|  |  |  |  |
| --- | --- | --- | --- |
| **CSY2030 Systems Design & Development**  **Assignment 2** | | | |
| **Assignment title:** | CSY2030 Individual Project to create a GUI based National Property Sales System | | |
| **Weighting:** | 50% | | |
| **Date of Issue:** | 15th February, 2020 | **Submission:** | 3th May, 2020 |
| **Student Name:** | Narukkottil Hameed Mohammed Unais | | |
| **Student ID:** | 18400985 | | |
| **Video Link:** | <https://northampton.mediaspace.kaltura.com/media/Unais+Narukkottil-18400985-SDD-AS2-video-file/0_4m6jjb2l> | | |
| **Assessment Criteria:** | | | |
| Design of System (Use Case Diagram, Class Diagram) - 15% | | | |
| Implementation - 45% | | | |
| Testing (black box and white box testing) - 15% | | | |
| Code Layout – 5% | | | |
| Quality of Report – 10% | | | |
| Demonstration – 10 | | | |

package com.unais;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

/\*\*

\* @author: Narukkottil Hameed Mohammed Unais

\* Student ID: 18400985

\* Module Title: Systems Design & Development

\* Module Code: CSY2030

\* Assignment Objective: Design, implement and test a stand-alone Property Sales system using object-oriented principles in Java.

\* Weighting: 50%

\* Deadline: 3rd May 2020

\*/

public class Main {

Connection con;

public static void main(String[] args){

createConnection();//will run the createConnection method;

}

public static void createConnection() {

try {//The following code connects the Java application to the mysql server

String serverName;

serverName = "localhost:3306";//The server name

String database;

database = "pms";//The database name

String databaseUserName;

databaseUserName = "root";//username

String databasePassword;

databasePassword = "";//password

Class.forName("com.mysql.cj.jdbc.Driver");

try {//username and password will be replace using the variable

Connection con = DriverManager.getConnection("jdbc:mysql://"+serverName+"/"+database+""

+ "?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", ""+databaseUserName+"", ""+databasePassword+"");

loginifconnected();//Will run the login if connected method and will open the login page if localhost connection is active and mysql server acan be accessed

} catch (SQLException ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Mysql not Connected.");//if connection is not available the exception will caught and shows thsi dialoge box.

}

} catch (ClassNotFoundException ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Mysql not Connected.");//if connection is not available the exception will caught and shows thsi dialoge box.

}

}

public static void loginifconnected(){//will open the Login page if database is conected and the mysql database exist.

Login loginModel = new Login();

loginModel.setVisible(true);

}

}

package com.unais;

import java.awt.HeadlessException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.UIManager;

import javax.swing.UnsupportedLookAndFeelException;

public class Login extends javax.swing.JFrame {

Connection con;

public Login() {

initComponents();//initialize GUI components.

createConnection();//Excecutes the createConnection() Method

lookandfeel();// Executes The lookandfeel() Method

}

private void createConnection() {// This will create a connection to the database.

try {

String serverName;

serverName = "localhost:3306";

String database;

database = "pms";

String databaseUserName;

databaseUserName = "root";

String databasePassword;

databasePassword = "";

Class.forName("com.mysql.cj.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://"+serverName+"/"+database+""

+ "?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", ""+databaseUserName+"", ""+databasePassword+"");

}

catch (SQLException | ClassNotFoundException ex) {

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, ex);

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLoginpanel = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jUsernameTextLogin = new javax.swing.JTextField();

jPasswordTextLogin = new javax.swing.JPasswordField();

jExitButtonLogin = new javax.swing.JButton();

jLoginButtonLogin = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setResizable(false);

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel1.setText("Login Page");

jLabel2.setText("Usename");

jLabel3.setText("Password");

jExitButtonLogin.setText("Exit");

jExitButtonLogin.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jExitButtonLoginActionPerformed(evt);

}

});

jLoginButtonLogin.setText("Login");

jLoginButtonLogin.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jLoginButtonLoginActionPerformed(evt);

}

});

javax.swing.GroupLayout jLoginpanelLayout = new javax.swing.GroupLayout(jLoginpanel);

jLoginpanel.setLayout(jLoginpanelLayout);

jLoginpanelLayout.setHorizontalGroup(

jLoginpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jLoginpanelLayout.createSequentialGroup()

.addContainerGap()

.addGroup(jLoginpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 382, Short.MAX\_VALUE)

.addGroup(jLoginpanelLayout.createSequentialGroup()

.addGroup(jLoginpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jExitButtonLogin, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jLoginpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jUsernameTextLogin)

.addComponent(jPasswordTextLogin)

.addComponent(jLoginButtonLogin, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))))

.addContainerGap())

);

jLoginpanelLayout.setVerticalGroup(

jLoginpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jLoginpanelLayout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 88, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(jLoginpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jUsernameTextLogin, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jLoginpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3)

.addComponent(jPasswordTextLogin, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jLoginpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jExitButtonLogin)

.addComponent(jLoginButtonLogin))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLoginpanel, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLoginpanel, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jLoginButtonLoginActionPerformed(java.awt.event.ActionEvent evt) {

try {

String username = jUsernameTextLogin.getText();//gets username from the login page

String password = jPasswordTextLogin.getText();//gets the password from the login page

String userDetail = ""+username+password+"";//pusts username and password in one String variable for comparison.

if("".equals(username) || "".equals(password)){//checks if the field is empty or not

JOptionPane.showMessageDialog(null, "Please fill in the forms");//if empty shows this dialog box

}else if (username.equals("admin") && password.equals("admin123")) {//Checks if the username and password maches the administrators username and password

JOptionPane.showMessageDialog(null, "Login as Administrator successful");//if true shows that the administer was successfull inlogginf in

Administrator administratorModel = new Administrator();

administratorModel.setVisible(true);//makes the Administrator jFrame visible

this.setVisible(false);//and Login one invisble

}

else {//if the username and password not equal to administrators next thing it will do is to check in the database if it belongs to any of the userss.

try {

Statement stmt = con.createStatement();

ResultSet checkUser = stmt.executeQuery("SELECT \* FROM branch WHERE username = '"+username+"'");//gets data from branch.

if(checkUser.next()){//Stores username and password in a string

String likeUsername = checkUser.getString("username");

String likePassword = checkUser.getString("password");

stmt.close();

String getUserDetail = ""+likeUsername+likePassword+"";//pusts username and passwoord from the database together to compare

if (getUserDetail.equals(userDetail)){//if the username and are corrects logisthe user in

JOptionPane.showMessageDialog(null, "Logged in as "+username+"");// Shows the user as logged in with the corresponding isername from the username text box.

Secretary propertyManagementModel = new Secretary(username);//opens the secretary Jframe with the username as a string together.

propertyManagementModel.setVisible(true);//will open Secretary.java

this.setVisible(false);//Will close login

}else {//This else statement will only be executed when the password is wrong as the username exist.

JOptionPane.showMessageDialog(null, "Password incorrect.");

}

}else {//this is only executed is the sql statement recieves no value which means the user deosn't exist

JOptionPane.showMessageDialog(null, "Username doesn't exist.");

}

}

catch (SQLException | HeadlessException e){

Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, e);

JOptionPane.showMessageDialog(null, "connection cannot be established");//shows this error if the database lost its connection.

}

}

}

catch(NullPointerException e) {

JOptionPane.showMessageDialog(null, "Error Logging in");//shows error logging in if any other problems are encountered.

}

}

private void jExitButtonLoginActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);//Exits the system if the exit button is pressed.

}

public void lookandfeel(){

try {

// Set System look and feel for the program.

UIManager.setLookAndFeel(

UIManager.getSystemLookAndFeelClassName());

}

catch (UnsupportedLookAndFeelException | ClassNotFoundException | InstantiationException | IllegalAccessException e) {

JOptionPane.showMessageDialog(null, "Cannot load the system UI");//Shows this error if UI manager fails to load system UI.

}

}

// Variables declaration - do not modify

private javax.swing.JButton jExitButtonLogin;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JButton jLoginButtonLogin;

private javax.swing.JPanel jLoginpanel;

private javax.swing.JPasswordField jPasswordTextLogin;

private javax.swing.JTextField jUsernameTextLogin;

// End of variables declaration

}

package com.unais;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class Administrator extends javax.swing.JFrame {

DefaultTableModel branchTable;//declares a new varaible for branchtable as tablemodel

Connection con;

public Administrator() {

initComponents();

createConnection();//will create connection to the database

displayBranches();//will run display branches method so all added branches show up when opened.

}

private void displayBranches() {//This method will display all of the branches in the table after it gets it from the branch table in database

branchTable = (DefaultTableModel) jBranchTableAdmin.getModel();//assigns table to the variable

try {

Statement stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM branch");//gets all the details from the branch

while (result.next()){//puts all the data in a string using a loop

String branchUsername = result.getString("username");

String branchName = result.getString("name");

String branchAddress = result.getString("address");

String branchPhone = result.getString("phone");

String branchEmail = result.getString("email");

String branchWeb = result.getString("web\_address");

branchTable.addRow(new Object[]{branchUsername, branchName, branchAddress, branchPhone, branchEmail, branchWeb});//as the loops goes it will insert all the data in the Table model

}

stmt.close();

} catch (SQLException ex) {

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void createConnection() {

try {//creates the connection to server

String serverName;

serverName = "localhost:3306";

String database;

database = "pms";

String databaseUserName;

databaseUserName = "root";

String databasePassword;

databasePassword = "";

Class.forName("com.mysql.cj.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://"+serverName+"/"+database+""

+ "?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC&allowMultiQueries=true", ""+databaseUserName+"", ""+databasePassword+"");

}

catch (SQLException | ClassNotFoundException ex) {

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Mysql not Connected.");//if connection is not available the exception will caught and shows thsi dialoge box.

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel13 = new javax.swing.JPanel();

jLabel27 = new javax.swing.JLabel();

jLabel28 = new javax.swing.JLabel();

jUsernameTextAdmin = new javax.swing.JTextField();

jLabel29 = new javax.swing.JLabel();

jPasswordTextAdmin = new javax.swing.JPasswordField();

jLabel30 = new javax.swing.JLabel();

jNameTextAdmin = new javax.swing.JTextField();

jLabel31 = new javax.swing.JLabel();

jScrollPane5 = new javax.swing.JScrollPane();

jAddressTextAdmin = new javax.swing.JTextArea();

jPhoneNumAdmin = new javax.swing.JTextField();

jLabel32 = new javax.swing.JLabel();

jLabel33 = new javax.swing.JLabel();

jEmailTextAdmin = new javax.swing.JTextField();

jLabel34 = new javax.swing.JLabel();

jWebTextAdmin = new javax.swing.JTextField();

jAddButtonAdmin = new javax.swing.JButton();

jSeparator4 = new javax.swing.JSeparator();

jLogoutButtonAdmin = new javax.swing.JButton();

jExitButtonAdmin = new javax.swing.JButton();

jPanel14 = new javax.swing.JPanel();

jScrollPane6 = new javax.swing.JScrollPane();

jBranchTableAdmin = new javax.swing.JTable();

jDeleteButtunAdmin = new javax.swing.JButton();

jModifyButtonAdmin = new javax.swing.JButton();

jLabel19 = new javax.swing.JLabel();

jSeparator3 = new javax.swing.JSeparator();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setMinimumSize(new java.awt.Dimension(606, 580));

jPanel13.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel27.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

jLabel27.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel27.setText("Add Branch");

jLabel28.setText("Username");

jLabel29.setText("Password");

jLabel30.setText("Name");

jLabel31.setText("Address");

jAddressTextAdmin.setColumns(20);

jAddressTextAdmin.setRows(5);

jScrollPane5.setViewportView(jAddressTextAdmin);

jLabel32.setText("Phone number");

jLabel33.setText("E-mail address");

jLabel34.setText("Web Address");

jAddButtonAdmin.setText("ADD");

jAddButtonAdmin.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jAddButtonAdminActionPerformed(evt);

}

});

jLogoutButtonAdmin.setText("Logout");

jLogoutButtonAdmin.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jLogoutButtonAdminActionPerformed(evt);

}

});

jExitButtonAdmin.setText("Exit");

jExitButtonAdmin.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jExitButtonAdminActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel13Layout = new javax.swing.GroupLayout(jPanel13);

jPanel13.setLayout(jPanel13Layout);

jPanel13Layout.setHorizontalGroup(

jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel13Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jSeparator4)

.addComponent(jLabel28, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel27, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel29, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jUsernameTextAdmin)

.addComponent(jNameTextAdmin, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel30, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPasswordTextAdmin)

.addComponent(jLabel31, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane5, javax.swing.GroupLayout.DEFAULT\_SIZE, 188, Short.MAX\_VALUE)

.addComponent(jLabel32, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPhoneNumAdmin)

.addComponent(jLabel33, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jEmailTextAdmin)

.addComponent(jLabel34, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jWebTextAdmin)

.addComponent(jAddButtonAdmin, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLogoutButtonAdmin, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jExitButtonAdmin, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

jPanel13Layout.setVerticalGroup(

jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel13Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel27, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel28)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jUsernameTextAdmin, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel29)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPasswordTextAdmin, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel30)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jNameTextAdmin, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel31)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane5, javax.swing.GroupLayout.PREFERRED\_SIZE, 51, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel32)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPhoneNumAdmin, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel33)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jEmailTextAdmin, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel34)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jWebTextAdmin, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jAddButtonAdmin)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jSeparator4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLogoutButtonAdmin)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jExitButtonAdmin)

.addContainerGap())

);

jPanel14.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jBranchTableAdmin.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jBranchTableAdmin.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Username", "Name", "Address", "Phone no", "E-mail address", "Web address"

}

) {

boolean[] canEdit = new boolean [] {

false, false, false, false, false, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane6.setViewportView(jBranchTableAdmin);

jDeleteButtunAdmin.setText("Delete branch");

jDeleteButtunAdmin.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jDeleteButtunAdminActionPerformed(evt);

}

});

jModifyButtonAdmin.setText("Modify branch details");

jModifyButtonAdmin.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jModifyButtonAdminActionPerformed(evt);

}

});

jLabel19.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

jLabel19.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel19.setText("Branch Details");

javax.swing.GroupLayout jPanel14Layout = new javax.swing.GroupLayout(jPanel14);

jPanel14.setLayout(jPanel14Layout);

jPanel14Layout.setHorizontalGroup(

jPanel14Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jSeparator3, javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(jPanel14Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel14Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel14Layout.createSequentialGroup()

.addGroup(jPanel14Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel19, javax.swing.GroupLayout.PREFERRED\_SIZE, 638, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel14Layout.createSequentialGroup()

.addComponent(jDeleteButtunAdmin)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jModifyButtonAdmin)))

.addGap(0, 0, Short.MAX\_VALUE))

.addComponent(jScrollPane6, javax.swing.GroupLayout.Alignment.TRAILING))

.addContainerGap())

);

jPanel14Layout.setVerticalGroup(

jPanel14Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel14Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel19, javax.swing.GroupLayout.PREFERRED\_SIZE, 45, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane6, javax.swing.GroupLayout.DEFAULT\_SIZE, 454, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel14Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jDeleteButtunAdmin)

.addComponent(jModifyButtonAdmin))

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel14, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPanel13, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel13, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel14, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jExitButtonAdminActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);//Exists the system if exit button is pressed.

}

private void jLogoutButtonAdminActionPerformed(java.awt.event.ActionEvent evt) {

Login loginModel = new Login();//will logout by closing the Administrator page and opening the login page.

loginModel.setVisible(true);

this.setVisible(false);

}

private void jAddButtonAdminActionPerformed(java.awt.event.ActionEvent evt) {

try{//Varaible declaration where all the data from the text field to add the branches are taken.

branchTable = (DefaultTableModel) jBranchTableAdmin.getModel();

String branchUsername;

branchUsername = jUsernameTextAdmin.getText();//gets username

String branchPassword;

branchPassword = jPasswordTextAdmin.getText();//gets password

String branchName;

branchName = jNameTextAdmin.getText();//gets branch name

String branchAddress;

branchAddress = jAddressTextAdmin.getText();//gets address

String branchPhone;

branchPhone = jPhoneNumAdmin.getText();//gets phone number

String branchEmail;

branchEmail = jEmailTextAdmin.getText();//gets email

String branchWeb;

branchWeb = jWebTextAdmin.getText();//gets web address

String likeUsername = "admin";//stores "admin" as a String in a variable admin cannot add a Secretary with username admin

if (likeUsername.equals(branchUsername)){//if the username is admin, will show a dialog box saying "Username cannot be admin".

JOptionPane.showMessageDialog(null, "Username cannot be admin");

}

else if ("".equals(branchUsername) || "".equals(branchPassword) || "".equals(branchName) || "".equals(branchAddress) || "".equals(branchPhone) || "".equals(branchEmail) || "".equals(branchWeb)){

JOptionPane.showMessageDialog(null, "Please fill all the Forms");//checks whether all the forms are filled up. if not a dialog box will show up saying to fill up all the forms

}

else {//if the username is not admin and all the forms are full the following code which will attempt to get the same username from the database will be executed.

Statement equalstmt = con.createStatement();

ResultSet checkUsername = equalstmt.executeQuery("SELECT username FROM branch WHERE username = '"+branchUsername+"'");

if(checkUsername.next()){

likeUsername = checkUsername.getString(1);

}

equalstmt.close();

if(likeUsername.equals(branchUsername)){

JOptionPane.showMessageDialog(null, "Username already exist");//if the username already exist in the database then the secretary cannot be added with the same username

}

else {//if username doesnt exist then the below code which will take all teh avalues and will the add it to the database.

try {

Statement stmt = con.createStatement();

String insertOperation = "INSERT INTO `branch` (`branch\_id`, `username`, `password`, `name`, `address`, `phone`, `email`, `web\_address`) VALUES "

+ "(NULL,'"+branchUsername+"','"+branchPassword+"','"+branchName+"','"+branchAddress+"','"+branchPhone+"','"+branchEmail+"','"+branchWeb+"');";

stmt.execute(insertOperation);

stmt.close();

}

catch (SQLException e){

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, e);

JOptionPane.showMessageDialog(null, "Data not saved in database");

}

branchTable.insertRow(branchTable.getRowCount(), new Object[]{branchUsername ,branchName ,branchAddress ,branchPhone ,branchEmail ,branchWeb}); // all the data will be inserted into the table in the GUI

//then all the fields will be cleared.

jWebTextAdmin.setText("");

jEmailTextAdmin.setText("");

jPhoneNumAdmin.setText("");

jAddressTextAdmin.setText("");

jNameTextAdmin.setText("");

jPasswordTextAdmin.setText("");

jUsernameTextAdmin.setText("");

JOptionPane.showMessageDialog(null, "Branch added Successfully");//if successfull will show a dialog box with this message.

}

}

}

catch(Exception e){

JOptionPane.showMessageDialog(null, "Data not entered into database");// if any unexpeced eception happens the following error message will show up.

}

}

private void jDeleteButtunAdminActionPerformed(java.awt.event.ActionEvent evt) {

branchTable = (DefaultTableModel) jBranchTableAdmin.getModel();//initailising the table

try {

int selectedRowIndex = jBranchTableAdmin.getSelectedRow();

String selectedRowUsername;

selectedRowUsername = jBranchTableAdmin.getValueAt(selectedRowIndex, 0).toString();

try {

Statement deletestmt = con.createStatement();

String deleteUser = "DELETE FROM branch WHERE username = '"+selectedRowUsername+"'; DELETE FROM sale WHERE username = '"+selectedRowUsername+"';";// will select the branch to be deleted based on which username was selected. and also removes all the properties associated with the branch .

deletestmt.execute(deleteUser);

deletestmt.close();

branchTable.removeRow(selectedRowIndex);

JOptionPane.showMessageDialog(null, "Branch and all associated property successfully removed");//will show this dialog box ipon successfull deletion of the product.

}

catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Deleting from database unsuccessfull");//shows error message unable to delete if any problem with the databse occurs.

}

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, "Please select a branch to delete.");//will show this error if Admin dont select any data in the table to delete before pressing the delete button.

}

}

private void jModifyButtonAdminActionPerformed(java.awt.event.ActionEvent evt) {

branchTable = (DefaultTableModel) jBranchTableAdmin.getModel();

try {

int selectedRowIndex = jBranchTableAdmin.getSelectedRow();

String selectedRowUsername;

selectedRowUsername = jBranchTableAdmin.getValueAt(selectedRowIndex, 0).toString();//get the username

Statement idstmt = con.createStatement();

ResultSet checkId = idstmt.executeQuery("SELECT branch\_id FROM branch WHERE username = '"+selectedRowUsername+"'");// statement to get the branch\_id of the selected username

if(checkId.next()){

String likeId = checkId.getString(1);

idstmt.close();

try{

new ModifyBranchAdmin(likeId).setVisible(true);//will open the ModifyBranchAdmin.java and will send the data of the admin

this.setVisible(false);//closes the current Administrator Window.

}

catch (Exception e){

JOptionPane.showMessageDialog(null, "Cannot Open Modify Settings");

}

}

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, "Please select a branch to Modify.");// will say to open a branch to modify it if none is choosen

}

}

// Variables declaration - do not modify

private javax.swing.JButton jAddButtonAdmin;

private javax.swing.JTextArea jAddressTextAdmin;

private javax.swing.JTable jBranchTableAdmin;

private javax.swing.JButton jDeleteButtunAdmin;

private javax.swing.JTextField jEmailTextAdmin;

private javax.swing.JButton jExitButtonAdmin;

private javax.swing.JLabel jLabel19;

private javax.swing.JLabel jLabel27;

private javax.swing.JLabel jLabel28;

private javax.swing.JLabel jLabel29;

private javax.swing.JLabel jLabel30;

private javax.swing.JLabel jLabel31;

private javax.swing.JLabel jLabel32;

private javax.swing.JLabel jLabel33;

private javax.swing.JLabel jLabel34;

private javax.swing.JButton jLogoutButtonAdmin;

private javax.swing.JButton jModifyButtonAdmin;

private javax.swing.JTextField jNameTextAdmin;

private javax.swing.JPanel jPanel13;

private javax.swing.JPanel jPanel14;

private javax.swing.JPasswordField jPasswordTextAdmin;

private javax.swing.JTextField jPhoneNumAdmin;

private javax.swing.JScrollPane jScrollPane5;

private javax.swing.JScrollPane jScrollPane6;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JSeparator jSeparator4;

private javax.swing.JTextField jUsernameTextAdmin;

private javax.swing.JTextField jWebTextAdmin;

// End of variables declaration

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.unais;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class ModifyBranchAdmin extends javax.swing.JFrame {

Connection con;

public ModifyBranchAdmin() {

initComponents();

}

public ModifyBranchAdmin(String likeId) {//gets all the data from the database of the selected username and displays it on the GUI

initComponents();

createConnection();

jIdTextModify.setText(likeId);

try {

Statement stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM branch where branch\_id="+likeId+"");

if (result.next()){

String branchUsername = result.getString("username");

String branchPassword = result.getString("password");

String branchName = result.getString("name");

String branchAddress = result.getString("address");

String branchPhone = result.getString("phone");

String branchEmail = result.getString("email");

String branchWeb = result.getString("web\_address");

jUsernameTextModify.setText(branchUsername);

jPasswordTextModify.setText(branchPassword);

jNameTextModify.setText(branchName);

jAddressTextModify.setText(branchAddress);

jPhoneNumModify.setText(branchPhone);

jEmailTextModify.setText(branchEmail);

jWebTextModify.setText(branchWeb);

}

stmt.close();

} catch (SQLException ex) {

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void createConnection() {

try {//creates the connection to server

String serverName;

serverName = "localhost:3306";

String database;

database = "pms";

String databaseUserName;

databaseUserName = "root";

String databasePassword;

databasePassword = "";

Class.forName("com.mysql.cj.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://"+serverName+"/"+database+""

+ "?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", ""+databaseUserName+"", ""+databasePassword+"");

}

catch (SQLException | ClassNotFoundException ex) {

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, ex);

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel13 = new javax.swing.JPanel();

jLabel27 = new javax.swing.JLabel();

jLabel28 = new javax.swing.JLabel();

jUsernameTextModify = new javax.swing.JTextField();

jLabel29 = new javax.swing.JLabel();

jPasswordTextModify = new javax.swing.JPasswordField();

jLabel30 = new javax.swing.JLabel();

jNameTextModify = new javax.swing.JTextField();

jLabel31 = new javax.swing.JLabel();

jScrollPane5 = new javax.swing.JScrollPane();

jAddressTextModify = new javax.swing.JTextArea();

jPhoneNumModify = new javax.swing.JTextField();

jLabel32 = new javax.swing.JLabel();

jLabel33 = new javax.swing.JLabel();

jEmailTextModify = new javax.swing.JTextField();

jLabel34 = new javax.swing.JLabel();

jWebTextModify = new javax.swing.JTextField();

jUpdateButtonModify = new javax.swing.JButton();

jCancelButtonModify = new javax.swing.JButton();

jIdTextModify = new javax.swing.JTextField();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DO\_NOTHING\_ON\_CLOSE);

setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

setResizable(false);

jPanel13.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel27.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

jLabel27.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel27.setText("Modify Branch");

jLabel28.setText("Username");

jLabel29.setText("Password");

jLabel30.setText("Name");

jLabel31.setText("Address");

jAddressTextModify.setColumns(20);

jAddressTextModify.setRows(5);

jScrollPane5.setViewportView(jAddressTextModify);

jLabel32.setText("Phone number");

jLabel33.setText("E-mail address");

jLabel34.setText("Web Address");

jUpdateButtonModify.setText("Modify");

jUpdateButtonModify.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jUpdateButtonModifyActionPerformed(evt);

}

});

jCancelButtonModify.setText("Cancel");

jCancelButtonModify.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jCancelButtonModifyActionPerformed(evt);

}

});

jIdTextModify.setEditable(false);

jLabel1.setText("branch\_id");

javax.swing.GroupLayout jPanel13Layout = new javax.swing.GroupLayout(jPanel13);

jPanel13.setLayout(jPanel13Layout);

jPanel13Layout.setHorizontalGroup(

jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel13Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel13Layout.createSequentialGroup()

.addComponent(jLabel29, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jPasswordTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, 290, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel13Layout.createSequentialGroup()

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel31, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel32, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jPhoneNumModify, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane5, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 289, Short.MAX\_VALUE)))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel13Layout.createSequentialGroup()

.addComponent(jLabel30, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jNameTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, 290, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel13Layout.createSequentialGroup()

.addComponent(jLabel34, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jWebTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, 290, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel13Layout.createSequentialGroup()

.addComponent(jLabel33, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jEmailTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, 290, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jLabel27, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jUpdateButtonModify, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jCancelButtonModify, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel13Layout.createSequentialGroup()

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel28, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(10, 10, 10)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jUsernameTextModify)

.addComponent(jIdTextModify, javax.swing.GroupLayout.DEFAULT\_SIZE, 290, Short.MAX\_VALUE))))

.addContainerGap())

);

jPanel13Layout.setVerticalGroup(

jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel13Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel27, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jIdTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel1))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jUsernameTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel28))

.addGap(18, 18, 18)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jPasswordTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel29))

.addGap(18, 18, 18)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jNameTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel30))

.addGap(18, 18, 18)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane5, javax.swing.GroupLayout.PREFERRED\_SIZE, 51, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel31))

.addGap(18, 18, 18)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jPhoneNumModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel32))

.addGap(18, 18, 18)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jEmailTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel33))

.addGap(18, 18, 18)

.addGroup(jPanel13Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jWebTextModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel34))

.addGap(18, 18, 18)

.addComponent(jUpdateButtonModify)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jCancelButtonModify)

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel13, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel13, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jCancelButtonModifyActionPerformed(java.awt.event.ActionEvent evt) {

Administrator AdministratorModel = new Administrator();//will close the ModifyBranchAdmin.java and opens administrator back up

AdministratorModel.setVisible(true);

this.setVisible(false);

}

private void jUpdateButtonModifyActionPerformed(java.awt.event.ActionEvent evt) {

String branchId;// intialises variable which will take all the data from the text box

branchId = jIdTextModify.getText();

String branchUsername;

branchUsername = jUsernameTextModify.getText();

String branchPassword;

branchPassword = jPasswordTextModify.getText();

String branchName;

branchName = jNameTextModify.getText();

String branchAddress;

branchAddress = jAddressTextModify.getText();

String branchPhone;

branchPhone = jPhoneNumModify.getText();

String branchEmail;

branchEmail = jEmailTextModify.getText();

String branchWeb;

branchWeb = jWebTextModify.getText();

try {

Statement stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM branch where branch\_id="+branchId+" ");//gets the username of the branch

if (result.next()){

String compareBranchUsername = result.getString("username");

stmt.close();

String likeUsername = "admin";

if (likeUsername.equals(branchUsername)){

JOptionPane.showMessageDialog(null, "Username cannot be admin");

}else{

Statement equalstmt = con.createStatement();

ResultSet checkUsername = equalstmt.executeQuery("SELECT username FROM branch WHERE username = '"+branchUsername+"'");// Gets username from other branches with similar username which will in turn allow the admin to check whether the username is used by the user or some other users.

if(checkUsername.next()){

likeUsername = checkUsername.getString(1);

}

equalstmt.close();

if(compareBranchUsername.equals(branchUsername) || !likeUsername.equals(branchUsername)){//if the username is not changed it will still execute the code even though similar username exist in the database since its the its own username.

try {

Statement updatestmt = con.createStatement();

String updateOperation = "UPDATE `branch` SET `username` = '"+branchUsername+"', `password` = '"+branchPassword+"', `name` = '"+branchName+"', `address` = '"+branchAddress+"', `phone` = '"+branchPhone+"', `email` = '"+branchEmail+"', `web\_address` = '"+branchWeb+"' WHERE `branch`.`branch\_id` = "+branchId+"";

updatestmt.execute(updateOperation);

updatestmt.close();

}

catch (SQLException e){

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, e);

JOptionPane.showMessageDialog(null, "Data not modified in database");

}

JOptionPane.showMessageDialog(null, "Data modified.");//will show a data modified message.

Administrator administratorModel = new Administrator();

administratorModel.setVisible(true);//it will also open the Administrator page after the modification is done.

this.setVisible(false);//closes the curront ModifyBranchAdmin window.

}else{//If the username already exist except the user whose data is being changed the below message will show up.

JOptionPane.showMessageDialog(null, "Username already exist");

}

}

}

} catch (Exception ex) {//will show the below error if some Exception happens

JOptionPane.showMessageDialog(null, "Data cannot be modified");

}

}

// Variables declaration - do not modify

private javax.swing.JTextArea jAddressTextModify;

private javax.swing.JButton jCancelButtonModify;

private javax.swing.JTextField jEmailTextModify;

private javax.swing.JTextField jIdTextModify;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel27;

private javax.swing.JLabel jLabel28;

private javax.swing.JLabel jLabel29;

private javax.swing.JLabel jLabel30;

private javax.swing.JLabel jLabel31;

private javax.swing.JLabel jLabel32;

private javax.swing.JLabel jLabel33;

private javax.swing.JLabel jLabel34;

private javax.swing.JTextField jNameTextModify;

private javax.swing.JPanel jPanel13;

private javax.swing.JPasswordField jPasswordTextModify;

private javax.swing.JTextField jPhoneNumModify;

private javax.swing.JScrollPane jScrollPane5;

private javax.swing.JButton jUpdateButtonModify;

private javax.swing.JTextField jUsernameTextModify;

private javax.swing.JTextField jWebTextModify;

// End of variables declaration

}

package com.unais;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class Secretary extends javax.swing.JFrame {

DefaultTableModel salesTable,soldTable,searchTable;//initialising three tables in Secretary.java

Connection con;

public Secretary() {

initComponents();

}

public Secretary(String databaseUsername) {

initComponents();

createConnection();

displayProperties(databaseUsername);

}

private void displayProperties(String databaseUsername){//this dispplay all the properties to the secretarty that is logged in

jStaticUserNameText.setText(databaseUsername);

salesTable = (DefaultTableModel) jSalesTable.getModel();

try {

Statement stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM sale WHERE username ='"+databaseUsername+"' AND sold = '0'");//will shoose the data to be shown in sales list table. this will only choose products which are of the users and which are not sold

while (result.next()){

String salesId = result.getString("sale\_id");

String salesType = result.getString("type");

String salesAddress = result.getString("address");

String salesNumRooms = result.getString("number\_of\_room");

String salesPrice = result.getString("price");

String salesNumFloors = result.getString("number\_of\_floor");

String salesGarden = result.getString("garden");

String salesGarage = result.getString("garage");

String salesMonthlyCharge = result.getString("monthly\_charge");

salesTable.addRow(new Object[]{salesId,salesType,salesAddress,salesNumRooms,salesPrice,salesNumFloors,salesGarden,salesGarage,salesMonthlyCharge});

}

stmt.close();

} catch (SQLException ex) {

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, ex);

}

soldTable = (DefaultTableModel) jSoldTable.getModel();

try {

Statement stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM sale WHERE username ='"+databaseUsername+"' AND sold = '1'");//will choose all the properties which are sold and which belongs too the corresponding user and shows it in sold table

while (result.next()){

String salesId = result.getString("sale\_id");

String salesType = result.getString("type");

String salesAddress = result.getString("address");

String salesNumRooms = result.getString("number\_of\_room");

String salesPrice = result.getString("price");

String salesSoldPrice = result.getString("sold\_price");

String salesNumFloors = result.getString("number\_of\_floor");

String salesGarden = result.getString("garden");

String salesGarage = result.getString("garage");

String salesMonthlyCharge = result.getString("monthly\_charge");

soldTable.addRow(new Object[]{salesId,salesType,salesAddress,salesNumRooms,salesPrice,salesSoldPrice,salesNumFloors,salesGarden,salesGarage,salesMonthlyCharge});

}

stmt.close();

} catch (SQLException ex) {

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void createConnection() {

try {//creates the connection to server

String serverName;

serverName = "localhost:3306";

String database;

database = "pms";

String databaseUserName;

databaseUserName = "root";

String databasePassword;

databasePassword = "";

Class.forName("com.mysql.cj.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://"+serverName+"/"+database+""

+ "?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", ""+databaseUserName+"", ""+databasePassword+"");

}

catch (SQLException | ClassNotFoundException ex) {

Logger.getLogger(Secretary.class.getName()).log(Level.SEVERE, null, ex);

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

buttonGroup2 = new javax.swing.ButtonGroup();

buttonGroup1 = new javax.swing.ButtonGroup();

buttonGroup3 = new javax.swing.ButtonGroup();

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jTabbedPane1 = new javax.swing.JTabbedPane();

jPanel3 = new javax.swing.JPanel();

jScrollPane3 = new javax.swing.JScrollPane();

jSalesTable = new javax.swing.JTable();

jSoldPriceText = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jSellPropertyButton = new javax.swing.JButton();

jModifyButtonText = new javax.swing.JButton();

jDeleteButtonSec = new javax.swing.JButton();

jPanel5 = new javax.swing.JPanel();

jScrollPane5 = new javax.swing.JScrollPane();

jSoldTable = new javax.swing.JTable();

jPanel4 = new javax.swing.JPanel();

jLabel11 = new javax.swing.JLabel();

jScrollPane4 = new javax.swing.JScrollPane();

jSearchText = new javax.swing.JTextArea();

jSearchButton = new javax.swing.JButton();

jScrollPane6 = new javax.swing.JScrollPane();

jSearchTable = new javax.swing.JTable();

jStaticUserNameText = new javax.swing.JTextField();

jLabel12 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jAddressText = new javax.swing.JTextArea();

jLabel5 = new javax.swing.JLabel();

jNumberOfRoomsText = new javax.swing.JTextField();

jLabel6 = new javax.swing.JLabel();

jSellingPriceText = new javax.swing.JTextField();

jLabel7 = new javax.swing.JLabel();

jNumberOfFloorsText = new javax.swing.JTextField();

jLabel8 = new javax.swing.JLabel();

jGardenYesRB = new javax.swing.JRadioButton();

jGardenNoRB = new javax.swing.JRadioButton();

jLabel9 = new javax.swing.JLabel();

jGarageYesRB = new javax.swing.JRadioButton();

jGarageNoRB = new javax.swing.JRadioButton();

jLabel10 = new javax.swing.JLabel();

jMonthlyChargeText = new javax.swing.JTextField();

jPropertyAddButton = new javax.swing.JButton();

jExitButtonSec = new javax.swing.JButton();

jLogoutButtonSec = new javax.swing.JButton();

jSeparator2 = new javax.swing.JSeparator();

jFlatRB = new javax.swing.JRadioButton();

jHouseRB = new javax.swing.JRadioButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setMinimumSize(new java.awt.Dimension(777, 545));

jPanel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel1.setText("Manage Property");

jSalesTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Sale ID", "Type", "Address", "Number of rooms", "Selling price", "Number of Floors", "Garden", "Garage", "Monthly charge"

}

) {

boolean[] canEdit = new boolean [] {

false, false, false, false, false, false, false, false, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane3.setViewportView(jSalesTable);

jLabel4.setText("Price:");

jSellPropertyButton.setText("Mark as Sold");

jSellPropertyButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jSellPropertyButtonActionPerformed(evt);

}

});

jModifyButtonText.setText("Modify Property");

jModifyButtonText.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jModifyButtonTextActionPerformed(evt);

}

});

jDeleteButtonSec.setText("Delete property");

jDeleteButtonSec.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jDeleteButtonSecActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);

jPanel3.setLayout(jPanel3Layout);

jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT\_SIZE, 633, Short.MAX\_VALUE)

.addGroup(jPanel3Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jDeleteButtonSec)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jModifyButtonText)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jSellPropertyButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jSoldPriceText, javax.swing.GroupLayout.PREFERRED\_SIZE, 112, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT\_SIZE, 424, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jSellPropertyButton)

.addComponent(jModifyButtonText)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jDeleteButtonSec)

.addComponent(jLabel4)

.addComponent(jSoldPriceText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap())

);

jTabbedPane1.addTab("Sales List", jPanel3);

jSoldTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Sale ID", "Type", "Address", "Number of rooms", "Selling price", "Sold Price", "Number of Floors", "Garden", "Garage", "Monthly charge"

}

) {

boolean[] canEdit = new boolean [] {

false, false, false, false, false, false, false, false, false, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane5.setViewportView(jSoldTable);

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);

jPanel5.setLayout(jPanel5Layout);

jPanel5Layout.setHorizontalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane5, javax.swing.GroupLayout.DEFAULT\_SIZE, 633, Short.MAX\_VALUE)

);

jPanel5Layout.setVerticalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane5, javax.swing.GroupLayout.DEFAULT\_SIZE, 469, Short.MAX\_VALUE)

);

jTabbedPane1.addTab("Sold Properties", jPanel5);

jLabel11.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel11.setText("Query a product by address.");

jSearchText.setColumns(20);

jSearchText.setRows(5);

jSearchText.setName(""); // NOI18N

jScrollPane4.setViewportView(jSearchText);

jSearchButton.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jSearchButton.setText("Search");

jSearchButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jSearchButtonActionPerformed(evt);

}

});

jSearchTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Sale\_ID", "Type", "Address", "Number of rooms", "Selling price", "Sold", "Sold price", "Number of Floors", "Garden", "Garage", "Monthly charge"

}

) {

boolean[] canEdit = new boolean [] {

true, true, true, true, true, false, true, true, true, true, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane6.setViewportView(jSearchTable);

if (jSearchTable.getColumnModel().getColumnCount() > 0) {

jSearchTable.getColumnModel().getColumn(5).setResizable(false);

jSearchTable.getColumnModel().getColumn(10).setResizable(false);

}

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addGap(20, 20, 20)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel11, javax.swing.GroupLayout.DEFAULT\_SIZE, 323, Short.MAX\_VALUE)

.addComponent(jSearchButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane4))

.addGroup(jPanel4Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane6, javax.swing.GroupLayout.DEFAULT\_SIZE, 613, Short.MAX\_VALUE)))

.addContainerGap())

);

jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel4Layout.createSequentialGroup()

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 47, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jSearchButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane6, javax.swing.GroupLayout.DEFAULT\_SIZE, 345, Short.MAX\_VALUE)

.addContainerGap())

);

jTabbedPane1.addTab("Query Properties", jPanel4);

jStaticUserNameText.setEditable(false);

jStaticUserNameText.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jStaticUserNameText.setHorizontalAlignment(javax.swing.JTextField.CENTER);

jStaticUserNameText.setText("Username");

jLabel12.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

jLabel12.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel12.setText("Secretary Username");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTabbedPane1)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jStaticUserNameText, javax.swing.GroupLayout.DEFAULT\_SIZE, 150, Short.MAX\_VALUE)

.addComponent(jLabel12, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel1)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel12)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jStaticUserNameText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jTabbedPane1)

.addContainerGap())

);

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel2.setText("Add property");

jLabel3.setText("Address");

jAddressText.setColumns(20);

jAddressText.setRows(5);

jScrollPane1.setViewportView(jAddressText);

jLabel5.setText("Number of Rooms");

jLabel6.setText("Selling Price");

jLabel7.setText("Number of floors (Only Houses)");

jLabel8.setText("Garden (Only Houses)");

buttonGroup1.add(jGardenYesRB);

jGardenYesRB.setText("Yes");

buttonGroup1.add(jGardenNoRB);

jGardenNoRB.setText("No");

jLabel9.setText("Garage (Only Houses)");

buttonGroup2.add(jGarageYesRB);

jGarageYesRB.setText("Yes");

buttonGroup2.add(jGarageNoRB);

jGarageNoRB.setText("No");

jLabel10.setText("Monthly Charge (Only Flats)");

jPropertyAddButton.setText("ADD");

jPropertyAddButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jPropertyAddButtonActionPerformed(evt);

}

});

jExitButtonSec.setText("Exit");

jExitButtonSec.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jExitButtonSecActionPerformed(evt);

}

});

jLogoutButtonSec.setText("Logout");

jLogoutButtonSec.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jLogoutButtonSecActionPerformed(evt);

}

});

buttonGroup3.add(jFlatRB);

jFlatRB.setText("Flat");

jFlatRB.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jFlatRBActionPerformed(evt);

}

});

buttonGroup3.add(jHouseRB);

jHouseRB.setText("House");

jHouseRB.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jHouseRBActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jSeparator2, javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 188, Short.MAX\_VALUE)

.addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jNumberOfRoomsText)

.addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jSellingPriceText)

.addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jNumberOfFloorsText)

.addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel9, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel10, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jMonthlyChargeText)

.addComponent(jPropertyAddButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jExitButtonSec, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLogoutButtonSec, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jHouseRB)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jFlatRB))

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jGarageYesRB)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jGarageNoRB))

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jGardenYesRB)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jGardenNoRB)))

.addGap(0, 0, Short.MAX\_VALUE)))

.addContainerGap())

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 55, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jNumberOfRoomsText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jSellingPriceText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel7)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jNumberOfFloorsText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel8)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jGardenYesRB)

.addComponent(jGardenNoRB))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel9)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jGarageYesRB)

.addComponent(jGarageNoRB))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel10)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jMonthlyChargeText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jHouseRB)

.addComponent(jFlatRB))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPropertyAddButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jSeparator2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLogoutButtonSec)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jExitButtonSec)

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jExitButtonSecActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);//Exits the system.

}

private void jLogoutButtonSecActionPerformed(java.awt.event.ActionEvent evt) {

Login loginModel = new Login();//will open the login button

loginModel.setVisible(true);

this.setVisible(false);

}

private void jDeleteButtonSecActionPerformed(java.awt.event.ActionEvent evt) {

try {

salesTable = (DefaultTableModel) jSalesTable.getModel();

int selectedRowIndex = jSalesTable.getSelectedRow();

String selectedRowSaleId;

selectedRowSaleId = jSalesTable.getValueAt(selectedRowIndex, 0).toString();

try {

Statement deletestmt = con.createStatement();

String deleteUser = "DELETE FROM sale WHERE sale\_id = '"+selectedRowSaleId+"'";//deletes the id from the selected User

deletestmt.execute(deleteUser);

deletestmt.close();

salesTable.removeRow(selectedRowIndex);

JOptionPane.showMessageDialog(null, "Property successfully removed");//will show this message if property is successfully removed

}

catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Deleting from database unsuccessfull");//will show this message if any problem is encountered

}

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, "Please select a property to delete.");//will show up if no propertyt is selected

}

}

private void jPropertyAddButtonActionPerformed(java.awt.event.ActionEvent evt) {

try{

String userName = jStaticUserNameText.getText();

boolean flat;

flat = jFlatRB.isSelected();

boolean house;

house = jHouseRB.isSelected();

String pType=null;//will first initialise Ptype as string

if (flat){//the corresponding value is added dippending on what radio button is pressed

pType = "Flat";

}

if (house){

pType = "House";

}

String pAddress;

pAddress = jAddressText.getText();

String pNumberOfRooms;

pNumberOfRooms = jNumberOfRoomsText.getText();

String pPrice;

pPrice = jSellingPriceText.getText();

String pNumberOfFloors;

pNumberOfFloors = jNumberOfFloorsText.getText();

boolean gardenYes;

gardenYes = jGardenYesRB.isSelected();

boolean gardenNo;

gardenNo = jGardenNoRB.isSelected();

String pGarden=null;

if (gardenNo){//just like flat data garage and garden also received its data on the basis of which radio button is pressed.

pGarden = "No";

}

if (gardenYes){

pGarden = "Yes";

}

boolean garageYes;

garageYes = jGarageYesRB.isSelected();

boolean garageNo;

garageNo = jGarageNoRB.isSelected();

String pGarage=null;

if (garageNo){

pGarage = "No";

}

if (garageYes){

pGarage = "Yes";

}

String pMonthlyCharge;

pMonthlyCharge = jMonthlyChargeText.getText();

//initialising is over.

if(pType==null){

JOptionPane.showMessageDialog(null, "Please select a property type");//if no property type is selected a prompt select one will be given.

}

if (flat){//If the lat is selected the following code will be executed

//The data for the Houses as Not applicable is given here as the Secretary has choosen Flat

pNumberOfFloors = "Not applicable";

pGarden = "Not applicable";

pGarage = "Not applicable";

if ("".equals(pAddress) || "".equals(pNumberOfRooms) || "".equals(pPrice) || "".equals(pMonthlyCharge)){//Checks whether anything is empty.

JOptionPane.showMessageDialog(null, "Please fill all the Forms");

}

else{//if nothing is blank the code to add the product is executed

try {

Statement stmt = con.createStatement();

String insertOperation = "INSERT INTO `sale` (`sale\_id`, `username`, `type`, `address`, `number\_of\_room`, `price`, `number\_of\_floor`, `garden`, `garage`, `monthly\_charge`, `sold\_price`) VALUES "

+ "(NULL,'"+userName+"','"+pType+"','"+pAddress+"','"+pNumberOfRooms+"','"+pPrice+"','"+pNumberOfFloors+"','"+pGarden+"','"+pGarage+"','"+pMonthlyCharge+"',NULL);";

stmt.execute(insertOperation);

stmt.close();

JOptionPane.showMessageDialog(null, "Property Added");

//The below code will turn all the textfield and radio button back to normal.

jAddressText.setText("");

jNumberOfRoomsText.setText("");

jSellingPriceText.setText("");

jNumberOfFloorsText.setText("");

jMonthlyChargeText.setText("");

jNumberOfFloorsText.setEnabled(true);

jGardenNoRB.setEnabled(true);

jGardenYesRB.setEnabled(true);

jGarageYesRB.setEnabled(true);

jGarageNoRB.setEnabled(true);

buttonGroup3.clearSelection();

reloadSalesTable();//will clear the sale table and reload it

} catch (SQLException ex) {

Logger.getLogger(Secretary.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Problem inserting Data");//Will show up if any problem is encountered when adding Data

}

}

}

if (house){//If the house is selected the below code is executed.

//Flat properties given the String value of not applicable.

pMonthlyCharge = "Not applicable";

if ("".equals(pAddress) || "".equals(pNumberOfRooms) || "".equals(pPrice) || "".equals(pNumberOfFloors) || pGarden==null || pGarage==null ){////checks if all forms are filled.

JOptionPane.showMessageDialog(null, "Please fill all the Forms");

}

else{//if nothing is blank the code to add the product is executed

try {

Statement stmt = con.createStatement();

String insertOperation = "INSERT INTO `sale` (`sale\_id`, `username`, `type`, `address`, `number\_of\_room`, `price`, `number\_of\_floor`, `garden`, `garage`, `monthly\_charge`) VALUES "

+ "(NULL,'"+userName+"','"+pType+"','"+pAddress+"','"+pNumberOfRooms+"','"+pPrice+"','"+pNumberOfFloors+"','"+pGarden+"','"+pGarage+"','"+pMonthlyCharge+"');";

stmt.execute(insertOperation);

stmt.close();

JOptionPane.showMessageDialog(null, "Property Added");//Prompt to say the product was added

//The below code will turn all the textfield and radio button back to normal.

jAddressText.setText("");

jNumberOfRoomsText.setText("");

jSellingPriceText.setText("");

jNumberOfFloorsText.setText("");

jMonthlyChargeText.setEnabled(true);

buttonGroup1.clearSelection();

buttonGroup2.clearSelection();

buttonGroup3.clearSelection();

reloadSalesTable();//will clear the sale table and reload it

} catch (SQLException ex) {

Logger.getLogger(Secretary.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Problem inserting Data");//Will show up if any problem is encountered when adding Data

}

}

}

}

catch(Exception e){

JOptionPane.showMessageDialog(null, "Error");//Shows error if ant problem is encountered

}

}

private void jFlatRBActionPerformed(java.awt.event.ActionEvent evt) {

//When the Flat radio button is pressed the text field and radio buttons for the house will be first cleared and then will be disabled.

buttonGroup2.clearSelection();

buttonGroup1.clearSelection();

jGardenNoRB.setEnabled(false);

jGardenYesRB.setEnabled(false);

jGarageYesRB.setEnabled(false);

jGarageNoRB.setEnabled(false);

jNumberOfFloorsText.setText("");

jNumberOfFloorsText.setEnabled(false);

//Also the flat properties option button if it was disabled by house radio button will be re enabled.

jMonthlyChargeText.setEnabled(true);

}

private void jHouseRBActionPerformed(java.awt.event.ActionEvent evt) {

//When the House radio button is pressed the text field for flat will be cleared and then will be disabled.

jMonthlyChargeText.setText("");

jMonthlyChargeText.setEnabled(false);

//The radio button and the text field for house will be re-enabled if it was disabled.

jGardenNoRB.setEnabled(true);

jGardenYesRB.setEnabled(true);

jGarageYesRB.setEnabled(true);

jGarageNoRB.setEnabled(true);

jNumberOfFloorsText.setEnabled(true);

}

private void jModifyButtonTextActionPerformed(java.awt.event.ActionEvent evt) {

salesTable = (DefaultTableModel) jSalesTable.getModel();

try {

int selectedRowIndex = jSalesTable.getSelectedRow();

//When the mofy button is selected all the data from the selected row in the tabled will be added to a variable and then will be sent to the ModifyBranchAdmin.java to be modified.

String likeUserName;

String likeSalesId;

String likePType;

String likeAddress;

String likeNumberOfRooms;

String likeSellingPrice;

String likeNumberOfFloors;

String likeGarden;

String likeGarage;

String likeMonthlyCharge;

likeUserName = jStaticUserNameText.getText();

likeSalesId = jSalesTable.getValueAt(selectedRowIndex, 0).toString();

likePType = jSalesTable.getValueAt(selectedRowIndex, 1).toString();

likeAddress = jSalesTable.getValueAt(selectedRowIndex, 2).toString();

likeNumberOfRooms = jSalesTable.getValueAt(selectedRowIndex, 3).toString();

likeSellingPrice = jSalesTable.getValueAt(selectedRowIndex, 4).toString();

likeNumberOfFloors = jSalesTable.getValueAt(selectedRowIndex, 5).toString();

likeGarden = jSalesTable.getValueAt(selectedRowIndex, 6).toString();

likeGarage = jSalesTable.getValueAt(selectedRowIndex, 7).toString();

likeMonthlyCharge = jSalesTable.getValueAt(selectedRowIndex, 8).toString();

//The ModifyBranchAdmin.java is being opened with all the property details addded to it.

new ModifyPropertySecretary(likeUserName,likeSalesId,likePType,likeAddress,likeNumberOfRooms,likeSellingPrice,likeNumberOfFloors,likeGarden,likeGarage,likeMonthlyCharge).setVisible(true);

this.setVisible(false);//will set the Secretaty GUI hidden.

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, "Please select a branch to Modify.");//Will Showup if secretary didnt select a product to make change to.

}

}

private void jSellPropertyButtonActionPerformed(java.awt.event.ActionEvent evt) {

try {//This will sell the product

int selectedRowIndex = jSalesTable.getSelectedRow();

String likeSalesId;

likeSalesId = jSalesTable.getValueAt(selectedRowIndex, 0).toString();

String soldPrice;

soldPrice = jSoldPriceText.getText();

if("".equals(soldPrice)){//if no product price is selected it ask for the user to give data.

JOptionPane.showMessageDialog(null, "Please set a selling price.");

}else{

try {

Statement stmt = con.createStatement();

String insertOperation = "UPDATE `sale` SET `sold` = '1', sold\_price='"+soldPrice+"' WHERE sale\_id='"+likeSalesId+"';";

//Will make the coloumn in the sale table to change the data to sold and add the sold price to sold\_price column.

stmt.execute(insertOperation);

stmt.close();

JOptionPane.showMessageDialog(null, "Property marked as sold");//when the property is sold the dialog box will show up.

reloadSalesTable();//will clear the sale table and reload it

reloadSoldTable();//will clear the sold table and reloads it

} catch (SQLException ex) {

Logger.getLogger(Secretary.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Problem Selling Property");

}

}

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, "Please select a branch to mark as sold.");// Have to select a property to sell or an error to select a product will come up.

}

}

private void jSearchButtonActionPerformed(java.awt.event.ActionEvent evt) {

searchTable = (DefaultTableModel) jSearchTable.getModel();

searchTable.setRowCount(0);

String userName = jStaticUserNameText.getText();

String searchAddress = jSearchText.getText();

//<!---Initailising varable end--->

try {

Statement stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM sale WHERE username ='"+userName+"' AND address like '%"+searchAddress+"%'");// will select a product if the product is equal to the Secretaries username and the whatever search term is if it is similar to the address

while (result.next()){

String salesId = result.getString("sale\_id");

String salesType = result.getString("type");

String salesAddress = result.getString("address");

String salesNumRooms = result.getString("number\_of\_room");

String salesPrice = result.getString("price");

String soldCheck = result.getString("sold");

if (soldCheck.equals("0")){

soldCheck="Not sold";//if the product is not sold the sold row will show "Not sold"

}else{

soldCheck="Sold";

}

String soldPrice = result.getString("sold\_price");

if (soldPrice == null){

soldPrice="No data";//if the product is not sold the sold Price will show "no data"

}

String salesNumFloors = result.getString("number\_of\_floor");

String salesGarden = result.getString("garden");

String salesGarage = result.getString("garage");

String salesMonthlyCharge = result.getString("monthly\_charge");

searchTable.addRow(new Object[]{salesId,salesType,salesAddress,salesNumRooms,salesPrice,soldCheck,soldPrice,salesNumFloors,salesGarden,salesGarage,salesMonthlyCharge});//will show the results

}

stmt.close();

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, "Problem Searching");//will popup if any unexpected problem is encountered.

}

}

// Variables declaration - do not modify

private javax.swing.ButtonGroup buttonGroup1;

private javax.swing.ButtonGroup buttonGroup2;

private javax.swing.ButtonGroup buttonGroup3;

private javax.swing.JTextArea jAddressText;

private javax.swing.JButton jDeleteButtonSec;

private javax.swing.JButton jExitButtonSec;

private javax.swing.JRadioButton jFlatRB;

private javax.swing.JRadioButton jGarageNoRB;

private javax.swing.JRadioButton jGarageYesRB;

private javax.swing.JRadioButton jGardenNoRB;

private javax.swing.JRadioButton jGardenYesRB;

private javax.swing.JRadioButton jHouseRB;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JButton jLogoutButtonSec;

private javax.swing.JButton jModifyButtonText;

private javax.swing.JTextField jMonthlyChargeText;

private javax.swing.JTextField jNumberOfFloorsText;

private javax.swing.JTextField jNumberOfRoomsText;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JPanel jPanel5;

private javax.swing.JButton jPropertyAddButton;

private javax.swing.JTable jSalesTable;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JScrollPane jScrollPane4;

private javax.swing.JScrollPane jScrollPane5;

private javax.swing.JScrollPane jScrollPane6;

private javax.swing.JButton jSearchButton;

private javax.swing.JTable jSearchTable;

private javax.swing.JTextArea jSearchText;

private javax.swing.JButton jSellPropertyButton;

private javax.swing.JTextField jSellingPriceText;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JTextField jSoldPriceText;

private javax.swing.JTable jSoldTable;

private javax.swing.JTextField jStaticUserNameText;

private javax.swing.JTabbedPane jTabbedPane1;

// End of variables declaration

private void reloadSalesTable() {//This Method will clear the sale table and then reload the data in the sale table. Used when adding new items and for searching

salesTable.setRowCount(0);

String userName = jStaticUserNameText.getText();

try {

Statement stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM sale WHERE username ='"+userName+"' AND sold='0' ");

while (result.next()){

String salesId = result.getString("sale\_id");

String salesType = result.getString("type");

String salesAddress = result.getString("address");

String salesNumRooms = result.getString("number\_of\_room");

String salesPrice = result.getString("price");

String salesNumFloors = result.getString("number\_of\_floor");

String salesGarden = result.getString("garden");

String salesGarage = result.getString("garage");

String salesMonthlyCharge = result.getString("monthly\_charge");

salesTable.addRow(new Object[]{salesId,salesType,salesAddress,salesNumRooms,salesPrice,salesNumFloors,salesGarden,salesGarage,salesMonthlyCharge});

}

stmt.close();

} catch (SQLException ex) {

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void reloadSoldTable() {//This Method will clear the sold table and then reload the data in the sold table. Used when selling items.

soldTable.setRowCount(0);

String userName = jStaticUserNameText.getText();

try {

Statement stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM sale WHERE username ='"+userName+"' AND sold='1' ");

while (result.next()){

String salesId = result.getString("sale\_id");

String salesType = result.getString("type");

String salesAddress = result.getString("address");

String salesNumRooms = result.getString("number\_of\_room");

String salesPrice = result.getString("price");

String soldPrice = result.getString("sold\_price");

String salesNumFloors = result.getString("number\_of\_floor");

String salesGarden = result.getString("garden");

String salesGarage = result.getString("garage");

String salesMonthlyCharge = result.getString("monthly\_charge");

soldTable.addRow(new Object[]{salesId,salesType,salesAddress,salesNumRooms,salesPrice,soldPrice,salesNumFloors,salesGarden,salesGarage,salesMonthlyCharge});

}

stmt.close();

} catch (SQLException ex) {

Logger.getLogger(Administrator.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

package com.unais;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class ModifyPropertySecretary extends javax.swing.JFrame {

Connection con;

public ModifyPropertySecretary() {

initComponents();

} //Will get all the data of the property from Secretary.java

public ModifyPropertySecretary(String likeUserName,String likeSalesId,String likePType,String likeAddress,String likeNumberOfRooms,String likeSellingPrice,String likeNumberOfFloors,String likeGarden,String likeGarage,String likeMonthlyCharge) {

initComponents();

createConnection();

jUserNameModify.setText(likeUserName);

jSalesIdModify.setText(likeSalesId);

if(likePType.equals("House")){//If The house is choosen the following code will be used to fill up the text fields

jGardenNoRBModify.setEnabled(true);

jGardenYesRBModify.setEnabled(true);

jGarageYesRBModify.setEnabled(true);

jGarageNoRBModify.setEnabled(true);

jNumberOfFloorsModify.setEnabled(true);

jMonthlyChargeModify.setText("");

jMonthlyChargeModify.setEnabled(false);

jAddressModify.setText(likeAddress);

jNumberOfRoomsModify.setText(likeNumberOfRooms);

jSellingPriceModify.setText(likeSellingPrice);

jNumberOfFloorsModify.setText(likeNumberOfFloors);

if (likeGarden.equals("Yes")){

jGardenYesRBModify.setSelected(true);

}else{

jGardenNoRBModify.setSelected(true);

}

if (likeGarage.equals("Yes")){

jGarageYesRBModify.setSelected(true);

}else{

jGarageNoRBModify.setSelected(true);

}

jHouseRBModify.setSelected(true);

}else{//If The flat is choosen the following code will be used to fill up the text fields

buttonGroup2.clearSelection();

buttonGroup1.clearSelection();

jGardenNoRBModify.setEnabled(false);

jGardenYesRBModify.setEnabled(false);

jGarageYesRBModify.setEnabled(false);

jGarageNoRBModify.setEnabled(false);

jMonthlyChargeModify.setEnabled(true);

jNumberOfFloorsModify.setText("");

jNumberOfFloorsModify.setEnabled(false);

jAddressModify.setText(likeAddress);

jNumberOfRoomsModify.setText(likeNumberOfRooms);

jSellingPriceModify.setText(likeSellingPrice);

jMonthlyChargeModify.setText(likeMonthlyCharge);

jFlatRBModify.setSelected(true);

}

}

private void createConnection() {

try {//creates the connection to server

String serverName;

serverName = "localhost:3306";

String database;

database = "pms";

String databaseUserName;

databaseUserName = "root";

String databasePassword;

databasePassword = "";

Class.forName("com.mysql.cj.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://"+serverName+"/"+database+""

+ "?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", ""+databaseUserName+"", ""+databasePassword+"");

}

catch (SQLException | ClassNotFoundException ex) {

Logger.getLogger(Secretary.class.getName()).log(Level.SEVERE, null, ex);

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

buttonGroup1 = new javax.swing.ButtonGroup();

buttonGroup2 = new javax.swing.ButtonGroup();

buttonGroup3 = new javax.swing.ButtonGroup();

jPanel2 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jAddressModify = new javax.swing.JTextArea();

jLabel5 = new javax.swing.JLabel();

jNumberOfRoomsModify = new javax.swing.JTextField();

jLabel6 = new javax.swing.JLabel();

jSellingPriceModify = new javax.swing.JTextField();

jLabel7 = new javax.swing.JLabel();

jNumberOfFloorsModify = new javax.swing.JTextField();

jLabel8 = new javax.swing.JLabel();

jGardenYesRBModify = new javax.swing.JRadioButton();

jGardenNoRBModify = new javax.swing.JRadioButton();

jLabel9 = new javax.swing.JLabel();

jGarageYesRBModify = new javax.swing.JRadioButton();

jGarageNoRBModify = new javax.swing.JRadioButton();

jLabel10 = new javax.swing.JLabel();

jMonthlyChargeModify = new javax.swing.JTextField();

jPropertyModifyButton = new javax.swing.JButton();

jCancelButtonModify = new javax.swing.JButton();

jFlatRBModify = new javax.swing.JRadioButton();

jHouseRBModify = new javax.swing.JRadioButton();

jLabel1 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jSalesIdModify = new javax.swing.JTextField();

jLabel11 = new javax.swing.JLabel();

jUserNameModify = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.DO\_NOTHING\_ON\_CLOSE);

setResizable(false);

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel2.setText("Edit Property");

jLabel3.setText("Address");

jAddressModify.setColumns(20);

jAddressModify.setRows(5);

jScrollPane1.setViewportView(jAddressModify);

jLabel5.setText("Number of Rooms");

jLabel6.setText("Selling Price");

jLabel7.setText("Number of floors (Only Houses)");

jLabel8.setText("Garden (Only Houses)");

buttonGroup1.add(jGardenYesRBModify);

jGardenYesRBModify.setText("Yes");

buttonGroup1.add(jGardenNoRBModify);

jGardenNoRBModify.setText("No");

jGardenNoRBModify.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jGardenNoRBModifyActionPerformed(evt);

}

});

jLabel9.setText("Garage (Only Houses)");

buttonGroup2.add(jGarageYesRBModify);

jGarageYesRBModify.setText("Yes");

buttonGroup2.add(jGarageNoRBModify);

jGarageNoRBModify.setText("No");

jLabel10.setText("Monthly Charge (Only Flats)");

jPropertyModifyButton.setText("Modify Properties");

jPropertyModifyButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jPropertyModifyButtonActionPerformed(evt);

}

});

jCancelButtonModify.setText("Cancel ");

jCancelButtonModify.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jCancelButtonModifyActionPerformed(evt);

}

});

buttonGroup3.add(jFlatRBModify);

jFlatRBModify.setText("Flat");

jFlatRBModify.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jFlatRBModifyActionPerformed(evt);

}

});

buttonGroup3.add(jHouseRBModify);

jHouseRBModify.setText("House");

jHouseRBModify.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jHouseRBModifyActionPerformed(evt);

}

});

jLabel1.setText("Property Type");

jLabel4.setText("Property ID");

jSalesIdModify.setEditable(false);

jLabel11.setText("Secretary Username");

jUserNameModify.setEditable(false);

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jCancelButtonModify, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel10, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel9, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT\_SIZE, 164, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jLabel11)

.addGap(0, 0, Short.MAX\_VALUE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jPropertyModifyButton, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jHouseRBModify)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jFlatRBModify))

.addComponent(jMonthlyChargeModify, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jSellingPriceModify, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jNumberOfRoomsModify, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 241, Short.MAX\_VALUE)

.addComponent(jNumberOfFloorsModify)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jGardenYesRBModify)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jGardenNoRBModify, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jGarageYesRBModify)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jGarageNoRBModify, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(jSalesIdModify)

.addComponent(jUserNameModify))))

.addContainerGap())

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel11)

.addComponent(jUserNameModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel4, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jSalesIdModify, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 55, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jNumberOfRoomsModify)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jSellingPriceModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel7)

.addComponent(jNumberOfFloorsModify, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel8, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jGardenYesRBModify)

.addComponent(jGardenNoRBModify)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jGarageYesRBModify)

.addComponent(jGarageNoRBModify))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jMonthlyChargeModify)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jHouseRBModify)

.addComponent(jFlatRBModify)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jPropertyModifyButton)

.addComponent(jCancelButtonModify))

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void jPropertyModifyButtonActionPerformed(java.awt.event.ActionEvent evt) {

try{//Initailisation same as in adding a product.

String saleId = jSalesIdModify.getText();

boolean flat;

flat = jFlatRBModify.isSelected();

boolean house;

house = jHouseRBModify.isSelected();

String pType=null;

if (flat){

pType = "Flat";

}

if (house){

pType = "House";

}

String pAddress;

pAddress = jAddressModify.getText();

String pNumberOfRooms;

pNumberOfRooms = jNumberOfRoomsModify.getText();

String pPrice;

pPrice = jSellingPriceModify.getText();

String pNumberOfFloors;

pNumberOfFloors = jNumberOfFloorsModify.getText();

boolean gardenYes;

gardenYes = jGardenYesRBModify.isSelected();

boolean gardenNo;

gardenNo = jGardenNoRBModify.isSelected();

String pGarden=null;

if (gardenNo){

pGarden = "No";

}

if (gardenYes){

pGarden = "Yes";

}

boolean garageYes;

garageYes = jGarageYesRBModify.isSelected();

boolean garageNo;

garageNo = jGarageNoRBModify.isSelected();

String pGarage=null;

if (garageNo){

pGarage = "No";

}

if (garageYes){

pGarage = "Yes";

}

String pMonthlyCharge;

pMonthlyCharge = jMonthlyChargeModify.getText();

//Initialising end.

if(pType==null){

JOptionPane.showMessageDialog(null, "Please select a property type");

}

if (flat){

pNumberOfFloors = "Not applicable";

pGarden = "Not applicable";

pGarage = "Not applicable";

if ("".equals(pAddress) || "".equals(pNumberOfRooms) || "".equals(pPrice) || "".equals(pMonthlyCharge)){

JOptionPane.showMessageDialog(null, "Please fill all the Forms");

}

else{

try {//Will update the data with the new one in the GUI.

Statement stmt = con.createStatement();

String insertOperation = "UPDATE `sale` SET `type` = '"+pType+"', `address` = '"+pAddress+"', `number\_of\_room` = '"+pNumberOfRooms+"', `price` = '"+pPrice+"',"

+ " `number\_of\_floor` = '"+pNumberOfFloors+"', `garden` = '"+pGarden+"', `garage` = '"+pGarage+"', `monthly\_charge` = '"+pMonthlyCharge+"' WHERE `sale\_id` = "+saleId+";";

stmt.execute(insertOperation);

stmt.close();

JOptionPane.showMessageDialog(null, "Property Modified");

Secretary secretaryModel = new Secretary(jUserNameModify.getText());

secretaryModel.setVisible(true);

this.setVisible(false);

} catch (SQLException ex) {

Logger.getLogger(Secretary.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Problem inserting Data");

}

}

}

if (house){

pMonthlyCharge = "Not applicable";

if ("".equals(pAddress) || "".equals(pNumberOfRooms) || "".equals(pPrice) || "".equals(pNumberOfFloors) || pGarden==null || pGarage==null ){

JOptionPane.showMessageDialog(null, "Please fill all the Forms");

}

else{

try {

Statement stmt = con.createStatement();

String insertOperation = "UPDATE `sale` SET `type` = '"+pType+"', `address` = '"+pAddress+"', `number\_of\_room` = '"+pNumberOfRooms+"', `price` = '"+pPrice+"',"

+ " `number\_of\_floor` = '"+pNumberOfFloors+"', `garden` = '"+pGarden+"', `garage` = '"+pGarage+"', `monthly\_charge` = '"+pMonthlyCharge+"' WHERE `sale\_id` = "+saleId+";";

stmt.execute(insertOperation);

stmt.close();

JOptionPane.showMessageDialog(null, "Property Modified");

Secretary secretaryModel = new Secretary(jUserNameModify.getText());

secretaryModel.setVisible(true);

this.setVisible(false);

} catch (SQLException ex) {

Logger.getLogger(Secretary.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Problem inserting Data");

}

}

}

}

catch(Exception e){

e.printStackTrace();

JOptionPane.showMessageDialog(null, "Error");//This will show up if any encounter with errors.

}

}

private void jCancelButtonModifyActionPerformed(java.awt.event.ActionEvent evt) {

Secretary secretaryModel = new Secretary(jUserNameModify.getText());

secretaryModel.setVisible(true);

this.setVisible(false);

}

private void jFlatRBModifyActionPerformed(java.awt.event.ActionEvent evt) {

//Same behaviour as the flat radiobutton in one in Secretary.java to add properties

buttonGroup2.clearSelection();

buttonGroup1.clearSelection();

jGardenNoRBModify.setEnabled(false);

jGardenYesRBModify.setEnabled(false);

jGarageYesRBModify.setEnabled(false);

jGarageNoRBModify.setEnabled(false);

jMonthlyChargeModify.setEnabled(true);

jNumberOfFloorsModify.setText("");

jNumberOfFloorsModify.setEnabled(false);

}

private void jHouseRBModifyActionPerformed(java.awt.event.ActionEvent evt) {

//Same behaviour as the House radiobutton in one in Secretary.java to add properties

jGardenNoRBModify.setEnabled(true);

jGardenYesRBModify.setEnabled(true);

jGarageYesRBModify.setEnabled(true);

jGarageNoRBModify.setEnabled(true);

jNumberOfFloorsModify.setEnabled(true);

jMonthlyChargeModify.setText("");

jMonthlyChargeModify.setEnabled(false);

}

private void jGardenNoRBModifyActionPerformed(java.awt.event.ActionEvent evt) {

}

// Variables declaration - do not modify

private javax.swing.ButtonGroup buttonGroup1;

private javax.swing.ButtonGroup buttonGroup2;

private javax.swing.ButtonGroup buttonGroup3;

private javax.swing.JTextArea jAddressModify;

private javax.swing.JButton jCancelButtonModify;

private javax.swing.JRadioButton jFlatRBModify;

private javax.swing.JRadioButton jGarageNoRBModify;

private javax.swing.JRadioButton jGarageYesRBModify;

private javax.swing.JRadioButton jGardenNoRBModify;

private javax.swing.JRadioButton jGardenYesRBModify;

private javax.swing.JRadioButton jHouseRBModify;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JTextField jMonthlyChargeModify;

private javax.swing.JTextField jNumberOfFloorsModify;

private javax.swing.JTextField jNumberOfRoomsModify;

private javax.swing.JPanel jPanel2;

private javax.swing.JButton jPropertyModifyButton;

private javax.swing.JTextField jSalesIdModify;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField jSellingPriceModify;

private javax.swing.JTextField jUserNameModify;

// End of variables declaration

}

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |