table for tool 1002: should return 'Grommets'
Search Result: 1002 Grommets 20 23.45 8001

Searching table for tool 9000: should fail to find a tool
Search Failed to find a tool matching ID: 9000

Trying to remove the table
Removed Table ToolTable

The program is finished running
```

Hint: There are many ways to create a database using command line, MySQL Workbench or even a java program, but the database only needs to be created once.


Once created, you should be able to connect to it with `InventoryManager` by providing the correct path.

Task 2 - Next, your job is to replace the `Statement` object in `InventoryManager` with a `PreparedStatement`. You will need to modify each method accessing the database to make the program function as it did before.

Note: For full marks in this exercise, wildcard characters (?) should be used in the `addItem` and `searchTool` methods.

What to Submit: Please submit all the java files including the files you have created/modified in a zip folder. You need to provide the Javadoc comments in your code, but you don't need to submit the HTML files.


Exercise - 2: Client Management System-GUI (12 Marks)

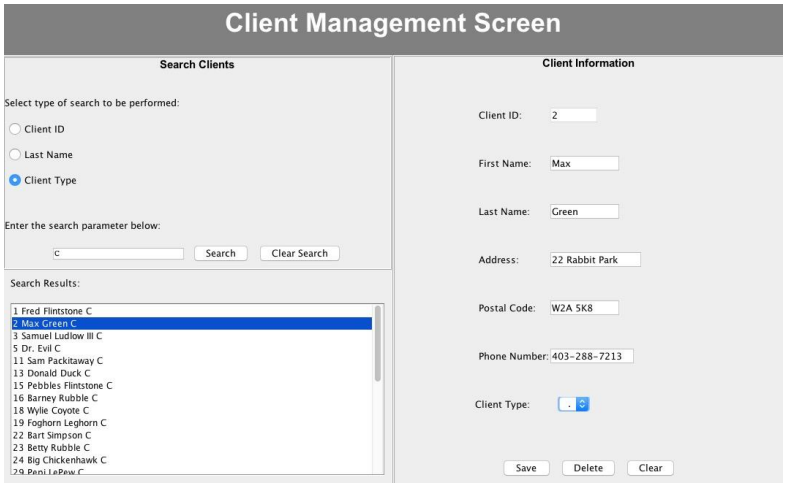
In this exercise, you are supposed to create the GUI required for a client management system. You will complete this assignment in Exercise 3.

A client has an ID which is a unique number assigned to the client when defining a new client. Firstname, lastname and the address of the client can have maximum length of 20, 20, and 50 characters respectively. The maximum length of the postal code and phone number fields are 7, and 12 characters respectively. A client can be of two types: Residential (R), or Commercial (C).

The client management system will support following functionalities:

- add a new client
- modify an existing information for a client
- remove a client from the database
- search for a client based on id number, name or type.

A sample GUI is shown in the following image (this is just an example; you are free to design your GUI differently). In the left panel, the user can search for a client based on three criteria. The results of the search are shown the bottom left side of the window. This list is created using a `JScrollPane`. The items of the list have a listener which enables us to click on them to see the detailed information of the customer in the right window. In the right side of the window, we can modify an existing client, or delete it. We can also add new clients.



What to Submit: Please submit all the java files including the files you have created/modified, along with a PDF file containing a sample output (i.e. screen shots of your GUI) of your program in a zip folder. You need to provide the Javadoc comments in your code, but you don't need to submit the HTML files.



Exercise 3: Client Management System Functionality (18 Marks)

Task 1 – Create a database which has client table with following attributes. The information of the clients that you need to add to your table is given as a text file and you can download it from D2L. Client.txt contains following information:

```
First name; Last name; Address; Postal code; Phone number; Client type
```

Your frontend (i.e. GUI) has to communicate with the backend (i.e. database) to store or retrieve the client information. You **cannot** use the text file as your database!

Client		
id	NUMBER(4)	PRIMARY KEY
firstname	VARCHAR2(20)	NOT NULL
lastname	VARCHAR2(20)	NOT NULL
address	VARCHAR2(50)	NOT NULL
postalCod	CHAR(7)	NOT NULL
phoneNumber	CHAR(12)	NOT NULL
clientType	CHAR(1)	NOT NULL

Task 2 – You need to implement the functionalities of the search, add, delete and save/modify buttons, by adding event listeners to your GUI. The required functionalities are described in the previous exercise.

Following rules must be considered when adding or editing information of a client. If an invalid input is entered, a proper message has to be shown on the window.

- They are in the format A1A 1A1, where A is a letter and 1 is a digit, with a space separating the third and fourth characters.
- The phone number is in the format of 111-111-1111, where 1 is a digit.
- A client can be of two types: Residential (R), or Commercial (C).
- The maximum length of first name, last name, and address are 20, 20, 50, respectively.
- The id field is a unique field assigned by the system when defining a new client and it cannot be changed by the user.

What to Submit: Please submit all the java files including the files you have created/modified in a zip folder. You need to provide the Javadoc comments in your code, but you don't need to submit the HTML files.


Exercise 4: MVC Design Pattern (12 Marks)

In this exercise you will modify the design of your exercise 3 such that it follows the MVC design pattern. In doing so, you will separate the code responsible for the database (i.e. Model), and the user interface (i.e. GUI), and enable their communication through the controller. Please carefully consult the MVC example provided in class.

What to Submit: Please submit all the java files including the files you have created/modified in a zip folder. You need to provide the Javadoc comments in your code, but you don't need to submit the HTML files.

How to submit: Include all your files for this lab in one folder, zip your folder and upload it in D2L before the deadline.