CSI 203 – Final Project

# GUI programming using swing

**To:** *Dr.Lobo*

**From**: *Suvin Seal*

**Subject:** *Java, Object Oriented Programming*

**Date:** *December 9th, 2018.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Title:**

Integration of SQLite and java for data collection/management.

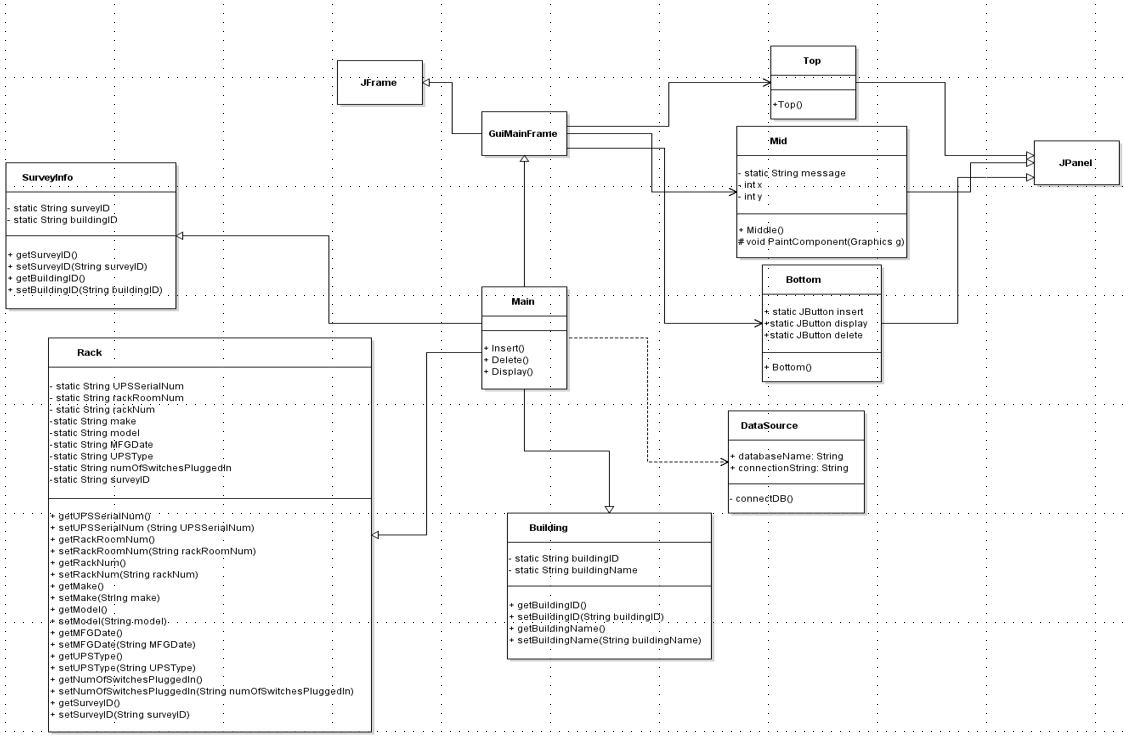
**Overview:**

The main goal of this project is to decrease the work load of student employees who work at the Office of Information technology at WAC. To collect data and store it on a database, the Java database connectivity library JDBC will be used to store data entered by the users. Java swing will be used to make buttons and have GUI display which will make the software user friendly and easy to use.

**Background:**

To collect data from wi-fi routers and UPS information, students usually use PowerApps which is integrated with SharePoint, however, PowerApps only works with internet connection. On situations where students have no access to the internet, this software can help students collect data and store it in a database. This data can later be exported to the SharePoint website or other DBMS like MS Access where queries can be coded for data management.

**Deliverables:**



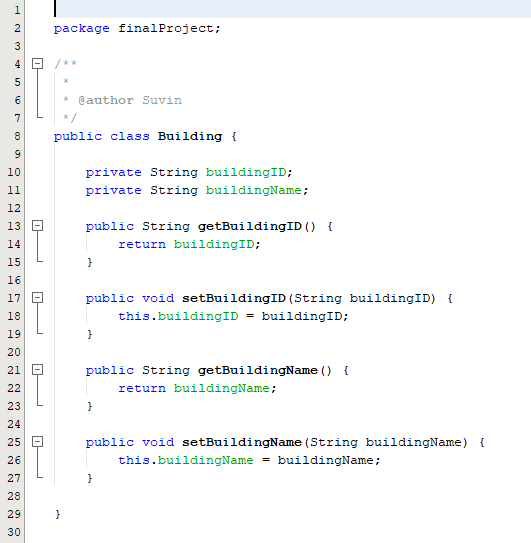
The project has two parts. The first part is to use JDBC, an API in Java which is used for client-side handling and managing a database. For my project, I used JDBC to connect my Java code with SQLite. Data transfer is much faster in SQLite and Java compared to databases like mySQL workbench or Microsoft SQL server because of which SQLite was chose to be the primary database in this project.

Classes SurveyInfo, Rack and Building are three classes which have the accessors and mutators to set and get all attributes from the tables in SQLite. Once the queries are called in Java, these classes are used to make changes in the attributes of the tables in the database.

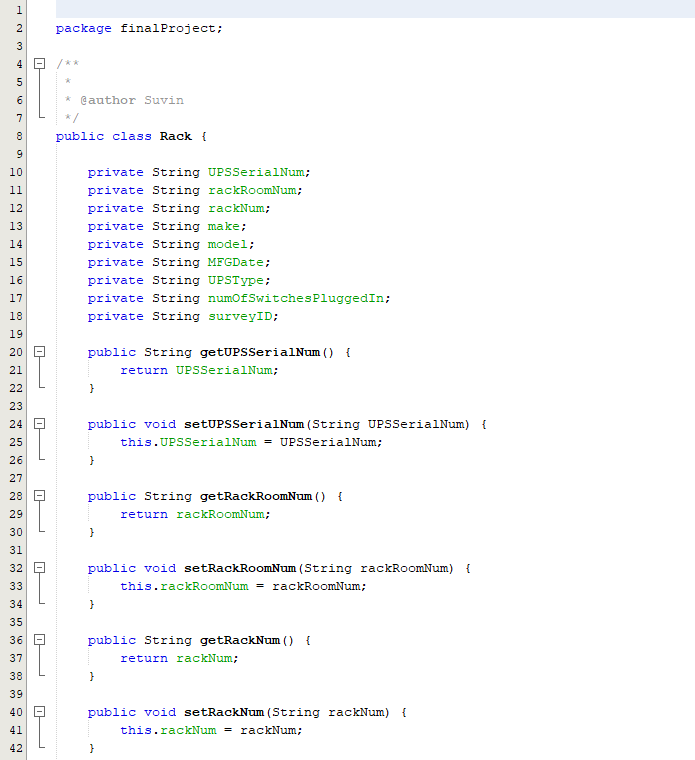
SurveyInfo (Accessors and Mutators):



Building (Accessors and Mutators):

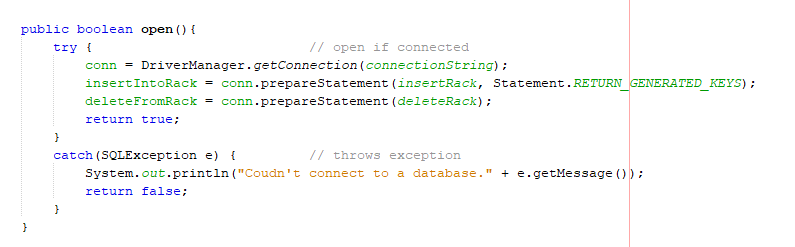


Rack (Accessors and Mutators):

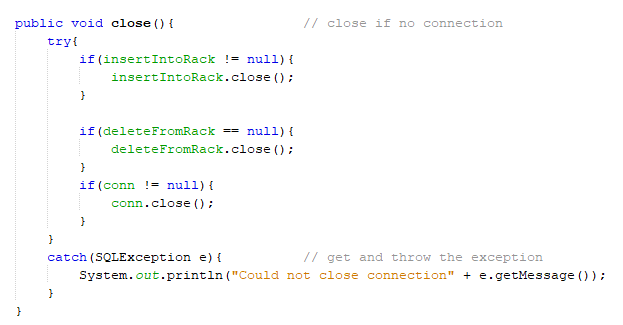


The class DataSource sets the connection string and opens the database to perform queries in it.

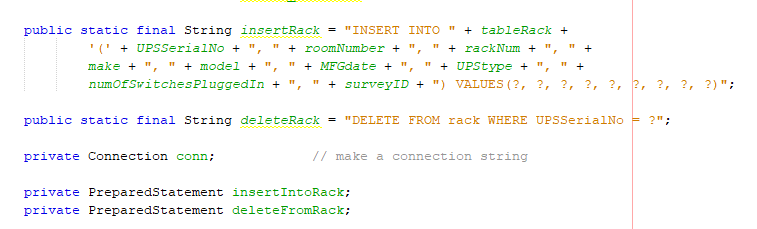




This class also helps close the database to avoid any form of data leak.

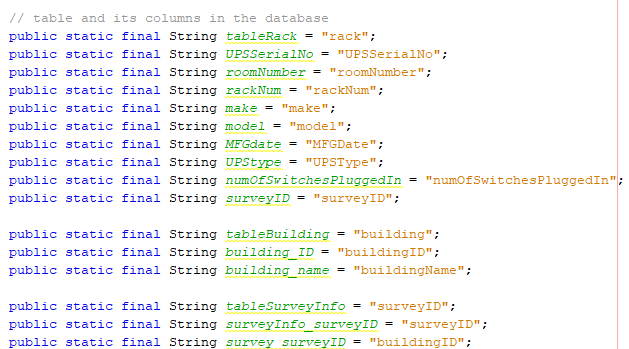


The database is vulnerable to hacking if normal SQL statements are used to code queries in Java. To avoid this, we use Prepared Statement. This is done using placeholder character where queries in SQL end with a placeholder character which is a “?”.



The most important part about the DataSource class is that it helps set up queries which is then later executed in Main.

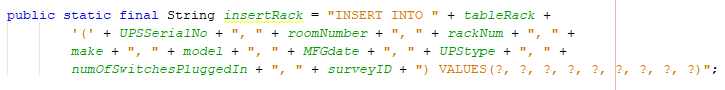
The first aspect is to create variables of all the table names and its attributes and then refer to these tables while designing the queries.

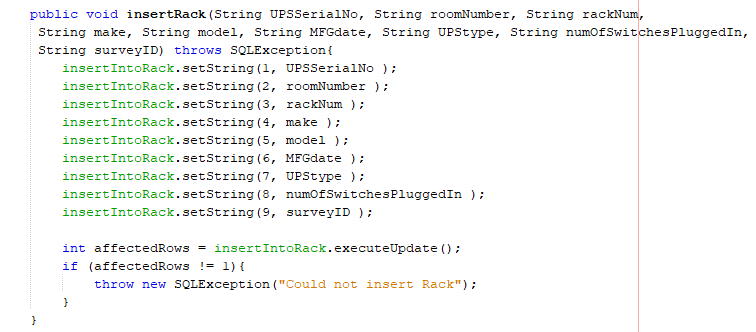


The first query used is the standard “SELECT \* FROM *tableName*”. This query is used to display all the information on SharePoint. To display the output of the attributes from the table. The rows are stored using a List data structure. Note: To execute the query, it is important to first connect to the database. This is possible by using try and catch where we try getting a connection string using Statement library in java. It is a library used for creating a static SQL statement and returning the results it produces.



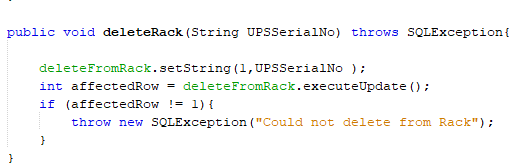
The next query coded is the insert query. This is how a typical query in SQL looks like “INSET INTO *tableName* VALUES (*value1*, *value2*, …,*etc*) ”. Notice the use of placeholder’s in string insertRack. The number of placeholder character directly depends on the number of inputs given by the user.



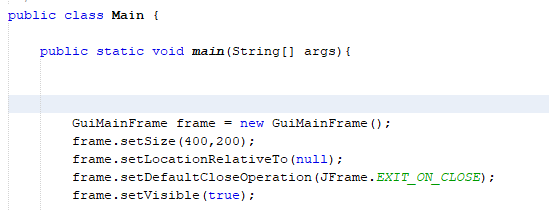


The third and the final query would be the delete query. Delete query in SQL: “DELETE FROM *tableName* WHERE *condition*”. As UPSSerialNum is the primary key, it is treated as the id and is used in the condition.

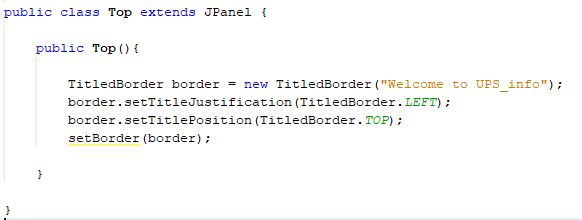




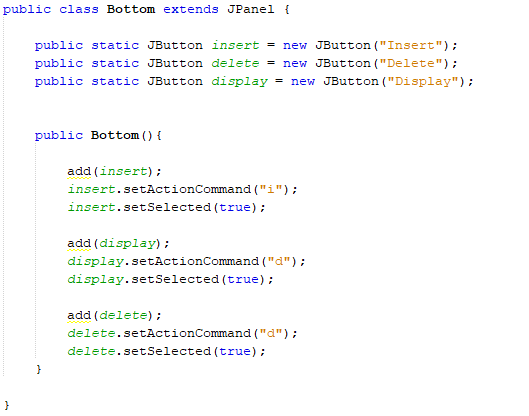
The second part of the project is to use buttons to call the queries in main by creating a GUI using JFrame and JPanel. A GUI has three main components. The first is the border and the size of the frame where the user interacts with buttons. The size should not be too large or too small and hence 400 × 200 would be the ideal size. The frame size is set in main.



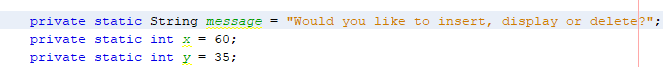
The second aspect of GUI should be the intro message as to what it does. This was achieved by using a class called bottom where I used TitledBorder, a part of java swing library to include the message “Welcome to UPS\_info” every time this frame popped up. It as class will be later inherited, it is important to override the object created in this class.



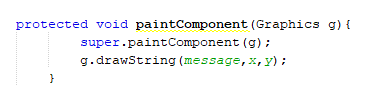
Class Bottom would have all the buttons which are used to execute all the queries in main. This class inherits JPanel to create three buttons named Insert, Delete and Display.



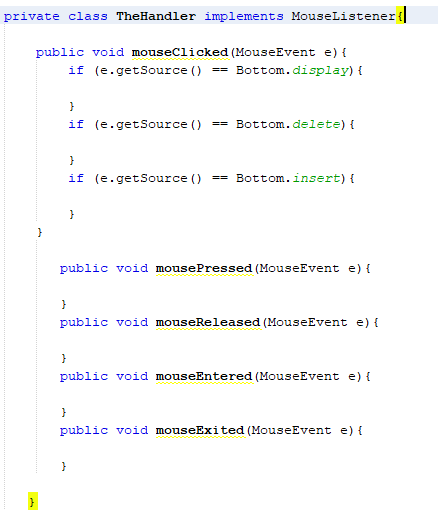
The last step of the process is to create a handler class which forms a connection between these three buttons and the three queries. This is done in class Mid. The first part is to display a string which asks user which query they would like to perform. This string is also supposed to be positioned appropriately to make it look professional. This is done by setting its x and y co-ordinates.



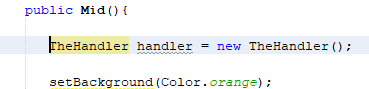
Class Mid has one method, one constructor and a handler Class. The method is the paint component which is used to repaint the background with the message on it.

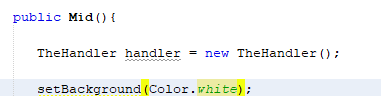


The second class is the handler class. This is used to implement mouseListener. Mouse Listener is used in java when the user uses mouse to interact with java GUI.



In the constructor, first an handler object is created and then the color of the background is set.

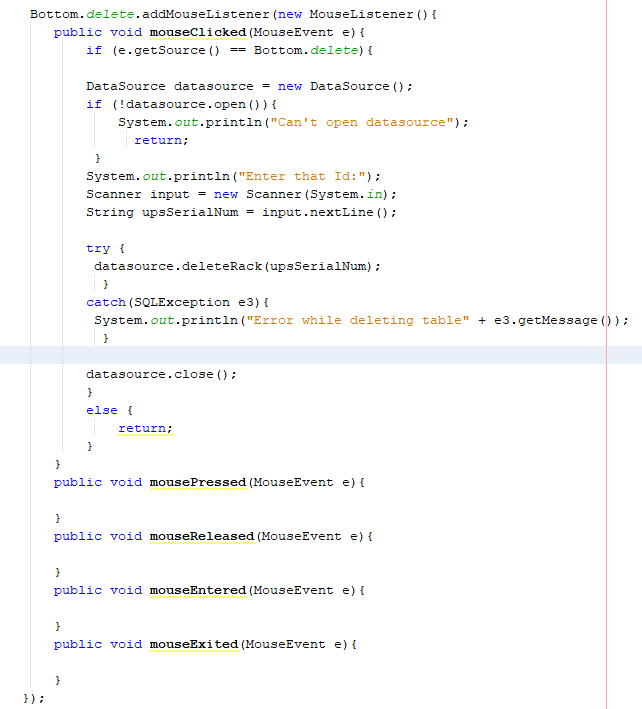




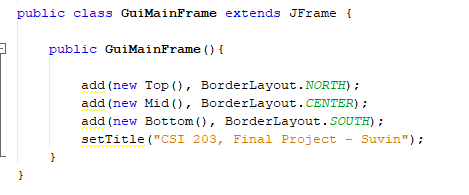
After creating the handler object, we use it with the display button which executes the select query and display the output in the console.



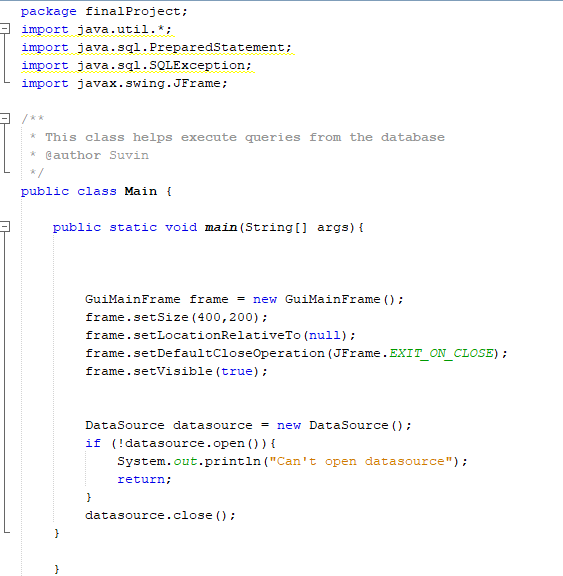
We repeat the same for insert and delete buttons.



The classes Top, Mid and Bottom are overridden in GuiMainFrame class as shown below. This class inherits from JFrame and the title of the frame is also set in this class.

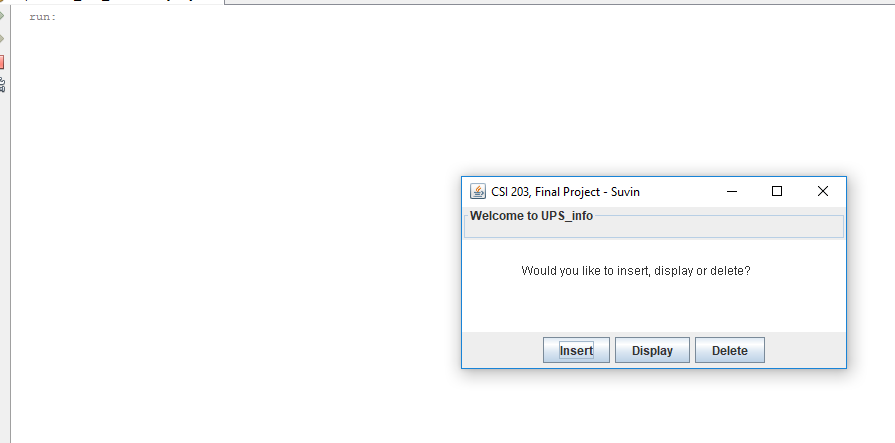


We use Main class to create a new GuiMainFrame object and open the database to execute and deploy all the queries in the database.

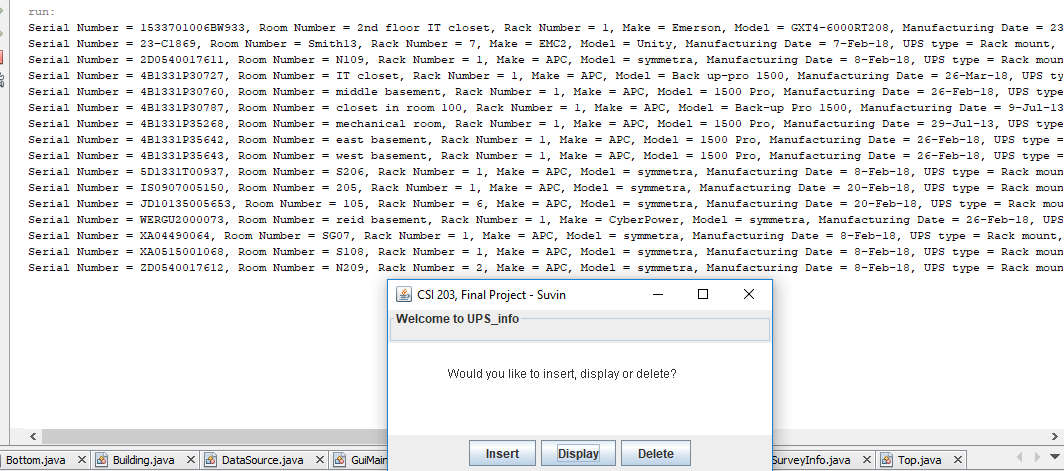


**Testing:**

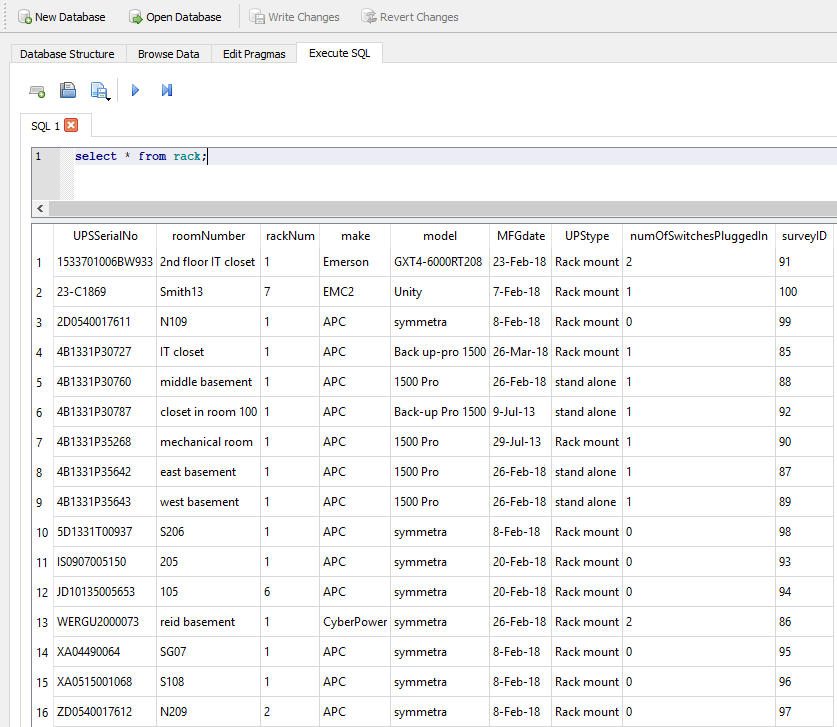
This is how the GUI looks like when the program is run.



When we click on the display button, we can get all the data from table rack on the database.

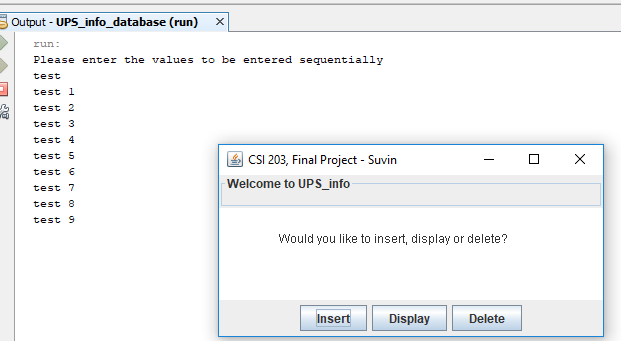


To make sure that the values displayed here are the same on the database. Let’s run the select statement directly on the database.

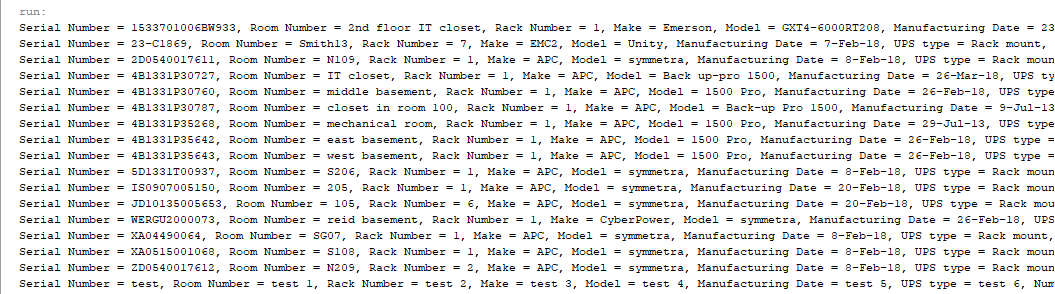


As shown above, the values are same which means the program is successfully using data from the database.

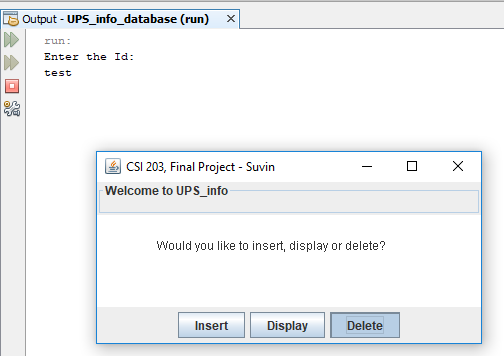
Clicking on the insert button will allow user to enter information using the insert button. For this demonstration, lets insert the following values on the database.



To make sure that the values are inserted on the database, using display button again will give us the updated database. As shown on the screenshot below, the last row has been updates with the values entered above by the user.



To demonstrate the delete button, the test row which was added would be deleted.



Using the display button again to demonstrate the results gives us the following values. As shown below, the row with test values were successfully deleted.

