Juan Gonzalez

ITMD 411

**Dao.java**

**package** javaapplication1;

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.time.LocalDateTime;

**import** java.time.format.DateTimeFormatter;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** Dao {

// instance fields

**static** Connection *connect* = **null**;

Statement statement = **null**;

// constructor

**public** Dao() {

}

**public** Connection getConnection() {

// Setup the connection with the DB

**try** {

*connect* = DriverManager

.*getConnection*(ConnectionURL.*url*);

// connect = DriverManager

// .getConnection("jdbc:mysql://localhost/test", "root", "4111");

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** *connect*;

}

// CRUD implementation

**public** **void** createTables() {

// variables for SQL Query table creations

**final** String createTicketsTable = "CREATE TABLE jgonz\_tickets(tid INT AUTO\_INCREMENT PRIMARY KEY, ticket\_desc VARCHAR(200), ticket\_issuer VARCHAR(30), status VARCHAR(10), start\_date VARCHAR(10), end\_date VARCHAR(10))";

**final** String createUsersTable = "CREATE TABLE jgonz\_users(uid INT AUTO\_INCREMENT PRIMARY KEY, uname VARCHAR(30), upass VARCHAR(30), admin int)";

**final** String createLoginsTable = "create table jgonz\_logins(uid INT, loginday varchar(20), logintime varchar(20));";

**try** {

// execute queries to create tables

statement = getConnection().createStatement();

statement.executeUpdate(createLoginsTable);

statement.executeUpdate(createTicketsTable);

statement.executeUpdate(createUsersTable);

System.***out***.println("Created tables in given database...");

// end create table

// close connection/statement object

statement.close();

*connect*.close();

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}

// add users to user table

// addUsers();

}

**public** **void** addUsers() {

// add list of users from userlist.csv file to users table

// variables for SQL Query inserts

String sql;

Statement statement;

BufferedReader br;

List<List<String>> array = **new** ArrayList<>(); // list to hold (rows & cols)

// read data from file

**try** {

br = **new** BufferedReader(**new** FileReader(**new** File("./userlist.csv")));

String line;

**while** ((line = br.readLine()) != **null**) {

array.add(Arrays.*asList*(line.split(",")));

}

} **catch** (Exception e) {

System.***out***.println("There was a problem loading the file");

}

**try** {

// Setup the connection with the DB

statement = getConnection().createStatement();

// create loop to grab each array index containing a list of values

// and PASS (insert) that data into your User table

**for** (List<String> rowData : array) {

sql = "insert into jgonz\_users(uname,upass,admin) " + "values('" + rowData.get(0) + "'," + " '"

+ rowData.get(1) + "','" + rowData.get(2) + "');";

statement.executeUpdate(sql);

}

System.***out***.println("Inserts completed in the given database...");

// close statement object

statement.close();

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}

}

**public** **int** insertRecords(String ticketName, String ticketDesc, String status, String startDate, String endDate) {

**int** id = 0;

**try** {

statement = getConnection().createStatement();

statement.executeUpdate("Insert into jgonz\_tickets" + "(ticket\_issuer, ticket\_desc, status, start\_date, end\_date) values(" + " '"

+ ticketName + "','" + ticketDesc + "','" + status + "','" + startDate + "','" + endDate + "');" , Statement.***RETURN\_GENERATED\_KEYS***);

// retrieve ticket id number newly auto generated upon record insertion

ResultSet resultSet = **null**;

resultSet = statement.getGeneratedKeys();

**if** (resultSet.next()) {

// retrieve first field in table

id = resultSet.getInt(1);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** id;

}

**public** ResultSet readRecords(**boolean** isAdmin, String userName) {

ResultSet results = **null**;

**try** {

statement = *connect*.createStatement();

**if**(isAdmin) {

results = statement.executeQuery("SELECT \* FROM jgonz\_tickets");

} **else** {

results = statement.executeQuery("SELECT \* FROM jgonz\_tickets WHERE ticket\_issuer = '" +userName+ "'");

}

//connect.close();

} **catch** (SQLException e1) {

e1.printStackTrace();

}

**return** results;

}

//update ticket

**public** **int** updateTicket(**int** ticketID, String ticketDesc, String ticketIssr, String status, String startDate, String endDate) {

String sql = "update jgonz\_tickets set ticket\_desc = '" + ticketDesc + "', ticket\_issuer = '" +ticketIssr + "', status = '"+ status + "', start\_date = '" +startDate+ "', end\_date = '" +endDate+"' where tid = "+ticketID+";";

**int** returnValue = -1;

**try** {

statement = *connect*.createStatement();

returnValue = statement.executeUpdate(sql);

**if** (returnValue == 1) {

System.***out***.println("Ticket ID " + ticketID + " exists and was modified.");

} **else** {

System.***out***.println("Error.");

}

statement.close();

} **catch** (SQLException e1) {

e1.printStackTrace();

}

**return** returnValue;

}

**public** String[] getTicketData(**int** ticketID) **throws** SQLException {

ResultSet results = **null**;

**try** {

statement = *connect*.createStatement();

results = statement.executeQuery("select \* from jgonz\_tickets where tid = "+ticketID+";");

//connect.close();

} **catch** (SQLException e1) {

e1.printStackTrace();

}

results.next();

String[] data = **new** String[7];

**for** (**int** i=1; i<=6; i++) {

data[i] = results.getString(i);

}

**return** data;

}

//delete ticket

**public** **int** deleteTicket(**int** ticketID) {

String sql = "delete from jgonz\_tickets where tid = "+ticketID+ ";";

**int** returnValue = -1;

**try** {

statement = *connect*.createStatement();

returnValue = statement.executeUpdate(sql);

**if** (returnValue == 1) {

System.***out***.println("Ticket ID " + ticketID + " exists and was deleted.");

} **else** {

System.***out***.println("Ticket ID " + ticketID + " does not exist.");

}

statement.close();

} **catch** (SQLException e1) {

e1.printStackTrace();

}

**return** returnValue;

}

**public** String getCurrentDay() {

DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("MM/dd/YYYY");

LocalDateTime now = LocalDateTime.*now*();

**return** dtf.format(now);

}

**public** String getMonthFromToday() {

DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("MM/dd/YYYY");

LocalDateTime now = LocalDateTime.*now*();

now = now.plusMonths(1);

**return** dtf.format(now);

}

**public** String getCurrentDayTime() {

DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("MM/dd/YYYY HH:mm:ss");

LocalDateTime now = LocalDateTime.*now*();

**return** dtf.format(now);

}

**public** **int** closeTicket(**int** ticketID, String userName, **boolean** isAdmin) {

**if** (isAdmin) {

System.***out***.println("Ticket closing was requested by an admin.");

} **else** {

System.***out***.println("Ticket closing was requested by a regular user.");

}

String sqlRegular = "update jgonz\_tickets set status = 'Closed' where tid = " + ticketID + " and ticket\_issuer = '" + userName + "';";

String sqlAdmin = "update jgonz\_tickets set status = 'Closed' where tid = " + ticketID + ";";

**int** returnValue = -1;

**try** {

statement = getConnection().createStatement();

**if** (isAdmin) {

returnValue = statement.executeUpdate(sqlAdmin);

} **else** {

returnValue = statement.executeUpdate(sqlRegular);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**if**(returnValue == 1) {

System.***out***.println("Ticket closed successfully.");

} **else** {

System.***out***.println("Ticket was not closed successfully.");

}

**return** returnValue;

}

**public** **int** createUser(String userName, String userPass, String isAdmin) {

**int** returnValue = -1;

**try** {

statement = getConnection().createStatement();

returnValue = statement.executeUpdate("Insert into jgonz\_users" + "(uname, upass, admin) values(" + " '"

+ userName + "','" + userPass + "','" + isAdmin + "');" , Statement.***RETURN\_GENERATED\_KEYS***);

} **catch** (SQLException e) {

e.printStackTrace();

}

**if** (returnValue == 0) {

System.***out***.println("User create failed.");

} **else** {

System.***out***.println("User create success.");

}

**return** returnValue;

}

**public** **int** logTime(**int** userID) {

//islogin true = login

//islogin false = logout

String timeFull = getCurrentDayTime();

String sql;

String day = timeFull.substring(0,10);

String time = timeFull.substring(11,19);

sql = "insert into jgonz\_logins(uid, loginday, logintime) values (" + userID + ", '" + day + "', '" + time + "')";

**int** returnValue = -1;

**try** {

statement = *connect*.createStatement();

returnValue = statement.executeUpdate(sql);

statement.close();

} **catch** (SQLException e1) {

e1.printStackTrace();

}

**return** returnValue;

}

**public** ResultSet readLogins(**int** userID) {

String sqlRegular = "SELECT \* FROM jgonz\_logins";

String sqlSearch = "select \* from jgonz\_logins where uid = " + userID + ";";

ResultSet results = **null**;

**try** {

statement = *connect*.createStatement();

**if**(userID == -1) {

results = statement.executeQuery(sqlRegular);

} **else** {

results = statement.executeQuery(sqlSearch);

}

//connect.close();

} **catch** (SQLException e1) {

e1.printStackTrace();

}

**return** results;

}

**public** ResultSet getTicketResult(**int** ticketID, String userName, **boolean** isAdmin) {

String sql = "";

**if** (isAdmin) {

sql = "select \* from jgonz\_tickets where tid = "+ticketID+";";

} **else** {

sql = "select \* from jgonz\_tickets where tid = "+ticketID+" and ticket\_issuer = '" + userName + "';";

}

ResultSet results = **null**;

**try** {

statement = *connect*.createStatement();

results = statement.executeQuery(sql);

//connect.close();

} **catch** (SQLException e1) {

e1.printStackTrace();

}

**return** results;

}

}

**Login.java**

**package** javaapplication1;

**import** java.awt.GridLayout; //useful for layouts

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

//controls-label text fields, button

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPasswordField;

**import** javax.swing.JTextField;

@SuppressWarnings("serial")

**public** **class** Login **extends** JFrame {

Dao conn;

**public** Login() {

**super**("IIT Help Desk Login");

conn = **new** Dao();

conn.createTables();

setSize(400, 210);

setLayout(**new** GridLayout(4, 2));

setLocationRelativeTo(**null**); // centers window

// SET UP CONTROLS

JLabel lblUsername = **new** JLabel("Username", JLabel.***LEFT***);

JLabel lblPassword = **new** JLabel("Password", JLabel.***LEFT***);

JLabel lblStatus = **new** JLabel(" ", JLabel.***CENTER***);

// JLabel lblSpacer = new JLabel(" ", JLabel.CENTER);

JTextField txtUname = **new** JTextField(10);

JPasswordField txtPassword = **new** JPasswordField();

JButton btn = **new** JButton("Submit");

JButton btnExit = **new** JButton("Exit");

// constraints

lblStatus.setToolTipText("Contact help desk to unlock password");

lblUsername.setHorizontalAlignment(JLabel.***CENTER***);

lblPassword.setHorizontalAlignment(JLabel.***CENTER***);

// ADD OBJECTS TO FRAME

add(lblUsername); // 1st row filler

add(txtUname);

add(lblPassword); // 2nd row

add(txtPassword);

add(btn); // 3rd row

add(btnExit);

add(lblStatus); // 4th row

btn.addActionListener(**new** ActionListener() {

**int** count = 0; // count agent

@Override

**public** **void** actionPerformed(ActionEvent e) {

**boolean** admin = **false**;

count = count + 1;

// verify credentials of user (MAKE SURE TO CHANGE TO YOUR TABLE NAME BELOW)

String query = "SELECT \* FROM jgonz\_users WHERE uname = ? and upass = ?;";

**try** (PreparedStatement stmt = conn.getConnection().prepareStatement(query)) {

stmt.setString(1, txtUname.getText());

stmt.setString(2, txtPassword.~~getText~~());

ResultSet rs = stmt.executeQuery();

**if** (rs.next()) {

**int** userID = rs.getInt("uid");

conn.logTime(userID);

admin = rs.getBoolean("admin"); // get table column value

**new** Tickets(admin, rs.getString("uname"));

setVisible(**false**); // HIDE THE FRAME

dispose(); // CLOSE OUT THE WINDOW

} **else**

lblStatus.setText("Try again! " + (3 - count) + " / 3 attempts left");

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

});

btnExit.addActionListener(e -> System.*exit*(0));

setVisible(**true**); // SHOW THE FRAME

}

**public** **static** **void** main(String[] args) {

**new** Login();

}

}

**ticketsJTable.java**

**package** javaapplication1;

**import** java.sql.ResultSet;

**import** java.sql.ResultSetMetaData;

**import** java.sql.SQLException;

**import** java.util.Vector;

**import** javax.swing.table.DefaultTableModel;

**public** **class** ticketsJTable {

@SuppressWarnings("unused")

**private** **final** DefaultTableModel tableModel = **new** DefaultTableModel();

**public** **static** DefaultTableModel buildTableModel(ResultSet rs) **throws** SQLException {

ResultSetMetaData metaData = rs.getMetaData();

// names of columns

Vector<String> columnNames = **new** Vector<String>();

**int** columnCount = metaData.getColumnCount();

**for** (**int** column = 1; column <= columnCount; column++) {

columnNames.add(metaData.getColumnName(column));

}

// data of the table

Vector<Vector<Object>> data = **new** Vector<Vector<Object>>();

**while** (rs.next()) {

Vector<Object> vector = **new** Vector<Object>();

**for** (**int** columnIndex = 1; columnIndex <= columnCount; columnIndex++) {

vector.add(rs.getObject(columnIndex));

}

data.add(vector);

}

// return data/col.names for JTable

**return** **new** DefaultTableModel(data, columnNames);

}

}

**Tickets.java**

**package** javaapplication1;

**import** java.awt.Color;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.WindowAdapter;

**import** java.awt.event.WindowEvent;

**import** java.sql.SQLException;

**import** javax.swing.Box;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

@SuppressWarnings("serial")

**public** **class** Tickets **extends** JFrame **implements** ActionListener {

// class level member objects

Dao dao = **new** Dao(); // for CRUD operations

Boolean chkIfAdmin = **null**;

**private** String userName = **null**;

// Main menu object items

**private** JMenu mnuFile = **new** JMenu("File");

**private** JMenu mnuTickets = **new** JMenu("Tickets");

**private** JMenu mnuAdmin = **new** JMenu("Admin");

// Sub menu item objects for all Main menu item objects

JMenuItem mnuItemExit;

JMenuItem mnuItemUpdate;

JMenuItem mnuItemDelete;

JMenuItem mnuItemUserCreate;

JMenuItem mnuItemOpenTicket;

JMenuItem mnuItemViewTicket;

JMenuItem mnuItemViewTicketbyID;

JMenuItem mnuItemCloseTicket;

JMenuItem mnuItemLoginReport;

**public** Tickets(Boolean isAdmin, String userName) {

**super**(userName + " logged in!");

**this**.userName = userName;

chkIfAdmin = isAdmin;

createMenu();

prepareGUI();

}

**private** **void** createMenu() {

/\* Initialize sub menu items \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// initialize sub menu item for File main menu

mnuItemExit = **new** JMenuItem("Exit");

// add to File main menu item

mnuFile.add(mnuItemExit);

// initialize first sub menu items for Admin main menu

mnuItemUpdate = **new** JMenuItem("Update Ticket");

// add to Admin main menu item

mnuAdmin.add(mnuItemUpdate);

// initialize second sub menu items for Admin main menu

mnuItemDelete = **new** JMenuItem("Delete Ticket");

// add to Admin main menu item

mnuAdmin.add(mnuItemDelete);

// initialize create user for Admin main menu

mnuItemUserCreate = **new** JMenuItem("Create User");

// add to Admin main menu item

mnuAdmin.add(mnuItemUserCreate);

mnuItemLoginReport = **new** JMenuItem("Login Report");

mnuAdmin.add(mnuItemLoginReport);

// initialize first sub menu item for Tickets main menu

mnuItemOpenTicket = **new** JMenuItem("Create Ticket");

// add to Ticket Main menu item

mnuTickets.add(mnuItemOpenTicket);

// initialize second sub menu item for Tickets main menu

mnuItemViewTicket = **new** JMenuItem("View Ticket");

// add to Ticket Main menu item

mnuTickets.add(mnuItemViewTicket);

mnuItemViewTicketbyID = **new** JMenuItem("View Ticket by ID");

// add to Ticket Main menu item

mnuTickets.add(mnuItemViewTicketbyID);

mnuItemCloseTicket = **new** JMenuItem("Close Ticket");

// add to Ticket Main menu item

mnuTickets.add(mnuItemCloseTicket);

// initialize any more desired sub menu items below

/\* Add action listeners for each desired menu item \*\*\*\*\*\*\*\*\*\*\*\*\*/

mnuItemExit.addActionListener(**this**);

mnuItemUpdate.addActionListener(**this**);

mnuItemDelete.addActionListener(**this**);

mnuItemUserCreate.addActionListener(**this**);

mnuItemOpenTicket.addActionListener(**this**);

mnuItemViewTicket.addActionListener(**this**);

mnuItemViewTicketbyID.addActionListener(**this**);

mnuItemCloseTicket.addActionListener(**this**);

mnuItemLoginReport.addActionListener(**this**);

}

**private** **void** prepareGUI() {

// create JMenu bar

JMenuBar bar = **new** JMenuBar();

bar.add(mnuFile); // add main menu items in order, to JMenuBar

bar.add(mnuTickets);

**if**(chkIfAdmin) {

bar.add(mnuAdmin);

}

// add menu bar components to frame

setJMenuBar(bar);

addWindowListener(**new** WindowAdapter() {

// define a window close operation

**public** **void** windowClosing(WindowEvent wE) {

System.*exit*(0);

}

});

// set frame options

setSize(400, 400);

getContentPane().setBackground(Color.***LIGHT\_GRAY***);

setLocationRelativeTo(**null**);

setVisible(**true**);

}

@Override

**public** **void** actionPerformed(ActionEvent e) {

// implement actions for sub menu items

**if** (e.getSource() == mnuItemExit) {

System.*exit*(0);

} **else** **if** (e.getSource() == mnuItemOpenTicket) {

JTextField tdField = **new** JTextField(userName);

// get ticket information

String ticketName = JOptionPane.*showInputDialog*(tdField, "Enter your name");

String ticketDesc = JOptionPane.*showInputDialog*(**null**, "Enter a ticket description");

// insert ticket information to database

**int** id = dao.insertRecords(ticketName, ticketDesc, "Open", dao.getCurrentDay(), dao.getMonthFromToday());

// display results if successful or not to console / dialog box

**if** (id != 0) {

System.***out***.println("Ticket ID : " + id + " created successfully!!!");

JOptionPane.*showMessageDialog*(**null**, "Ticket id: " + id + " created");

} **else**

System.***out***.println("Ticket cannot be created!!!");

}

**else** **if** (e.getSource() == mnuItemViewTicket) {

// retrieve all tickets details for viewing in JTable

**try** {

// Use JTable built in functionality to build a table model and

// display the table model off your result set!!!

JTable jt = **new** JTable(ticketsJTable.*buildTableModel*(dao.readRecords(chkIfAdmin, userName)));

jt.setBounds(30, 40, 200, 400);

JScrollPane sp = **new** JScrollPane(jt);

add(sp);

setVisible(**true**); // refreshes or repaints frame on screen

} **catch** (SQLException e1) {

e1.printStackTrace();

}

} **else** **if** (e.getSource() == mnuItemDelete) {

String ticketIDstr = JOptionPane.*showInputDialog*(**null**, "Enter the ticket ID you wish to delete:");

**int** ticketID = Integer.*parseInt*(ticketIDstr);

**int** result = -1;

**int** reply = JOptionPane.*showConfirmDialog*(**null**, "Delete ticket ID " + ticketID + "?", "Delete Check", JOptionPane.***YES\_NO\_OPTION***);

**if** (reply == JOptionPane.***YES\_OPTION***) {

result = dao.deleteTicket(ticketID);

}

**if**(result == 1) {

JOptionPane.*showMessageDialog*(**null**, "Ticket ID " + ticketID + " deleted.");

} **else** **if** (result == -1) {

JOptionPane.*showMessageDialog*(**null**, "Deletion of Ticket ID " + ticketID + " was cancelled.");

}

**else** {

JOptionPane.*showMessageDialog*(**null**, "Ticket ID " + ticketID + " does not exist.");

}

} **else** **if** (e.getSource() == mnuItemUpdate) {

String ticketIDstr = JOptionPane.*showInputDialog*(**null**, "Enter the ticket ID you wish to update:");

**int** ticketID = Integer.*parseInt*(ticketIDstr);

String[] ticketData = **null**;

**try** {

ticketData = dao.getTicketData(ticketID);

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

JOptionPane.*showMessageDialog*(**null**, "Please enter a valid ticket ID.");

// e1.printStackTrace();

}

JTextField tdField = **new** JTextField(ticketData[2], 10);

JTextField tiField = **new** JTextField(ticketData[3], 10);

JTextField stField = **new** JTextField(ticketData[4], 10);

JTextField sdField = **new** JTextField(ticketData[5], 10);

JTextField edField = **new** JTextField(ticketData[6], 10);

JPanel myPanel = **new** JPanel();

myPanel.add(**new** JLabel("Ticket Description: "));

myPanel.add(tdField);

myPanel.add(Box.*createHorizontalStrut*(15));

myPanel.add(**new** JLabel("Ticket Issuer: "));

myPanel.add(tiField);

myPanel.add(Box.*createHorizontalStrut*(15));

myPanel.add(**new** JLabel("Ticket Status: "));

myPanel.add(stField);

myPanel.add(Box.*createHorizontalStrut*(15));

myPanel.add(**new** JLabel("Start Date: "));

myPanel.add(sdField);

myPanel.add(Box.*createHorizontalStrut*(15));

myPanel.add(**new** JLabel("End Date: "));

myPanel.add(edField);

myPanel.add(Box.*createHorizontalStrut*(15));

**int** result = JOptionPane.*showConfirmDialog*(**null**, myPanel,

"Editing ticket ID " + ticketID, JOptionPane.***OK\_CANCEL\_OPTION***);

**if** (result == JOptionPane.***OK\_OPTION***) {

**int** resultfromsql = dao.updateTicket(ticketID, tdField.getText(), tiField.getText(), stField.getText(), sdField.getText(), edField.getText());

System.***out***.println("Ticket ID " + ticketID + " modified.");

**if** (resultfromsql == 1) {

JOptionPane.*showMessageDialog*(**null**, "Ticket ID " + ticketID + " modified successfully.");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "There was an error in modifying Ticket ID " + ticketID + ".");

}

}

} **else** **if** (e.getSource() == mnuItemUserCreate) {

String newUserName = JOptionPane.*showInputDialog*(**null**, "Enter a user name:");

String newUserPass = JOptionPane.*showInputDialog*(**null**, "Enter a password:");

String newUserAdmin;

**int** reply = JOptionPane.*showConfirmDialog*(**null**, "Is this user an admin?", "Admin Check", JOptionPane.***YES\_NO\_OPTION***);

**if** (reply == JOptionPane.***YES\_OPTION***) {

newUserAdmin = "1";

} **else** {

newUserAdmin = "0";

}

**int** resultfromsql = dao.createUser(newUserName, newUserPass, newUserAdmin);

**if** (resultfromsql == 1) {

JOptionPane.*showMessageDialog*(**null**, "User was created successfully.");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "There was an error in creating the user.");

}

} **else** **if** (e.getSource() == mnuItemCloseTicket) {

String ticketIDstr = JOptionPane.*showInputDialog*(**null**, "Enter a ticket ID to close:");

**int** ticketID = Integer.*parseInt*(ticketIDstr);

**int** result = dao.closeTicket(ticketID, userName, chkIfAdmin);

**if**(result == 1) {

JOptionPane.*showMessageDialog*(**null**, "Ticket ID " + ticketID + " was closed.");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Ticket ID " + ticketID + " could not be closed.");

}

} **else** **if** (e.getSource() == mnuItemLoginReport) {

//retrieve login information

**int** reply = JOptionPane.*showConfirmDialog*(**null**, "Would you like to search a specific user's login times?", "Range Check", JOptionPane.***YES\_NO\_OPTION***);

**int** userID = -1;

**if** (reply == JOptionPane.***YES\_OPTION***) {

String userIDstr = JOptionPane.*showInputDialog*(**null**, "Enter a user ID to search: ");

userID = Integer.*parseInt*(userIDstr);

}

**try** {

// Use JTable built in functionality to build a table model and

// display the table model off your result set!!!

JTable jt = **new** JTable(ticketsJTable.*buildTableModel*(dao.readLogins(userID)));

jt.setBounds(30, 40, 200, 400);

JScrollPane sp = **new** JScrollPane(jt);

add(sp);

setVisible(**true**); // refreshes or repaints frame on screen

} **catch** (SQLException e1) {

e1.printStackTrace();

}

} **else** **if** (e.getSource() == mnuItemViewTicketbyID) {

String ticketIDstr = JOptionPane.*showInputDialog*(**null**, "Enter a ticket ID to view:");

**int** ticketID = Integer.*parseInt*(ticketIDstr);

**try** {

JTable jt = **new** JTable(ticketsJTable.*buildTableModel*(dao.getTicketResult(ticketID, userName, chkIfAdmin)));

jt.setBounds(30, 40, 200, 400);

JScrollPane sp = **new** JScrollPane(jt);

add(sp);

setVisible(**true**); // refreshes or repaints frame on screen

} **catch** (SQLException e1) {

e1.printStackTrace();

}

}

/\*

\* continue implementing any other desired sub menu items (like for update and

\* delete sub menus for example) with similar syntax & logic as shown above

\*/

}

}

**ConnectionURL.java:**

**package** javaapplication1;

**public** **class** ConnectionURL {

**public** **static** String *url* = "redacted";

}