### CSCI 470/680E Fall 2014 -- Assignment 6 -100points Java Database and client/server

**Description**

Assignment 6 is a 3-tier program that involves client-server communication as well as JDBC.

**-------------------------------------------------**

You will write both the Server and the Client. Base your Server program on the example Server program (<http://www.seasite.niu.edu/cs580java/javaServer.html> ) and example gone over in class.

The Server program must be stored and run on hopper.cs.niu.edu. Your instructor will let you know which port your server will use. Ports available are from 9711 to 9790.

Write your own client (You can start from <http://www.seasite.niu.edu/cs580java/javaClient.html>).  The client is a command line program. You need to design a simple text-based menu. It needs to be clear enough for the user to select one of the JDBC commands (e.g. 1. ADD; 2. SEARCH; 3. DELETE etc. See details of the commands below.), and pass necessary information to the program. Use prompt to remind user about the required input. Use validation to ensure the correct data types.

The commands and information sent from the client to server should be the following SQL requests:

* **ADD** a new customer. The user will need to select this command, and then enter 4 pieces of information on the client application based on the 4 columns of the **cust** table explained below. Once all the information are provided by the user, the client sends them to the server.
* **SEARCH** for a customer by SSN. The user will need to select this command, and then enter an ssn. Once all the information are provided, the client sends them to the server.
* **DELETE** a customer. The user will need to select this command, and then enter an ssn. Once all the information are provided, the client sends them to the server.
* **UPDATE** a customer **address**. The user will need to select this command, and then enter an ssn and the new address. Once all the information are provided, the client sends them to the server.
* **GETALL** - when the user selects this command, the client sends the choice to the server.

Development and testing of client-server apps is somewhat awkward. To test a new version of the server, you will have to

* *kill* the currently running server program
* start the new one (it is recommended to run the server in the background)

Testing a new version of the client is simpler - just re-compile and re-run.

**Application Protocol:**

You can use Strings to transfer messages back and forth between the client and server, including the type of command and the related information (and eventually the feedback from server).

**Use** [**Object Serialization**](http://www.seasite.niu.edu/cs580java/Object_Serialization.html)(e.g. serialized object of a class that holds the customer information, <http://www.seasite.niu.edu/cs580java/Object_Serialization.html> )to send messages back and forth between the client and the server.  To minimize communications, include a transaction field in the objects sent from the client to the server to indicate the type of transactions (e.g. add, update, or list etc.), as opposed to sending a message indicating the transaction type followed by a separate message with the transaction details.  You will need to devise a way for the server to send multiple records from a *GetAll* request and for the client to retrieve those records. Even if you choose to use Strings to send messages back and forth, you are still required to study the material for object serialization.

**Database table schema**

The only access you have to the database is through your Java Server program (after part 1). The database you connect to contains one table. The table is named **cust**, so this is the table you should specify in your SQL queries. This table is probably **not** initially empty.

The cust table has 4 columns. These are:

Column Number         Column Name         Data Type  
        1                         name                         char(20)  
        2                         ssn                            char(10)  
        3                         address                     char(40)  
        4                         code                          int

The ssn column has a unique primary index on it and thus all values in this column must be unique.  All database names (table name, column names) are lowercase and must be coded as such in SQL statements.

**Submission**

Bring to Professor Leon at your assigned date and time and you will demo it in person for him.

This is what the what one can make using “xming” to run java database pgms…….

