**JAVA SWINGS BASED- APARTMENT MANAGEMENT SYSTEM**

**- SQL CONNECTIVITY USING JDBC**

*A*

*Report*

*Submitted in partial fulfillment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

# INFORMATION TECHNOLOGY

By

Harshavardhan Bachu<1602-19-737-072>

Under the guidance of Ms B. Leelavathy



Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-3

# BONAFIDE CERTIFICATE

This is to certify that this project report titled ‘**Apartment Management System**’ is a project work of Mr. Harshavardhan Bachu bearing roll no. 1602-19-737-07 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature Signature

Internal Examiner External Examiner

**ABSTRACT:**

Apartment Management System is a console based application which allows

Residents of an apartment to view all their announcements, payments, charges, expenses. It also allows admins and members of the committee to keep an eye on flow of non-residential members in and out the apartment.

**REQUIREMENT ANALYSIS:**

**List of Tables:**

* Users
* Apartments
* Flats
* Residents
* Visitors
* Committee
* Charges
* Expenses
* Payments
* Announcements

**List of attributes with their domain types:**

**Users:**

* User Name varchar2(20)
* Login id varchar2(15)
* Password varchar2(20)
* User type varchar2(10)

**Flats:**

* Flat Id number(3)
* Apartment Id number(3)
* Flat number number(4)
* Flat type varchar(10)

**Residents:**

* Resident Id number(3)
* Flat Id number(3)
* Resident Name varchar2(20)
* Resident Type varchar(10)
* Resident Phone Number number(10)

**Expenses:**

* Expense Id number(25)
* Apartment Id number(3)
* Expense Type varchar2(20)
* Expense Description varchar2(200)
* Expense Amount number(20)

**Charges:**

* Charges Id number(25)
* Flat Id number(3)
* Charges Type varchar2(20)
* Charges Description varchar2(200)
* Charges Amount number(20)

**Payments:**

* Payment Id number(25)
* Charges Id number(25)
* Resident Id number(3)
* Payment Amount number(20)
* Payment Date date
* Payment Type varchar2(10)

**Visitors:**

* Visitor Id number(25)
* Visitor Name varchar2(20)
* Visitor Number number(10)
* In Date date
* In Time varchar2(10)
* Out Time varchar2(10)
* Flat Id number(3)

**Committee:**

* Member Id number(5)
* Member Name varchar2(20)
* Flat Number number(3)
* Member Type varchar2(20)

**Announcements:**

* Announcement Id number()
* Announcement Description varchar2(200)
* Announcement Start Date date
* Announcement End Date date

## AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based apartment management system that allows us to manage the activities going on in the apartment , expenses and visitors . These values are to be updated in the database using JDBC connectivity.

## ARCHITECTURE AND TECHNOLOGY

**Software used:** Java Eclipse, Oracle 11g Database, Java SE version 13, SQL\*Plus.

**Java SWINGS:**

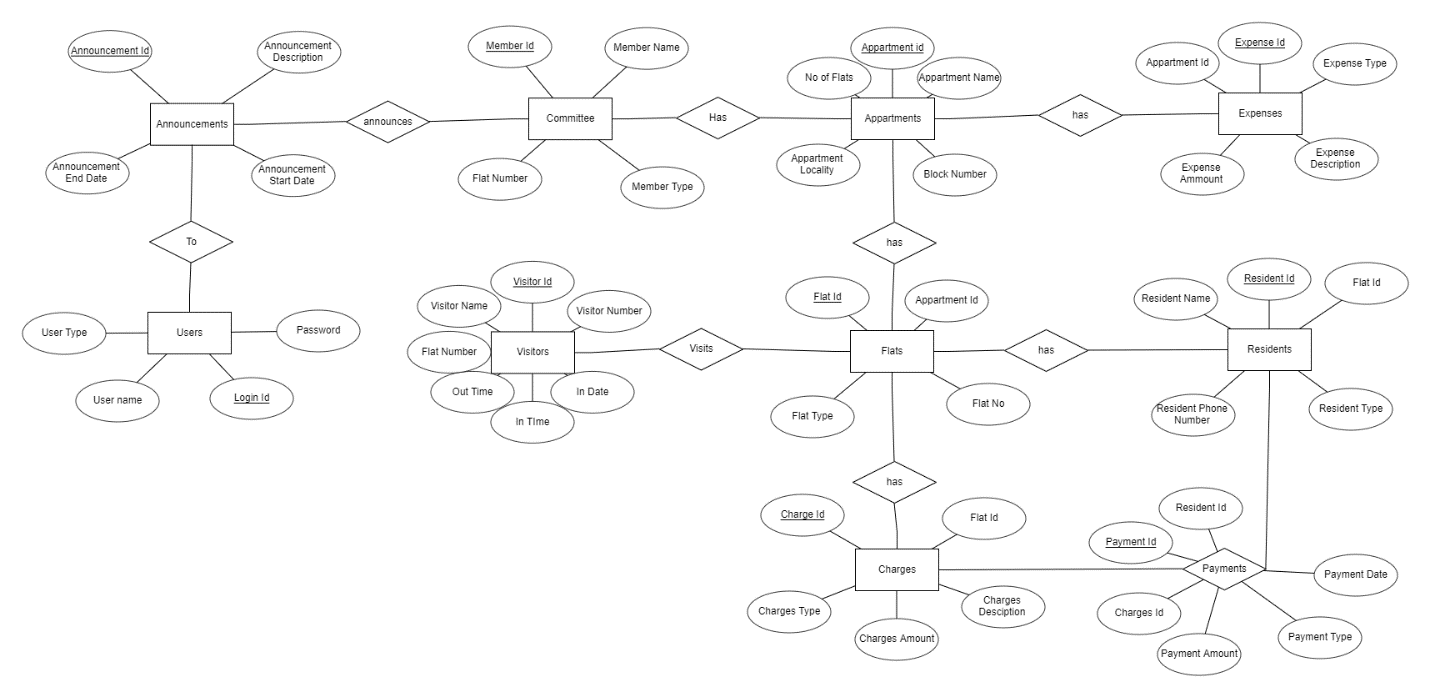
**Java SWINGS** is an API to develop GUI or window-based applications in java. Java SWING components are platform-independent. It is lightweight. The javax.swing package provides classes for SWING API such as JTextField, JLabel, JTextArea, JRadioButton, JCheckBox, JChoice, JList etc.

**SQL:**

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

**DESIGN:**

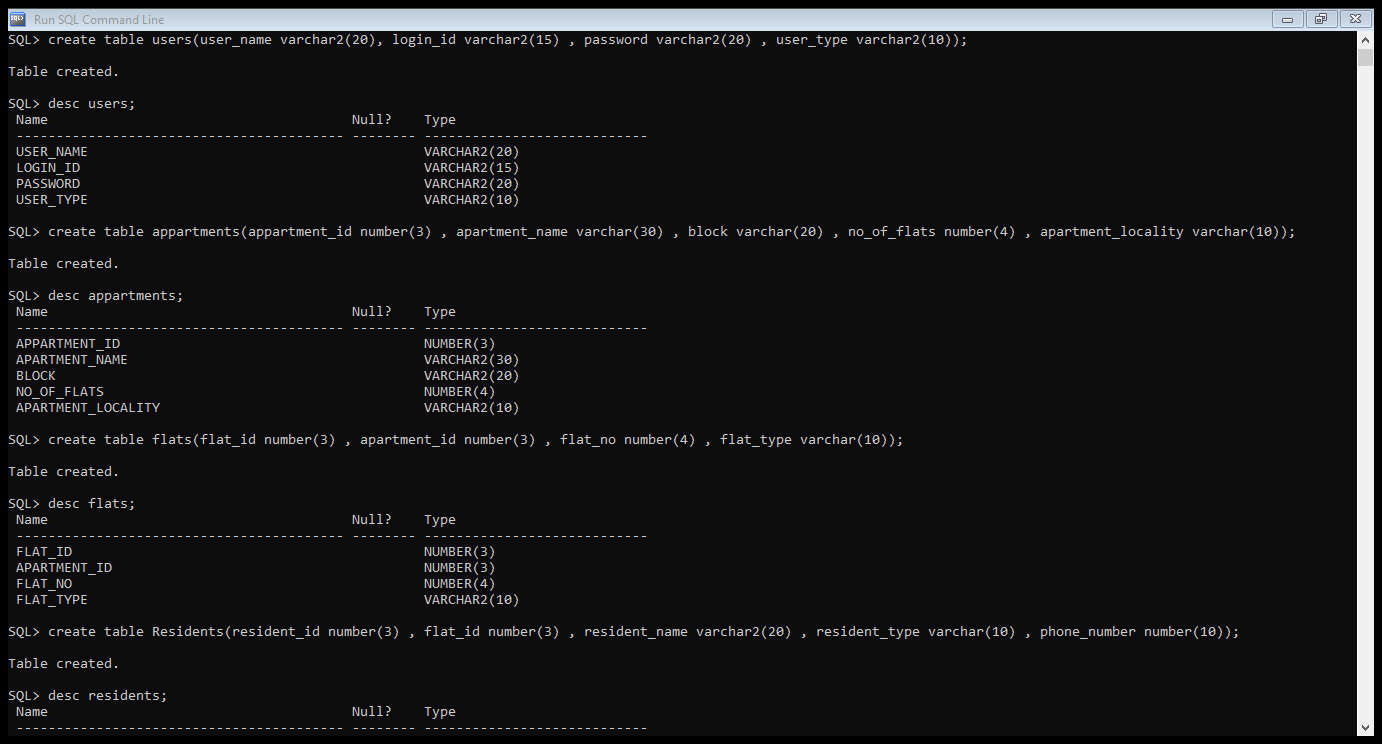
**ER DIAGRAM:**

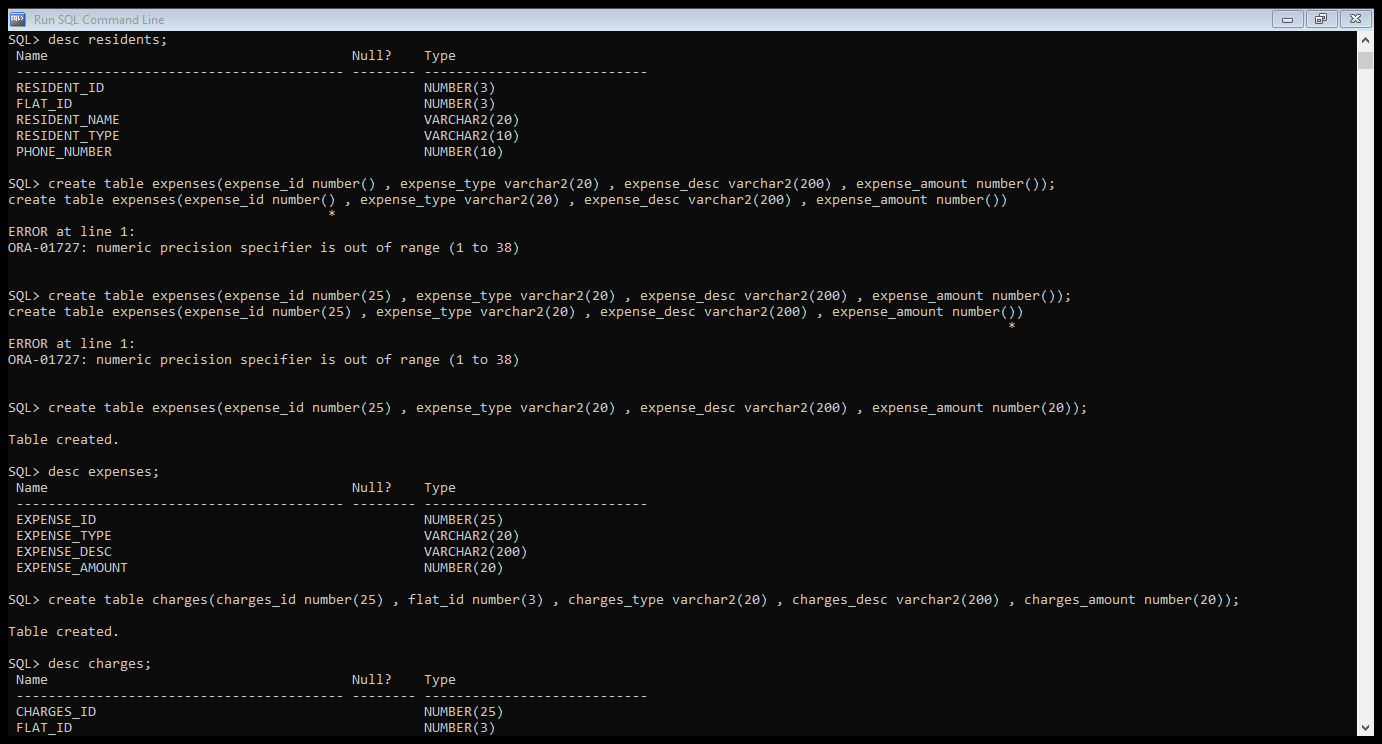


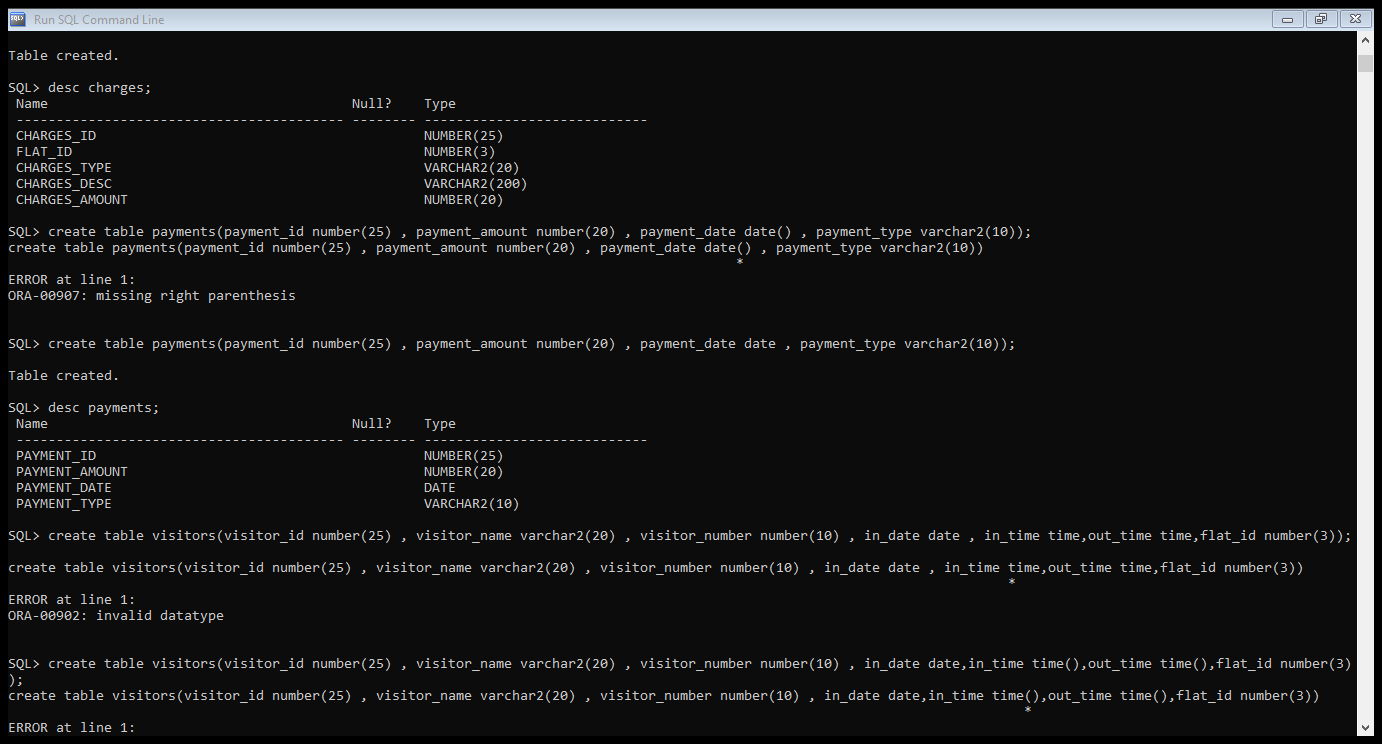
**Mapping Cardinality and participation constraints:**

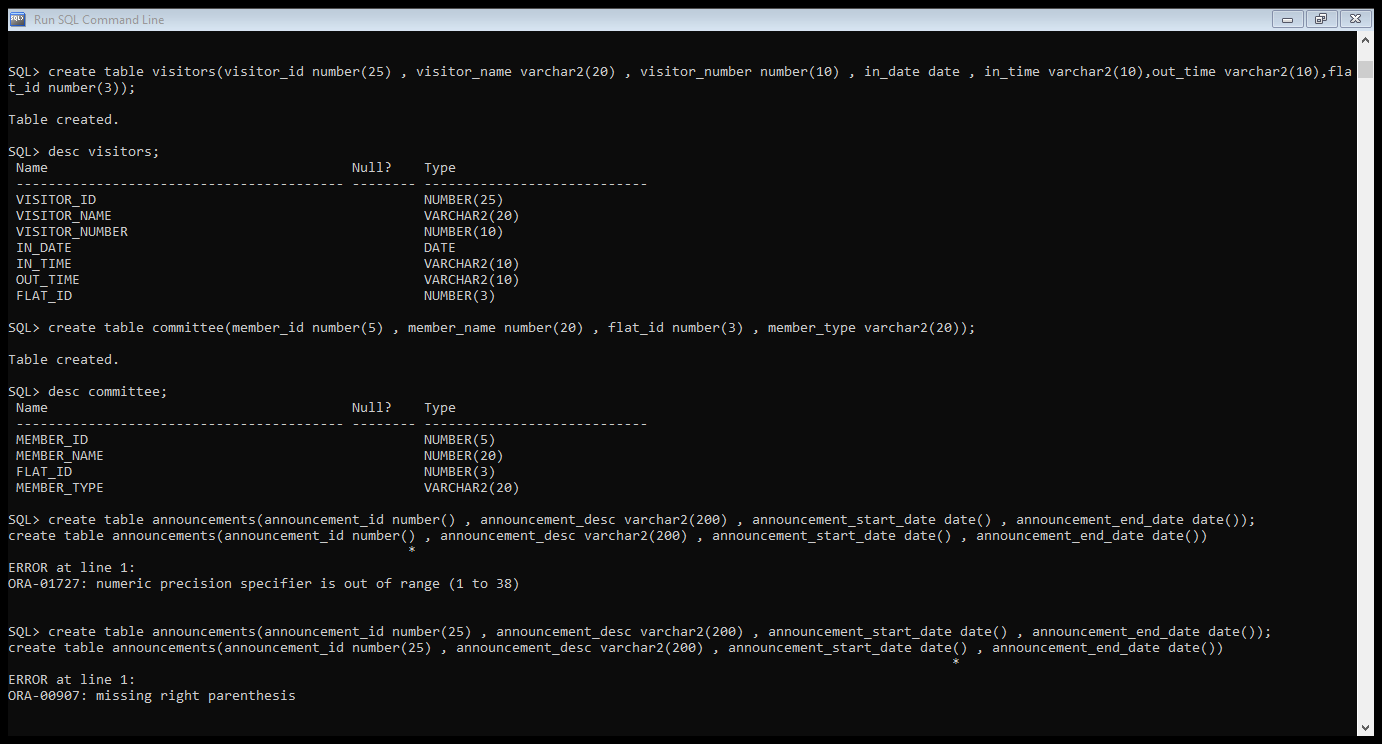
Every user who login into application need user id and password. So, login has full participation to users. A user can only be either admin or resident so it is one to one relationship. One admin can view any number of expenses, charges, users so it is one to many relationship. One user can view any number of payments of that particular user, so it is also one to many relationship.

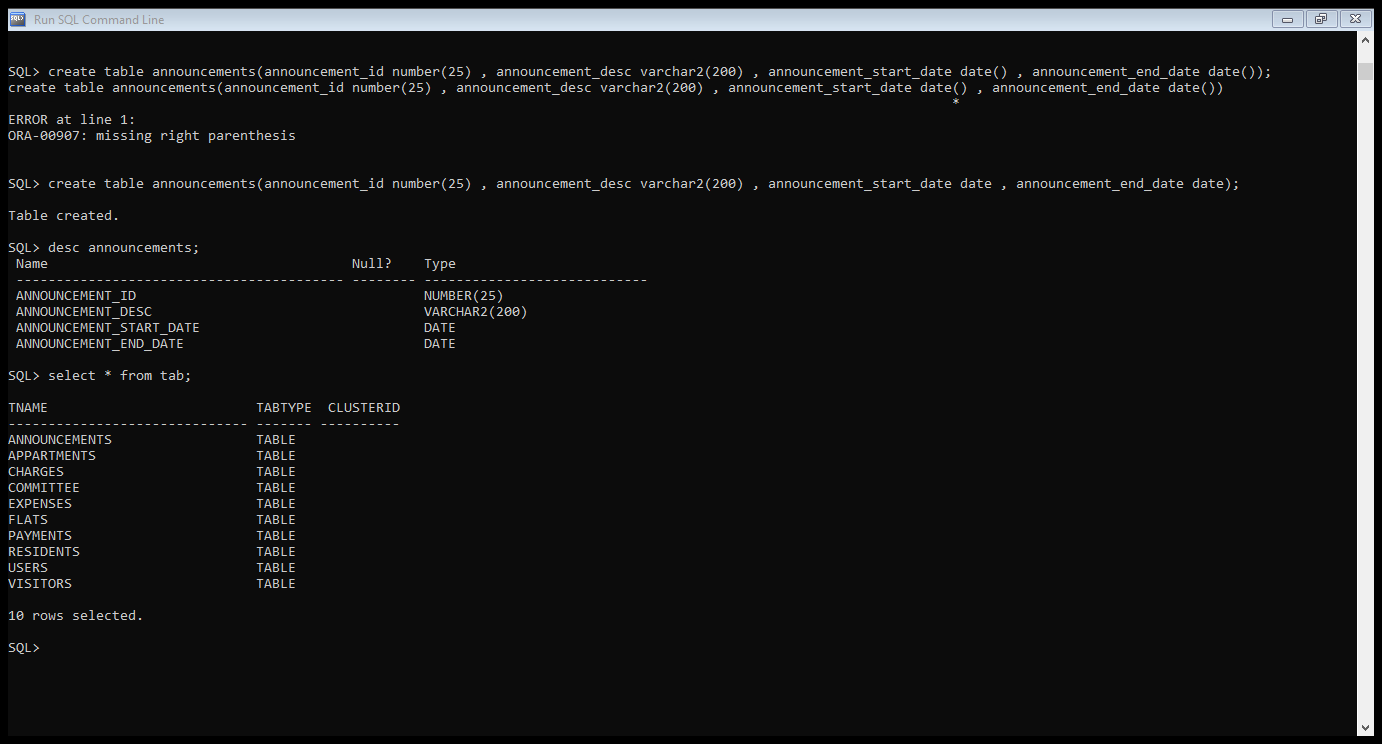
**DDL COMMANDS:**

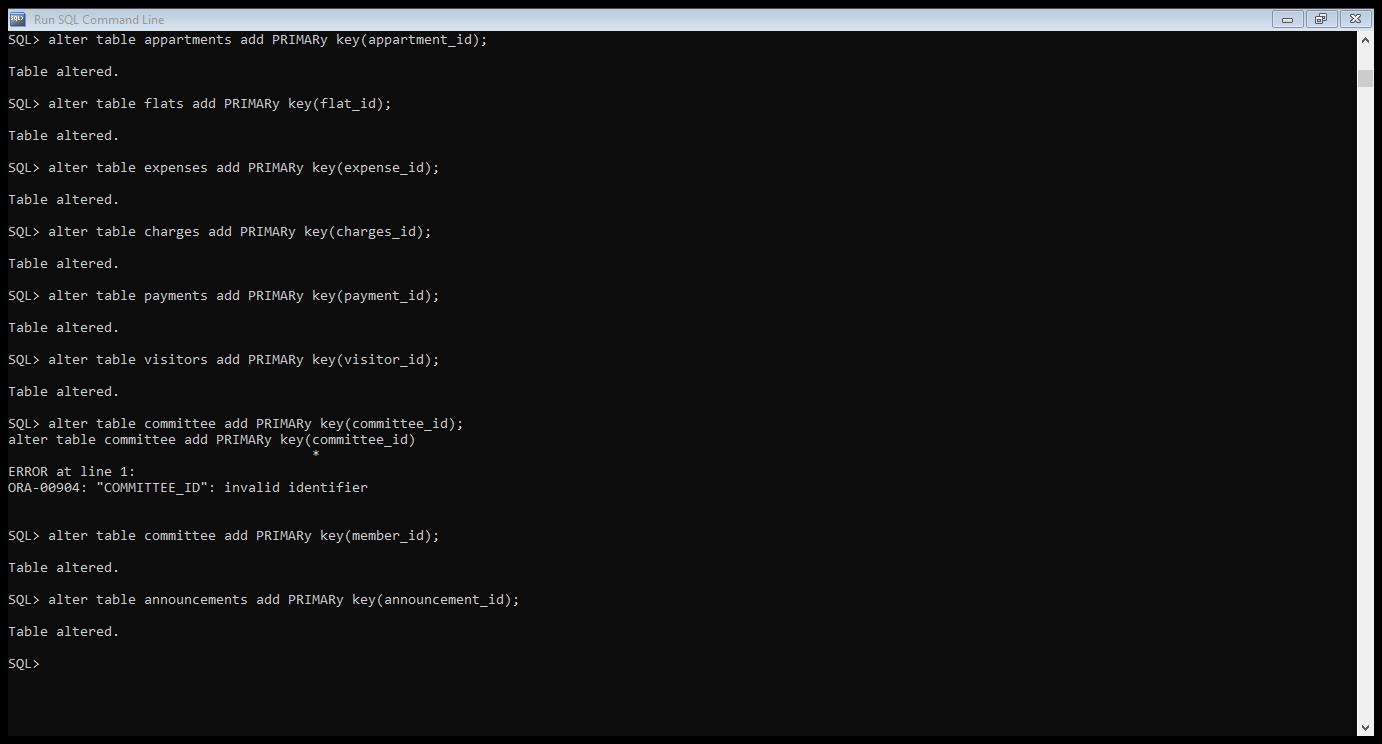
****

****

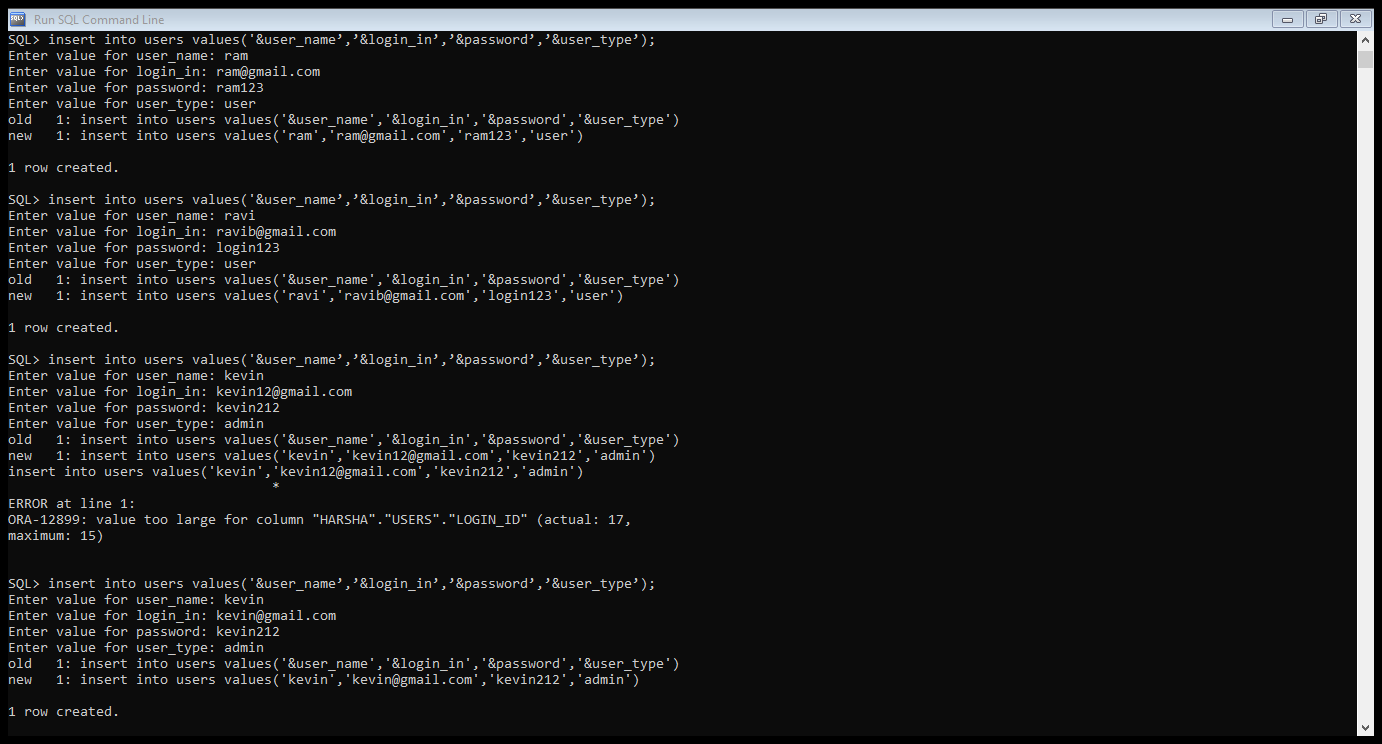
****

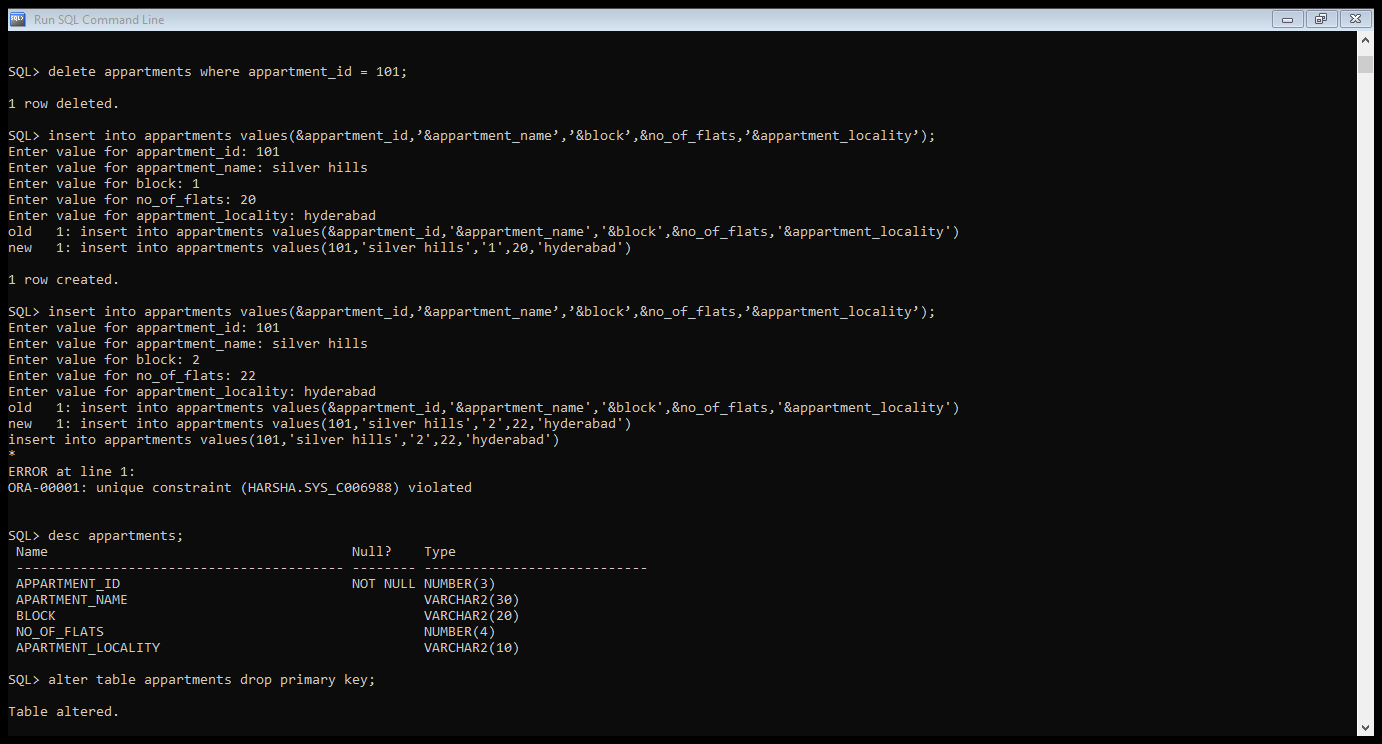
****

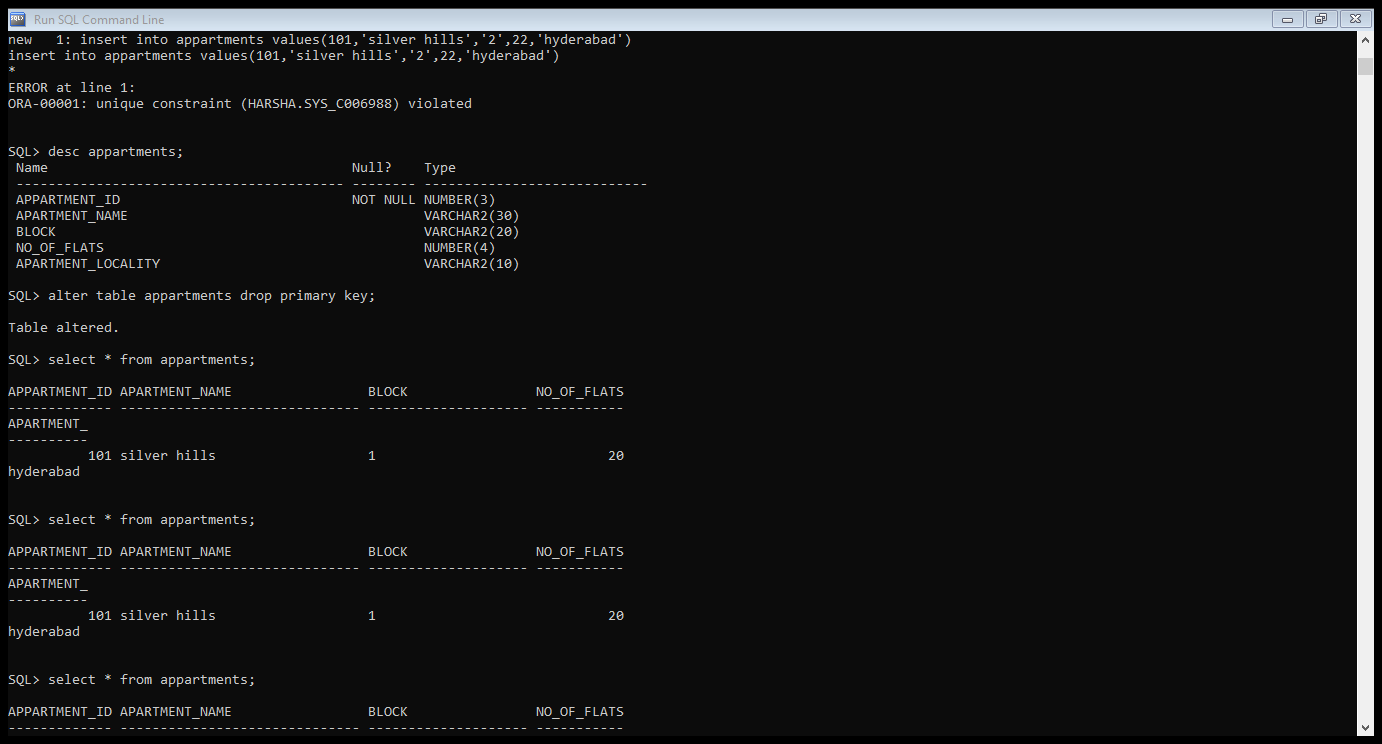
****

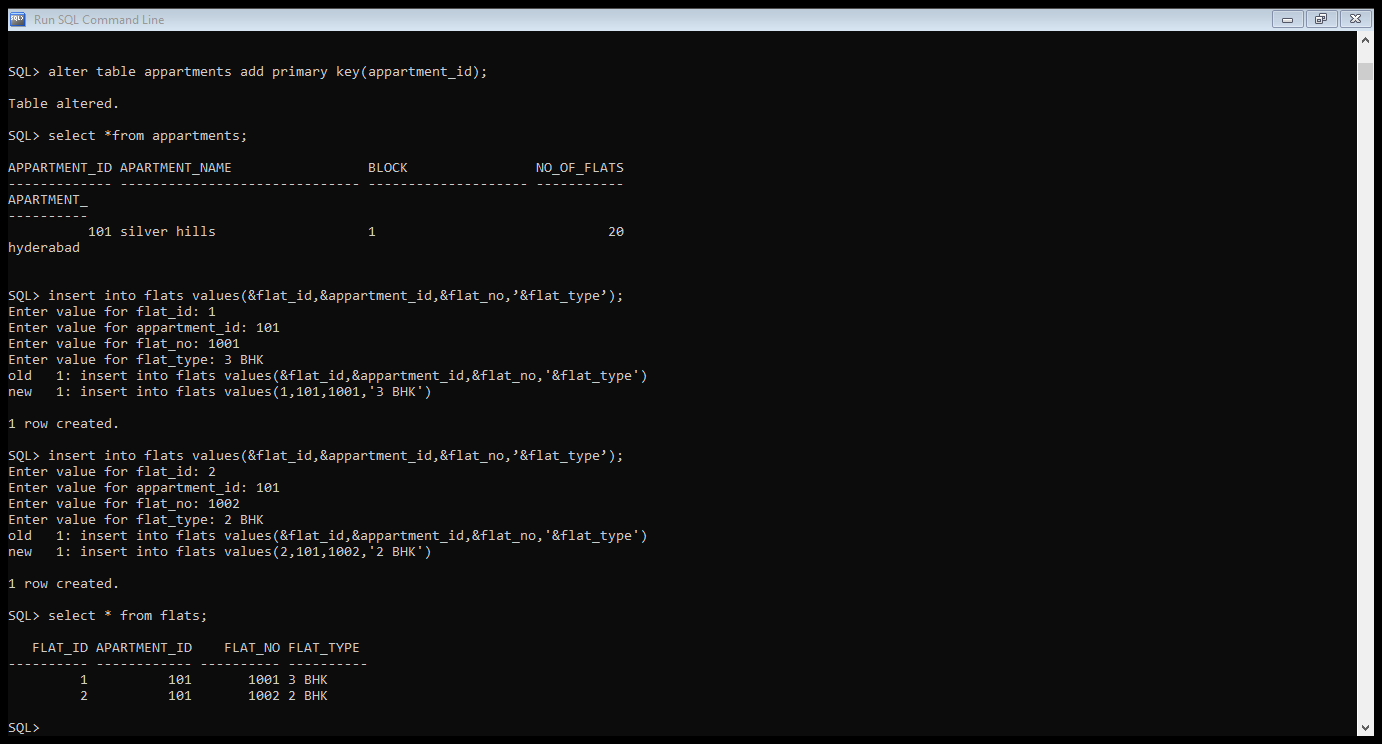
****

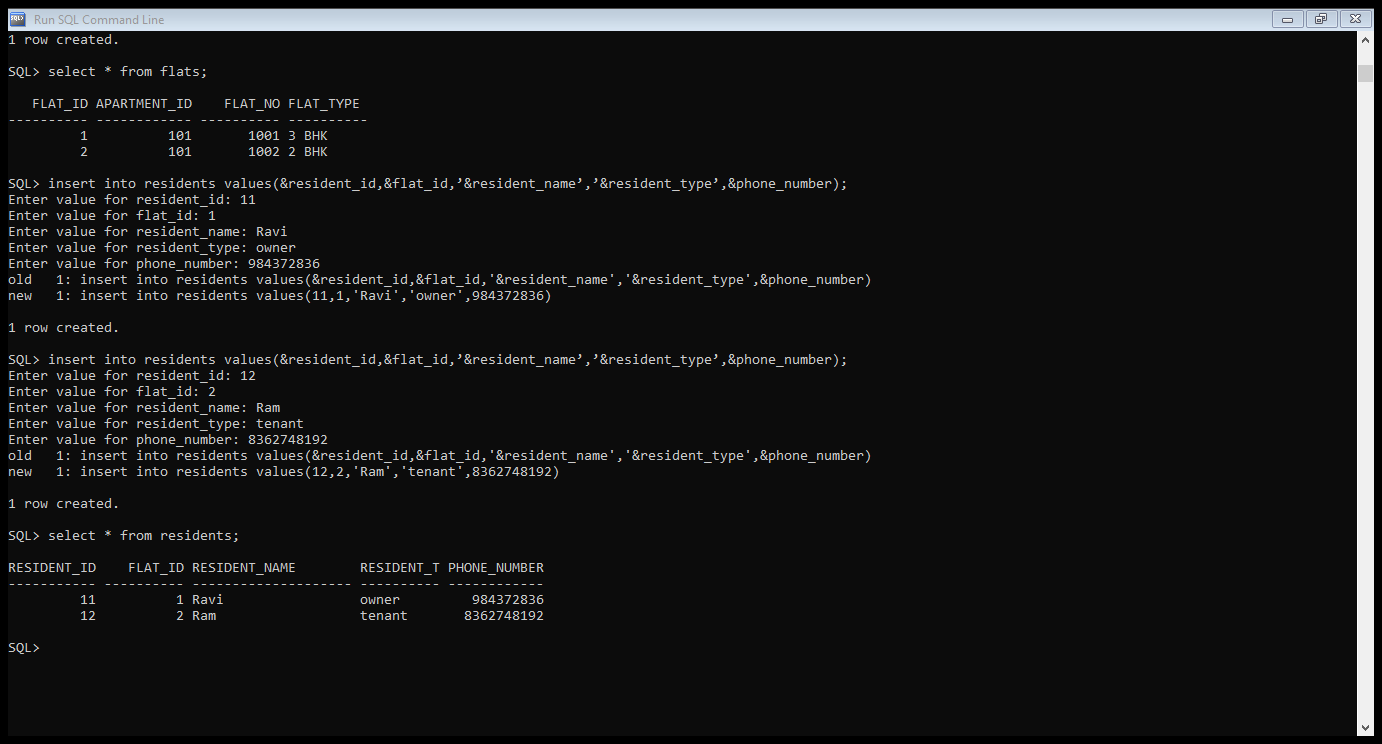
**DML COMMANDS:**

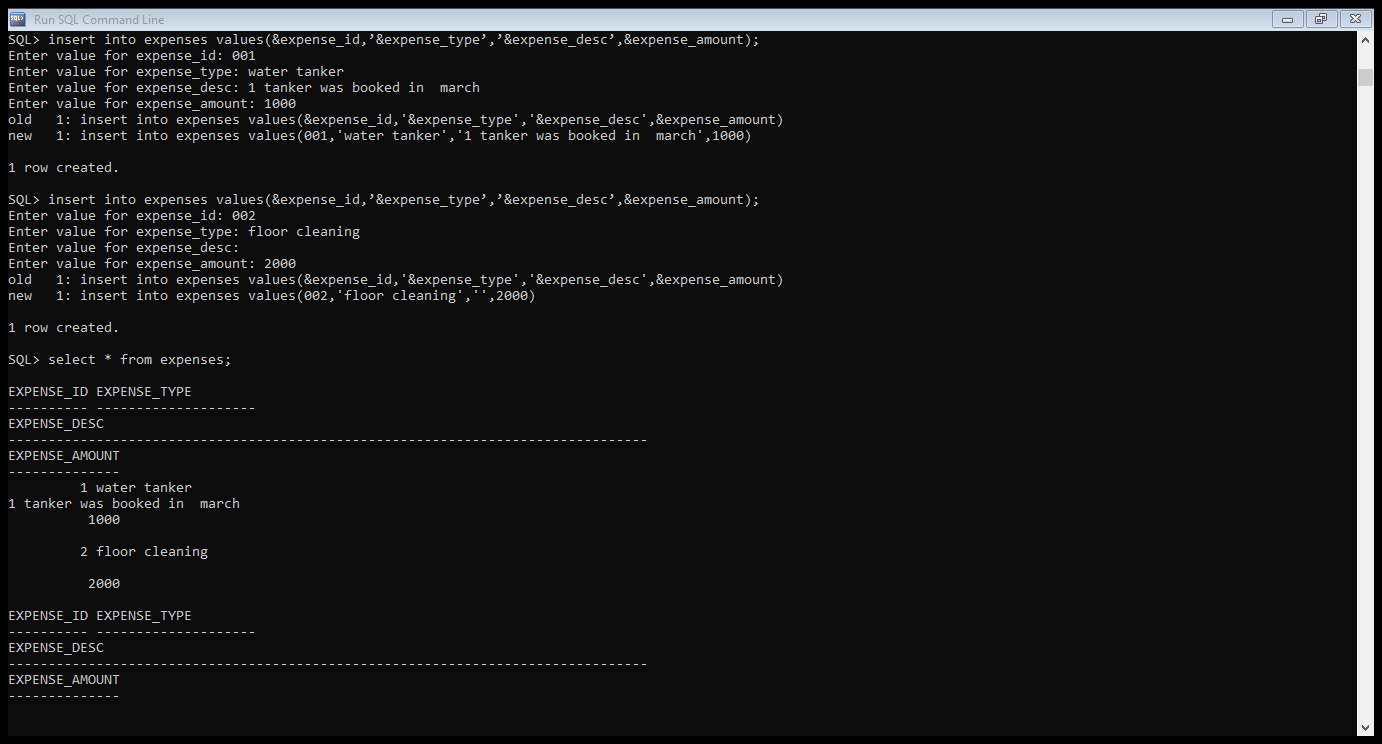


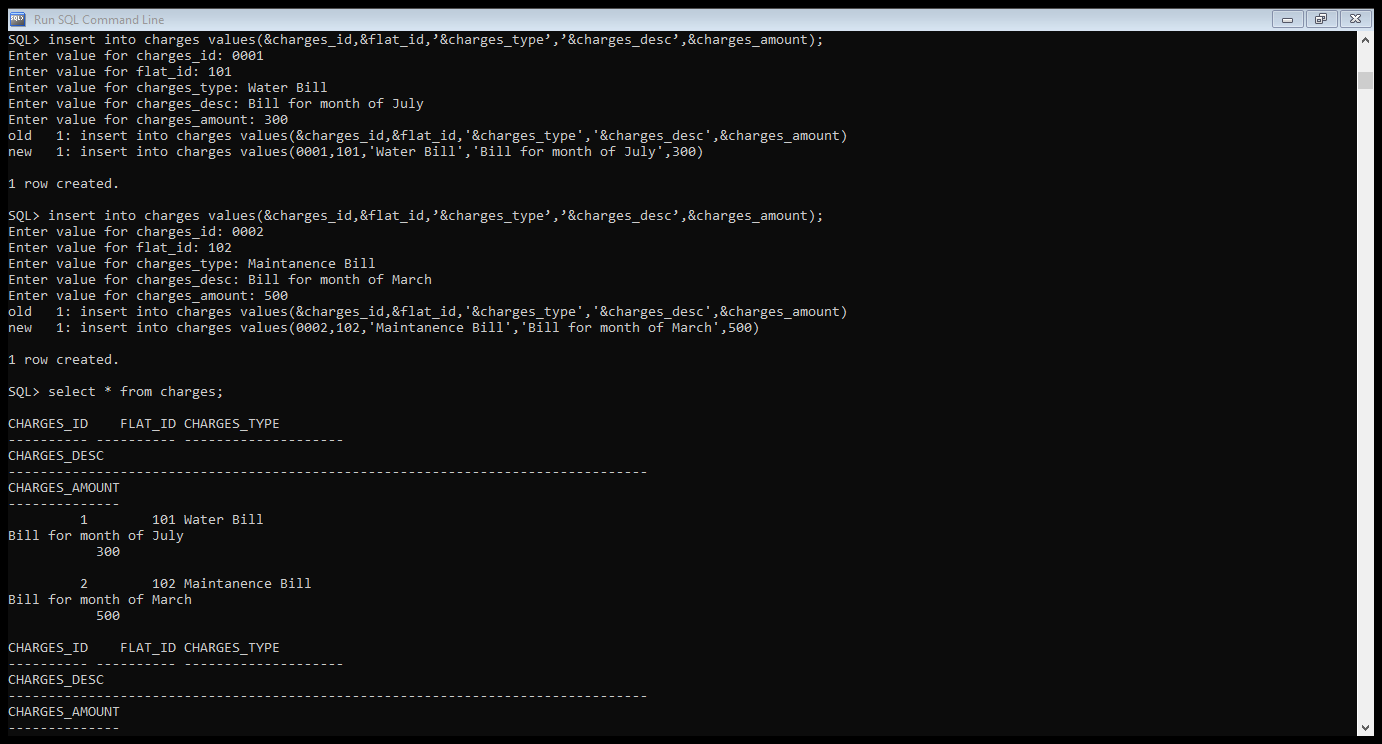
****

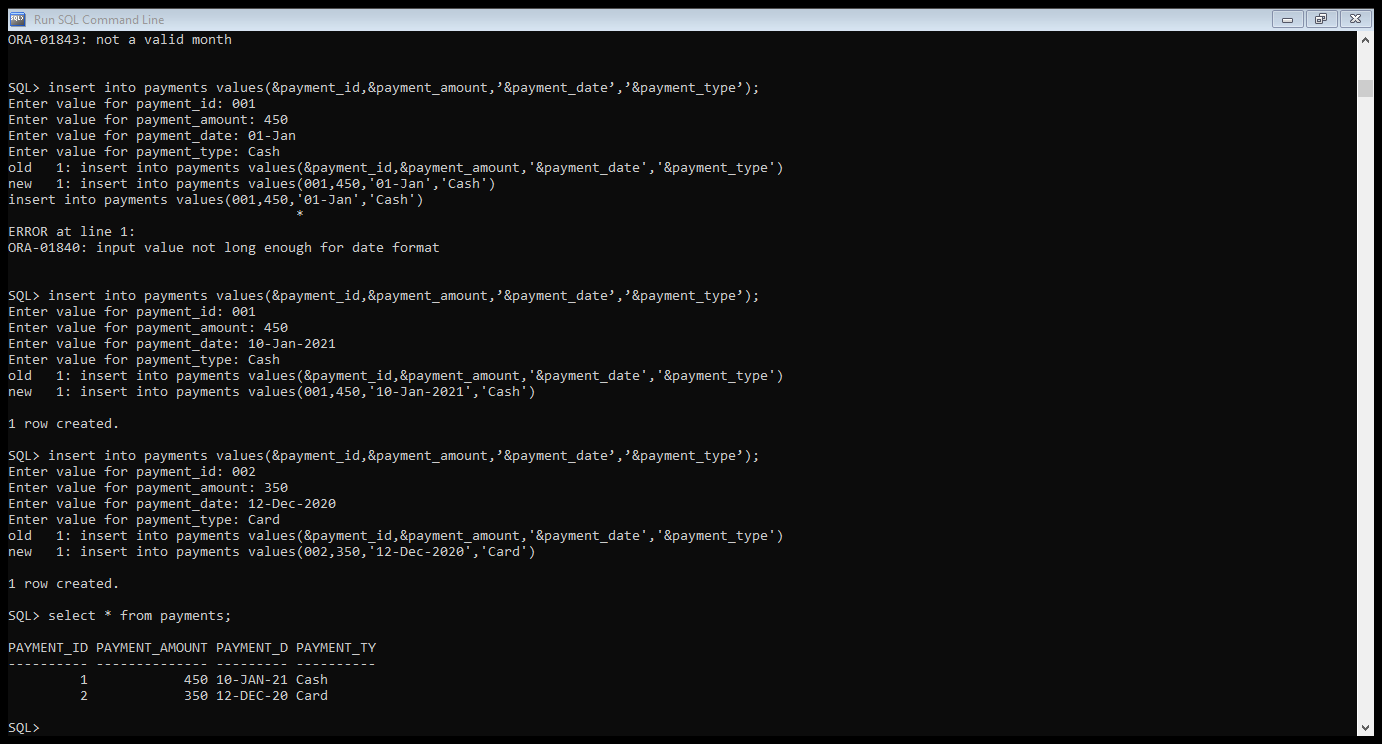
****

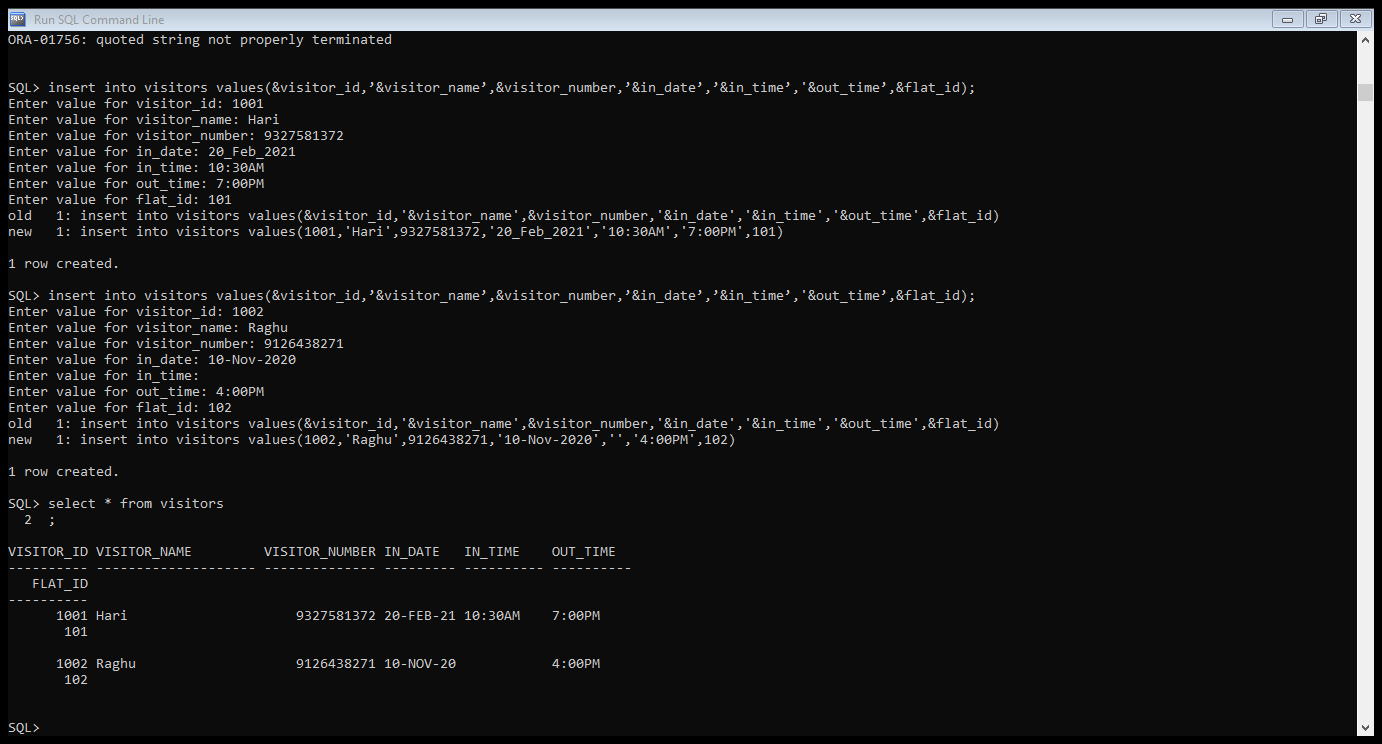
****

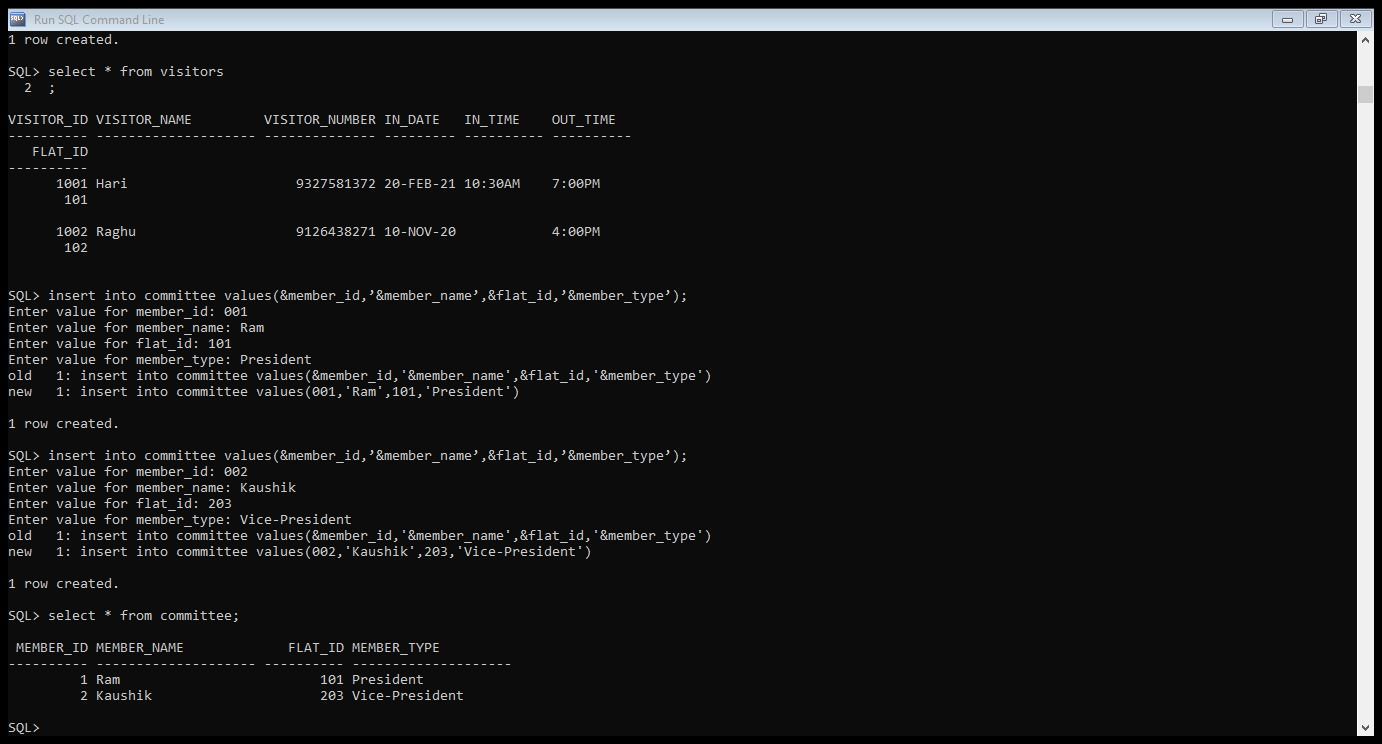
****

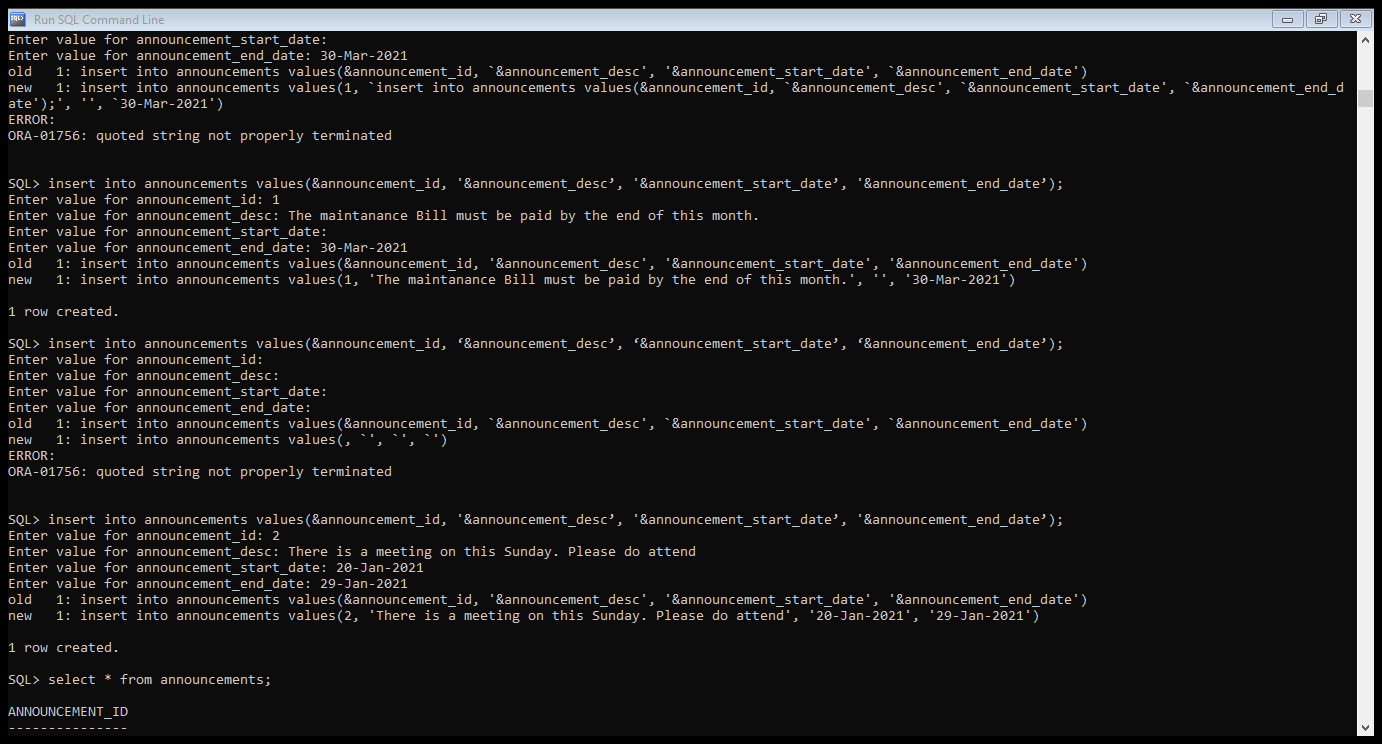
****

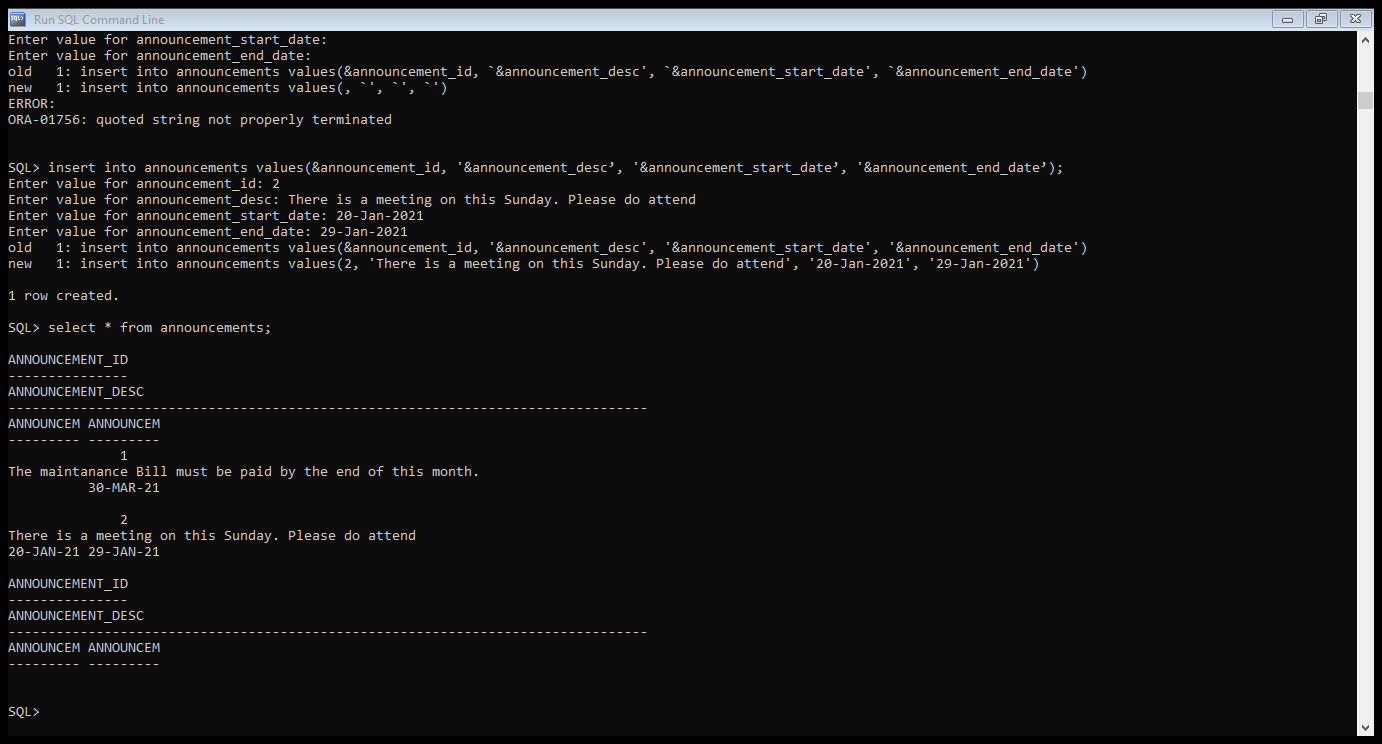
****

****

****

****

****

****

## IMPLEMENTATION

**Front end programs and its connectivity**

**Java Database Connectivity** (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

**public** **class** MyDbConnection {

**public** Connection getConnection() {

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection conn = DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:xe","Harsha","vasavi");

**return** conn;

}

**catch**(Exception e) {

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

}

**return** **null**;

}

}

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

# LOGINPAGE (MAIN METHOD):

**package** ams;

**import** java.awt.EventQueue;

**import** ams.\*;

**import** java.sql.\*;

**import** javax.sql.\*;

**import** javax.swing.\*;

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

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

**public** **class** login {

**private** JFrame frmApartmentManagementSystem;

**private** JTextField txtLoginId;

**private** JPasswordField txtPassword;

**public** **static** String *userName*;

**public** **static** String *loginId*;

**public** **static** String *userType*;

**public** **static** String *password*;

**public** **static** **int** *flatNo*;

/\*\*

\* Launch the application.

\*/

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

login window = **new** login();

window.frmApartmentManagementSystem.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

**public** login() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

**private** **void** initialize() {

frmApartmentManagementSystem = **new** JFrame();

frmApartmentManagementSystem.setTitle("Apartment Management System - Log In");

frmApartmentManagementSystem.setBounds(100, 100, 450, 300);

frmApartmentManagementSystem.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

frmApartmentManagementSystem.getContentPane().setLayout(**null**);

txtLoginId = **new** JTextField();

txtLoginId.setBounds(198, 47, 96, 20);

frmApartmentManagementSystem.getContentPane().add(txtLoginId);

txtLoginId.setColumns(10);

JLabel lblNewLabel = **new** JLabel("Login Id");

lblNewLabel.setBounds(76, 50, 84, 14);

frmApartmentManagementSystem.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = **new** JLabel("Password");

lblNewLabel\_1.setBounds(76, 88, 84, 14);

frmApartmentManagementSystem.getContentPane().add(lblNewLabel\_1);

JButton btnLogin = **new** JButton("Login");

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

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

**boolean** isValidUser=**false**;

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt = "select user\_name,login\_id,user\_type,password from users where login\_id = '"+txtLoginId.getText()+"' and password ='"+**new** String(txtPassword.getPassword())+"'";

//System.out.println(txt);

ResultSet rs=stmt.executeQuery(txt);

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

isValidUser = **true**;

*userName* = rs.getString(1);

*loginId* = rs.getString(2);

*userType* = rs.getString(3);

*password* = rs.getString(4);

}

rs.close();

**if**(isValidUser) {

txt = "select flat\_no from residents where resident\_name in( select user\_name from users where user\_name ='"+*userName*+"')";

ResultSet rs1 = stmt.executeQuery(txt);

**if**(rs1.next()) {

*flatNo* = rs1.getInt(1);

}

}

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

**if**(isValidUser) {

frmApartmentManagementSystem.setVisible(**false**);

home h = **new** home();

h.setVisible(**true**);

}

**else** {

JOptionPane.*showMessageDialog*(**null**,"Invalid User");

}

}

});

btnLogin.setBounds(150, 160, 89, 23);

frmApartmentManagementSystem.getContentPane().add(btnLogin);

txtPassword = **new** JPasswordField();

txtPassword.setBounds(198, 85, 96, 20);

frmApartmentManagementSystem.getContentPane().add(txtPassword);

}

}

# HOME:

**package** ams;

**import** java.awt.BorderLayout;

**import** java.awt.EventQueue;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.JLabel;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**import** javax.swing.AbstractAction;

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

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

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** javax.swing.Action;

**import** javax.swing.DefaultListModel;

**import** javax.swing.JButton;

**import** javax.swing.JTextField;

**import** javax.swing.SwingConstants;

**import** javax.swing.JList;

**import** javax.swing.event.MenuListener;

**import** javax.swing.event.MenuEvent;

**import** java.awt.Font;

**import** java.awt.Color;

**public** **class** home **extends** Menu {

**private** **final** Action action = **new** SwingAction();

**static** home *frame*;

/\*\*

\* Launch the application.

\*/

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

*frame* = **new** home();

*frame*.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** home() {

setTitle("Apartment Management System - Home");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 695, 523);

**try** {

DefaultListModel<String> l1 = **new** DefaultListModel<>();

l1.addElement(" Welcome "+login.*userName*+" ");

l1.addElement(" ");

l1.addElement(" ");

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="select announcement\_desc from announcements where sysdate between ANNOUNCEMENT\_START\_DATE and ANNOUNCEMENT\_END\_DATE";

ResultSet rs = stmt.executeQuery(txt);

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

l1.addElement(" - "+rs.getString(1));

System.***out***.println(" - "+rs.getString(1));

}

getContentPane().setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Apartment Management System - Home",SwingConstants.***CENTER***);

lblNewLabel.setForeground(Color.***BLACK***);

lblNewLabel.setBackground(**new** Color(227, 227, 227));

lblNewLabel.setOpaque(**true**);

lblNewLabel.setFont(**new** Font("Stencil", Font.***PLAIN***, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = **new** JLabel("Announcements");

lblNewLabel\_1.setBackground(Color.***WHITE***);

lblNewLabel\_1.setForeground(Color.***RED***);

lblNewLabel\_1.setFont(**new** Font("Lucida Handwriting", Font.***ITALIC***, 16));

lblNewLabel\_1.setBounds(10, 86, 168, 27);

getContentPane().add(lblNewLabel\_1);

JList<String> list = **new** JList<>(l1);

list.setVisible(**true**);

list.setBounds(0,60,679,524);

getContentPane().add(list);

}

**catch**(Exception e) {

e.printStackTrace();

}

}

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

}

**private** **class** SwingAction **extends** AbstractAction {

**public** SwingAction() {

putValue(***NAME***, "SwingAction");

putValue(***SHORT\_DESCRIPTION***, "Some short description");

}

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

}

}

}

# USERS:

package ams;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.ListSelectionModel;

import javax.swing.SwingConstants;

import javax.swing.border.EmptyBorder;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.DefaultTableModel;

import javax.swing.JLabel;

import javax.swing.JMenuBar;

import javax.swing.JMenu;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.AbstractAction;

import java.awt.event.ActionEvent;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import javax.swing.Action;

import javax.swing.JButton;

import javax.swing.event.MenuListener;

import javax.swing.event.MenuEvent;

import javax.swing.JPasswordField;

import javax.swing.JComboBox;

public class Users extends Menu {

private JPanel contentPane;

static Users frame;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new Users();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

JMenuItem mntmAddUserMenuItem;

JMenuItem mntmEditUserMenuItem;

JMenuItem mntmDeleteUserMenuItem;

JButton btnAddUserButton;

JButton btnDeleteUserButton;

JButton btnUpdateUserButton;

JButton btnClearButton;

JButton btnCancelButton;

JLabel lblNewLabel;

JLabel lblNewLabel\_1;

JLabel lblNewLabel\_2;

private JTextField textFieldUserName;

private JTextField textFieldLogInId;

private JPasswordField textFieldPassword;

private final Action action = new SwingAction();

private JTable table\_1;

private JScrollPane js = new JScrollPane(table\_1);

private JLabel lblNewLabel\_3;

JComboBox comboBoxUserType;

public Users() {

setTitle("ApartmentManagement System - Users");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 695, 523);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("Apartment Management System - Users",SwingConstants.CENTER);

lblNewLabel.setForeground(Color.BLACK);

lblNewLabel.setBackground(new Color(227, 227, 227));

lblNewLabel.setOpaque(true);

lblNewLabel.setFont(new Font("Stencil", Font.PLAIN, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

if(true) {

comboBoxUserType = new JComboBox();

comboBoxUserType.setBounds(320, 173, 152, 25);

contentPane.add(comboBoxUserType);

lblNewLabel\_3 = new JLabel("User Type");

lblNewLabel\_3.setBounds(163, 173, 120, 25);

contentPane.add(lblNewLabel\_3);

btnAddUserButton = new JButton("Add User");

btnAddUserButton.setBounds(256, 223, 96, 23);

getContentPane().add(btnAddUserButton);

btnAddUserButton.addActionListener(action);

btnCancelButton = new JButton("Cancel");

btnCancelButton.setBounds(157, 223, 89, 23);

contentPane.add(btnCancelButton);

btnCancelButton.addActionListener(action);

btnCancelButton.setVisible(false);

btnDeleteUserButton = new JButton("Delete User");

btnDeleteUserButton.setBounds(404, 223, 120, 23);

getContentPane().add(btnDeleteUserButton);

btnDeleteUserButton.addActionListener(action);

btnDeleteUserButton.setVisible(false);

btnUpdateUserButton = new JButton("Update User");

btnUpdateUserButton.setBounds(256, 223, 112, 23);

btnUpdateUserButton.setVisible(false);

getContentPane().add(btnUpdateUserButton);

btnUpdateUserButton.addActionListener(action);

btnClearButton = new JButton("Clear");

btnClearButton.setBounds(394, 223, 103, 23);

contentPane.add(btnClearButton);

btnClearButton.addActionListener(action);

try {

MyDbConnection connect = new MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select type from UserType";

ResultSet rs1 = stmt.executeQuery(txt);

while(rs1.next()) {

comboBoxUserType.addItem(rs1.getString(1));

}

txt = "select user\_name,login\_id,user\_type from users";

ResultSet rs = stmt.executeQuery(txt);

DefaultTableModel tableModel = new DefaultTableModel() ;

table\_1 = new JTable(tableModel);

tableModel.addColumn("User Name");

tableModel.addColumn("Login Id");

tableModel.addColumn("User Type");

int i=0;

while(rs.next()) {

tableModel.insertRow(i,new Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.out.println(rs.getString(2));

i++;

}

table\_1.setVisible(true);

js.setBounds(20,290,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent e) {

if(!model.isSelectionEmpty()) {

int selectedRow = model.getMinSelectionIndex();

btnAddUserButton.setVisible(false);

btnUpdateUserButton.setVisible(true);

btnDeleteUserButton.setVisible(true);

btnCancelButton.setVisible(true);

btnClearButton.setVisible(false);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

try {

String txt = "select user\_name,login\_id,password from users where login\_id = '"+table\_1.getModel().getValueAt(selectedRow,1)+"'";

ResultSet rs = stmt.executeQuery(txt);

while(rs.next()) {

textFieldUserName.setText(rs.getString(1));

textFieldLogInId.setText(rs.getString(2));

textFieldPassword.setText(rs.getString(3));

}

}

catch(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

catch(Exception exe) {

exe.printStackTrace();

}

}

lblNewLabel = new JLabel("UserName");

lblNewLabel.setBounds(163, 86, 120, 25);

getContentPane().add(lblNewLabel);

lblNewLabel\_1 = new JLabel("LogIn Id");

lblNewLabel\_1.setBounds(163, 115, 120, 25);

getContentPane().add(lblNewLabel\_1);

lblNewLabel\_2 = new JLabel("Password");

lblNewLabel\_2.setBounds(163, 144, 120, 25);

getContentPane().add(lblNewLabel\_2);

textFieldUserName = new JTextField();

textFieldUserName.setBounds(320, 86, 152, 25);

getContentPane().add(textFieldUserName);

textFieldUserName.setColumns(10);

textFieldLogInId = new JTextField();

textFieldLogInId.setBounds(320, 115, 152, 25);

getContentPane().add(textFieldLogInId);

textFieldLogInId.setColumns(10);

textFieldPassword = new JPasswordField();

textFieldPassword.setBounds(320, 144, 152, 25);

getContentPane().add(textFieldPassword);

textFieldPassword.setColumns(10);

if(login.userType.equals("user")) {

textFieldLogInId.setEditable(false);

btnUpdateUserButton = new JButton("Update User");

btnUpdateUserButton.setBounds(195, 160, 112, 23);

btnUpdateUserButton.setVisible(true);

getContentPane().add(btnUpdateUserButton);

btnUpdateUserButton.addActionListener(action);

textFieldUserName.setText(login.userName);

textFieldLogInId.setText(login.loginId);

textFieldPassword.setText(login.password);

}

}

private class SwingAction extends AbstractAction {

public SwingAction() {

putValue(NAME, "Add User");

putValue(SHORT\_DESCRIPTION, "Some short description");

}

public void actionPerformed(ActionEvent e) {

try {

MyDbConnection connect = new MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

if(e.getSource()==btnAddUserButton) {

txt = "insert into users values('"+textFieldUserName.getText()+"','"+textFieldLogInId.getText()+"','"+textFieldPassword.getText()+"','"+comboBoxUserType.getItemAt(comboBoxUserType.getSelectedIndex())+"')";

stmt.executeQuery(txt);

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

textFieldUserName.setText("");

textFieldLogInId.setText("");

textFieldPassword.setText("");

}

else if(e.getSource()==btnDeleteUserButton) {

txt = "delete from users where user\_name='"+textFieldUserName.getText()+"' or login\_id='"+textFieldLogInId.getText()+"'";

stmt.executeQuery(txt);

JOptionPane.showMessageDialog(null,"User Deleted");

textFieldUserName.setText("");

textFieldLogInId.setText("");

textFieldPassword.setText("");

}

else if(e.getSource()==btnUpdateUserButton) {

if(login.userType.equals("user")) {

txt = "update users set user\_name='"+textFieldUserName.getText()+"',login\_id='"+textFieldLogInId.getText()+"',password = '"+textFieldPassword.getText()+"' where login\_id='"+textFieldLogInId.getText()+"'";

stmt.executeQuery(txt);

}

else if(login.userType.equals("admin")) {

txt = "update users set user\_name='"+textFieldUserName.getText()+"',login\_id='"+textFieldLogInId.getText()+"',password = '"+textFieldPassword.getText()+"',user\_type = '"+comboBoxUserType.getItemAt(comboBoxUserType.getSelectedIndex())+"' where login\_id='"+textFieldLogInId.getText()+"'";

stmt.executeQuery(txt);

}

JOptionPane.showMessageDialog(null,"User Updated");

textFieldUserName.setText("");

textFieldLogInId.setText("");

textFieldPassword.setText("");

}

else if(e.getSource()==btnCancelButton) {

btnAddUserButton.setVisible(true);

btnUpdateUserButton.setVisible(false);

btnDeleteUserButton.setVisible(false);

btnCancelButton.setVisible(false);

textFieldUserName.setText("");

textFieldLogInId.setText("");

textFieldPassword.setText("");

btnClearButton.setVisible(true);

}

else if(e.getSource()==btnClearButton) {

textFieldUserName.setText("");

textFieldLogInId.setText("");

textFieldPassword.setText("");

}

if(login.userType.equals("admin")){

txt = "select user\_name,login\_id,user\_type from users";

}

else if(login.userType.equals("user")){

txt = "select user\_name,login\_id,user\_type from users where login\_id = '"+login.loginId+"'";

}

ResultSet rs = stmt.executeQuery(txt);

DefaultTableModel tableModel = new DefaultTableModel() ;

table\_1 = new JTable(tableModel);

tableModel.addColumn("User Name");

tableModel.addColumn("Login Id");

tableModel.addColumn("User Type");

int i=0;

while(rs.next()) {

tableModel.insertRow(i,new Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.out.println(rs.getString(2));

i++;

}

table\_1.setVisible(true);

js.setBounds(20,290,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent e) {

if(!model.isSelectionEmpty()) {

int selectedRow = model.getMinSelectionIndex();

if(login.userType.equals("admin")){

btnAddUserButton.setVisible(false);

btnUpdateUserButton.setVisible(true);

btnDeleteUserButton.setVisible(true);

btnCancelButton.setVisible(true);

btnClearButton.setVisible(false);

}

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

try {

String txt = "select user\_name,login\_id,password from users where login\_id = '"+table\_1.getModel().getValueAt(selectedRow,1)+"'";

ResultSet rs = stmt.executeQuery(txt);

while(rs.next()) {

textFieldUserName.setText(rs.getString(1));

textFieldLogInId.setText(rs.getString(2));

textFieldPassword.setText(rs.getString(3));

}

}

catch(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

catch(Exception ex) {

System.out.println(ex);

JOptionPane.showMessageDialog(null,"Invalid Input. Please try again");

}

}

}

}

# ANNOUNCEMENTS:

**package** ams;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.sql.Date;

**import** java.awt.EventQueue;

**import** java.awt.Font;

**import** java.awt.Rectangle;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**import** javax.swing.AbstractAction;

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

**import** javax.swing.Action;

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

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** java.text.SimpleDateFormat;

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

**import** javax.swing.event.ListSelectionEvent;

**import** javax.swing.event.ListSelectionListener;

**import** javax.swing.event.MenuEvent;

**import** javax.swing.event.MenuListener;

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

**import** com.github.lgooddatepicker.components.DatePicker;

**import** com.github.lgooddatepicker.components.DateTimePicker;

**import** javax.swing.JLabel;

**import** javax.swing.JTextField;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.SwingConstants;

**import** javax.swing.JButton;

**import** javax.swing.UIManager;

**public** **class** Announcements **extends** Menu {

**private** JPanel contentPane;

**private** **final** Action action = **new** SwingAction();

**static** Announcements *frame*;

/\*\*

\* Launch the application.

\*/

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

*frame* = **new** Announcements();

*frame*.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

**private** JTextField textFieldAnnouncementDescription;

JLabel lblNewLabel;

JLabel lblNewLabel\_1;

JLabel lblNewLabel\_2;

JButton btnAddAnnouncementButton;

JButton btnUpdateAnnouncementButton;

JButton btnDeleteAnnouncementButton;

JButton btnCancelButton;

JButton btnClearButton;

DateTimePicker startDate;

DateTimePicker endDate;

**private** JTable table\_1;

**private** JScrollPane js = **new** JScrollPane(table\_1);

**private** JLabel lblNewLabel\_3;

**private** JTextField textFieldAnnouncementId;

**public** Announcements() {

setTitle("ApartmentManagement System - Announcements");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 695, 523);

**if**(login.*userName*.isEmpty()) {

dispose();

}

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Apartment Management System - Announcements",SwingConstants.***CENTER***);

lblNewLabel.setForeground(Color.***BLACK***);

lblNewLabel.setBackground(**new** Color(227, 227, 227));

lblNewLabel.setOpaque(**true**);

lblNewLabel.setFont(**new** Font("Stencil", Font.***PLAIN***, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

lblNewLabel\_3 = **new** JLabel("Announcement Id");

lblNewLabel\_3.setBounds(132, 67, 178, 25);

contentPane.add(lblNewLabel\_3);

lblNewLabel = **new** JLabel("Announcement Description");

lblNewLabel.setBounds(132, 96, 178, 25);

contentPane.add(lblNewLabel);

lblNewLabel\_1 = **new** JLabel("Start Date");

lblNewLabel\_1.setBounds(132, 125, 178, 25);

contentPane.add(lblNewLabel\_1);

lblNewLabel\_2 = **new** JLabel("End Date");

lblNewLabel\_2.setBounds(132, 154, 178, 25);

contentPane.add(lblNewLabel\_2);

textFieldAnnouncementId = **new** JTextField();

textFieldAnnouncementId.setBounds(320, 67, 152, 25);

contentPane.add(textFieldAnnouncementId);

textFieldAnnouncementId.setColumns(10);

textFieldAnnouncementId.setEditable(**false**);

textFieldAnnouncementDescription = **new** JTextField();

textFieldAnnouncementDescription.setBounds(320, 96, 250, 25);

contentPane.add(textFieldAnnouncementDescription);

textFieldAnnouncementDescription.setColumns(10);

startDate = **new** DateTimePicker();

startDate.setBounds(320, 125, 250, 25);

getContentPane().add(startDate);

endDate = **new** DateTimePicker();

endDate.setBounds(320, 154, 250, 25);

getContentPane().add(endDate);

btnAddAnnouncementButton = **new** JButton("Add");

btnAddAnnouncementButton.setBounds(254, 207, 89, 23);

contentPane.add(btnAddAnnouncementButton);

btnAddAnnouncementButton.setVisible(**true**);

btnAddAnnouncementButton.addActionListener(action);

btnUpdateAnnouncementButton = **new** JButton("Update");

btnUpdateAnnouncementButton.setBounds(254, 207, 96, 23);

contentPane.add(btnUpdateAnnouncementButton);

btnUpdateAnnouncementButton.addActionListener(action);

btnUpdateAnnouncementButton.setVisible(**false**);

btnDeleteAnnouncementButton = **new** JButton("Delete");

btnDeleteAnnouncementButton.setBounds(429, 207, 105, 23);

contentPane.add(btnDeleteAnnouncementButton);

btnDeleteAnnouncementButton.addActionListener(action);

btnDeleteAnnouncementButton.setVisible(**false**);

btnCancelButton = **new** JButton("Cancel");

btnCancelButton.setBounds(132, 207, 89, 23);

contentPane.add(btnCancelButton);

btnCancelButton.addActionListener(action);

btnCancelButton.setVisible(**false**);

btnClearButton = **new** JButton("Clear");

btnClearButton.setBounds(411, 207, 103, 23);

contentPane.add(btnClearButton);

btnClearButton.addActionListener(action);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select announcement\_id,announcement\_desc from announcements";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Announcement Id");

tableModel.addColumn("Announcement Description");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,290,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddAnnouncementButton.setVisible(**false**);

btnUpdateAnnouncementButton.setVisible(**true**);

btnDeleteAnnouncementButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select announcement\_id,announcement\_desc,Announcement\_start\_date,Announcement\_end\_date from announcements where announcement\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldAnnouncementId.setText(Integer.*toString*(rs.getInt(1)));

textFieldAnnouncementDescription.setText(rs.getString(2));

startDate.setDateTimeStrict(rs.getTimestamp(3).toLocalDateTime());

endDate.setDateTimeStrict(rs.getTimestamp(4).toLocalDateTime());

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

}

**void** clearText() {

textFieldAnnouncementId.setText("");

textFieldAnnouncementDescription.setText("");

startDate.setDateTimeStrict(**null**);

endDate.setDateTimeStrict(**null**);

}

**private** **class** SwingAction **extends** AbstractAction {

**public** SwingAction() {

putValue(***NAME***, "SwingAction");

putValue(***SHORT\_DESCRIPTION***, "Some short description");

}

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

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

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

txt = "insert into announcements(Announcement\_id,Announcement\_desc,Announcement\_start\_date,Announcement\_end\_date) values('"+textFieldAnnouncementId.getText()+"','"+textFieldAnnouncementDescription.getText()+"',to\_date('"+startDate.getDateTimeStrict().format(DateTimeFormatter.*ofPattern*("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi'),to\_date('"+endDate.getDateTimeStrict().format(DateTimeFormatter.*ofPattern*("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi'))";

System.***out***.println(txt);

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Added Announcement");

clearText();

}

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

txt = "delete from announcements where Announcement\_desc='"+textFieldAnnouncementDescription.getText()+"' ";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Announcement Deleted");

clearText();

}

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

txt = "update announcements set Announcement\_id ='"+Integer.*parseInt*(textFieldAnnouncementId.getText())+"', Announcement\_desc='"+textFieldAnnouncementDescription.getText()+"',Announcement\_start\_date= to\_date('"+startDate.getDateTimeStrict().format(DateTimeFormatter.*ofPattern*("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi'),Announcement\_end\_date = to\_date('"+endDate.getDateTimeStrict().format(DateTimeFormatter.*ofPattern*("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi') where Announcement\_id='"+textFieldAnnouncementId.getText()+"' ";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Announcement Updated");

clearText();

}

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

btnAddAnnouncementButton.setVisible(**true**);

btnUpdateAnnouncementButton.setVisible(**false**);

btnDeleteAnnouncementButton.setVisible(**false**);

btnCancelButton.setVisible(**false**);

clearText();

btnClearButton.setVisible(**true**);

}

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

clearText();

}

txt = "select announcement\_id,announcement\_desc from announcements";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Announcement Id");

tableModel.addColumn("Announcement Description");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,290,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddAnnouncementButton.setVisible(**false**);

btnUpdateAnnouncementButton.setVisible(**true**);

btnDeleteAnnouncementButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select announcement\_id,announcement\_desc,Announcement\_start\_date,Announcement\_end\_date from announcements where announcement\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldAnnouncementId.setText(Integer.*toString*(rs.getInt(1)));

textFieldAnnouncementDescription.setText(rs.getString(2));

startDate.setDateTimeStrict(rs.getTimestamp(3).toLocalDateTime());

endDate.setDateTimeStrict(rs.getTimestamp(4).toLocalDateTime());

}

}

**catch**(Exception exe) {

exe.printStackTrace();

JOptionPane.*showMessageDialog*(**null**,"Table error");

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

JOptionPane.*showMessageDialog*(**null**,"Invalid Input. Please try again");

}

}

}

}

# VISITORS:

package ams;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.text.SimpleDateFormat;

import java.time.format.DateTimeFormatter;

import javax.swing.AbstractAction;

import javax.swing.JFrame;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.border.EmptyBorder;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.event.MenuEvent;

import javax.swing.event.MenuListener;

import javax.swing.table.DefaultTableModel;

import com.github.lgooddatepicker.components.DatePicker;

import com.github.lgooddatepicker.components.DateTimePicker;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.ListSelectionModel;

import javax.swing.SwingConstants;

import javax.swing.JButton;

import javax.swing.Action;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

public class Visitors extends Menu {

private JPanel contentPane;

static Visitors frame;

private JTextField textFieldVisitorId;

private JTextField textFieldVisitorName;

private JTextField textFieldNumber;

private JTextField textFieldFlatId;

private final Action action = new SwingAction();

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new Visitors();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

JButton btnAddVisitorButton;

JButton btnUpdateVisitorButton;

JButton btnDeleteVisitorButton;

JButton btnClearButton;

JButton btnCancelButton;

JLabel lblNewLabel;

JLabel lblNewLabel\_1;

JLabel lblNewLabel\_2;

JLabel lblNewLabel\_3;

JLabel lblNewLabel\_4;

JLabel lblNewLabel\_5;

JLabel lblNewLabel\_6;

DateTimePicker inDateTime;

DateTimePicker outDateTime;

JTable table\_1;

private JScrollPane js = new JScrollPane(table\_1);

public Visitors() {

setTitle("Apartment Management System - Visitors");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 695, 523);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

if(login.userName.isEmpty()) {

dispose();

}

JLabel lblNewLabel = new JLabel("Apartment Management System - Visitors",SwingConstants.CENTER);

lblNewLabel.setForeground(Color.BLACK);

lblNewLabel.setBackground(new Color(227, 227, 227));

lblNewLabel.setOpaque(true);

lblNewLabel.setFont(new Font("Stencil", Font.PLAIN, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

lblNewLabel = new JLabel("Visitor Id");

lblNewLabel.setBounds(108, 68, 152, 25);

contentPane.add(lblNewLabel);

lblNewLabel\_1 = new JLabel("Visitor Name");

lblNewLabel\_1.setBounds(108, 97, 152, 25);

contentPane.add(lblNewLabel\_1);

lblNewLabel\_2 = new JLabel("Visitor Number");

lblNewLabel\_2.setBounds(108, 126, 152, 25);

contentPane.add(lblNewLabel\_2);

lblNewLabel\_3 = new JLabel("Visitor In Date and Time");

lblNewLabel\_3.setBounds(108, 155, 152, 25);

contentPane.add(lblNewLabel\_3);

lblNewLabel\_4 = new JLabel("Visitor Out Date and Time");

lblNewLabel\_4.setBounds(108, 184, 152, 25);

contentPane.add(lblNewLabel\_4);

lblNewLabel\_6 = new JLabel("Flat Id");

lblNewLabel\_6.setBounds(108, 213, 152, 25);

contentPane.add(lblNewLabel\_6);

textFieldVisitorId = new JTextField();

textFieldVisitorId.setBounds(270, 68, 161, 25);

contentPane.add(textFieldVisitorId);

textFieldVisitorId.setColumns(10);

textFieldVisitorId.setEditable(false);

textFieldVisitorName = new JTextField();

textFieldVisitorName.setBounds(270, 97, 161, 25);

contentPane.add(textFieldVisitorName);

textFieldVisitorName.setColumns(10);

textFieldNumber = new JTextField();

textFieldNumber.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

if(textFieldNumber.getText().length()>=10&&!(evt.getKeyChar()==KeyEvent.VK\_DELETE||evt.getKeyChar()==KeyEvent.VK\_BACK\_SPACE)||!(evt.getKeyChar()>= '0' &&evt.getKeyChar()<= '9')) {

//getToolkit().beep();

evt.consume();

}

}

});

textFieldNumber.setBounds(270, 126, 161, 25);

contentPane.add(textFieldNumber);

textFieldNumber.setColumns(10);

inDateTime = new DateTimePicker();

inDateTime.setBounds(270, 155, 250, 25);

contentPane.add(inDateTime);

textFieldFlatId = new JTextField();

textFieldFlatId.setBounds(270, 213, 161, 25);

contentPane.add(textFieldFlatId);

textFieldFlatId.setColumns(10);

outDateTime = new DateTimePicker();

outDateTime.setBounds(270, 184, 250, 25);

contentPane.add(outDateTime);

btnAddVisitorButton = new JButton("Add");

btnAddVisitorButton.setBounds(249, 266, 89, 23);

contentPane.add(btnAddVisitorButton);

btnAddVisitorButton.addActionListener(action);

btnAddVisitorButton.setVisible(true);

btnUpdateVisitorButton = new JButton("Update");

btnUpdateVisitorButton.setBounds(249, 266, 105, 23);

contentPane.add(btnUpdateVisitorButton);

btnUpdateVisitorButton.addActionListener(action);

btnUpdateVisitorButton.setVisible(false);

btnDeleteVisitorButton = new JButton("Delete");

btnDeleteVisitorButton.setBounds(401, 266, 119, 23);

contentPane.add(btnDeleteVisitorButton);

btnDeleteVisitorButton.addActionListener(action);

btnDeleteVisitorButton.setVisible(false);

btnCancelButton = new JButton("Cancel");

btnCancelButton.setBounds(131, 266, 89, 23);

contentPane.add(btnCancelButton);

btnCancelButton.addActionListener(action);

btnCancelButton.setVisible(false);

btnClearButton = new JButton("Clear");

btnClearButton.setBounds(375, 266, 103, 23);

contentPane.add(btnClearButton);

btnClearButton.addActionListener(action);

try {

MyDbConnection connect = new MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select visitor\_id,visitor\_name,visitor\_number,in\_date from visitors";

ResultSet rs = stmt.executeQuery(txt);

DefaultTableModel tableModel = new DefaultTableModel() ;

table\_1 = new JTable(tableModel);

tableModel.addColumn("Visitor Id");

tableModel.addColumn("Visitor Name");

tableModel.addColumn("Visitor Number");

int i=0;

while(rs.next()) {

tableModel.insertRow(i,new Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.out.println(rs.getString(2));

i++;

}

table\_1.setVisible(true);

js.setBounds(20,308,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent e) {

if(!model.isSelectionEmpty()) {

int selectedRow = model.getMinSelectionIndex();

btnAddVisitorButton.setVisible(false);

btnUpdateVisitorButton.setVisible(true);

btnDeleteVisitorButton.setVisible(true);

btnCancelButton.setVisible(true);

btnClearButton.setVisible(false);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

try {

String txt = "select visitor\_id,visitor\_name,visitor\_number,in\_date,out\_date,flat\_id from visitors where visitor\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

while(rs.next()) {

textFieldVisitorId.setText(Integer.toString(rs.getInt(1)));

textFieldVisitorName.setText(rs.getString(2));

textFieldNumber.setText(rs.getString(3));

inDateTime.setDateTimeStrict(rs.getTimestamp(4).toLocalDateTime());

inDateTime.setDateTimeStrict(rs.getTimestamp(5).toLocalDateTime());

textFieldFlatId.setText(Integer.toString(rs.getInt(6)));

}

}

catch(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

catch(Exception ex) {

System.out.println(ex);

}

}

void clearText() {

textFieldVisitorId.setText("");

textFieldVisitorName.setText("");

textFieldNumber.setText("");

inDateTime.setDateTimeStrict(null);

outDateTime.setDateTimeStrict(null);

textFieldFlatId.setText("");

}

private class SwingAction extends AbstractAction {

public SwingAction() {

putValue(NAME, "SwingAction");

putValue(SHORT\_DESCRIPTION, "Some short description");

}

public void actionPerformed(ActionEvent e) {

try {

MyDbConnection connect = new MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

if(e.getSource()==btnAddVisitorButton) {

txt = "insert into visitors(visitor\_id,visitor\_name,visitor\_number,in\_date,out\_date,flat\_id) values('"+textFieldVisitorId.getText()+"','"+textFieldVisitorName.getText()+"','"+textFieldNumber.getText()+"',to\_date('"+inDateTime.getDateTimeStrict().format(DateTimeFormatter.ofPattern("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi'),to\_date('"+outDateTime.getDateTimeStrict().format(DateTimeFormatter.ofPattern("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi'),'"+textFieldFlatId.getText()+"')";

stmt.executeQuery(txt);

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

clearText();

}

else if(e.getSource()==btnDeleteVisitorButton) {

txt = "delete from visitors where visitor\_id='"+textFieldVisitorId.getText()+"' ";

stmt.executeQuery(txt);

JOptionPane.showMessageDialog(null,"Visitor Deleted");

clearText();

}

else if(e.getSource()==btnUpdateVisitorButton) {

txt = "update visitors set visitor\_id ='"+Integer.parseInt(textFieldVisitorId.getText())+"', visitor\_name='"+textFieldVisitorName.getText()+"',visitor\_number='"+textFieldNumber.getText()+"',in\_date = to\_date('"+inDateTime.getDateTimeStrict().format(DateTimeFormatter.ofPattern("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi'),out\_date = to\_date('"+outDateTime.getDateTimeStrict().format(DateTimeFormatter.ofPattern("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi'),flat\_id = '"+Integer.parseInt(textFieldFlatId.getText())+"' where visitor\_id='"+textFieldVisitorId.getText()+"' ";

stmt.executeQuery(txt);

JOptionPane.showMessageDialog(null,"Visitor Updated");

clearText();

}

else if(e.getSource()==btnCancelButton) {

btnAddVisitorButton.setVisible(true);

btnUpdateVisitorButton.setVisible(false);

btnDeleteVisitorButton.setVisible(false);

btnCancelButton.setVisible(false);

clearText();

}

else if(e.getSource()==btnClearButton) {

clearText();

}

txt = "select visitor\_id,visitor\_name,visitor\_number,in\_date from visitors";

ResultSet rs = stmt.executeQuery(txt);

DefaultTableModel tableModel = new DefaultTableModel() ;

table\_1 = new JTable(tableModel);

tableModel.addColumn("Visitor Id");

tableModel.addColumn("Visitor Name");

tableModel.addColumn("Visitor Number");

int i=0;

while(rs.next()) {

tableModel.insertRow(i,new Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.out.println(rs.getString(2));

i++;

}

table\_1.setVisible(true);

js.setBounds(20,308,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent e) {

if(!model.isSelectionEmpty()) {

int selectedRow = model.getMinSelectionIndex();

btnAddVisitorButton.setVisible(false);

btnUpdateVisitorButton.setVisible(true);

btnDeleteVisitorButton.setVisible(true);

btnCancelButton.setVisible(true);

btnClearButton.setVisible(false);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

try {

String txt = "select visitor\_id,visitor\_name,visitor\_number,in\_date,out\_date,flat\_id from visitors where visitor\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

while(rs.next()) {

textFieldVisitorId.setText(Integer.toString(rs.getInt(1)));

textFieldVisitorName.setText(rs.getString(2));

textFieldNumber.setText(rs.getString(3));

inDateTime.setDateTimeStrict(rs.getTimestamp(4).toLocalDateTime());

inDateTime.setDateTimeStrict(rs.getTimestamp(5).toLocalDateTime());

textFieldFlatId.setText(Integer.toString(rs.getInt(6)));

}

}

catch(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

catch(Exception ex) {

System.out.println(ex);

}

}

}

}

# Expenses:

**package** ams;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.EventQueue;

**import** java.awt.Font;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.event.ListSelectionEvent;

**import** javax.swing.event.ListSelectionListener;

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

**import** javax.swing.JMenuBar;

**import** javax.swing.JLabel;

**import** javax.swing.JTextField;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.SwingConstants;

**import** javax.swing.JMenu;

**import** javax.swing.AbstractAction;

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

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** javax.swing.Action;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**import** javax.swing.JButton;

**import** javax.swing.event.MenuListener;

**import** javax.swing.event.MenuEvent;

**public** **class** Expenses **extends** Menu {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

Expenses frame = **new** Expenses();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

JLabel lblNewLabel;

**private** JLabel lblNewLabel\_1;

**private** JLabel lblNewLabel\_2;

**private** JLabel lblNewLabel\_3;

**private** JTextField textFieldExpenseId;

**private** JTextField textFieldExpenseType;

**private** JTextField textFieldExpenseDescription;

**private** JTextField textFieldExpenseAmount;

JButton btnClearButton;

JButton btnCancelButton;

**private** **final** Action action = **new** SwingAction();

**private** JButton btnAddExpensesButton;

**private** JButton btnUpdateExpensesButton;

**private** JButton btnDeleteExpensesButton;

**private** JTable table\_1;

**private** JScrollPane js = **new** JScrollPane(table\_1);

**public** Expenses() {

setTitle("Apartment Management System - Expenses");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 695, 523);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Apartment Management System - Expenses",SwingConstants.***CENTER***);

lblNewLabel.setForeground(Color.***BLACK***);

lblNewLabel.setBackground(**new** Color(227, 227, 227));

lblNewLabel.setOpaque(**true**);

lblNewLabel.setFont(**new** Font("Stencil", Font.***PLAIN***, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

lblNewLabel = **new** JLabel("Expense Id");

lblNewLabel.setBounds(163, 75, 120, 25);

contentPane.add(lblNewLabel);

lblNewLabel\_1 = **new** JLabel("Expense Type");

lblNewLabel\_1.setBounds(163, 104, 120, 25);

contentPane.add(lblNewLabel\_1);

lblNewLabel\_2 = **new** JLabel("Expense description");

lblNewLabel\_2.setBounds(163, 133, 120, 25);

contentPane.add(lblNewLabel\_2);

lblNewLabel\_3 = **new** JLabel("Expense Amount");

lblNewLabel\_3.setBounds(163, 162, 120, 25);

contentPane.add(lblNewLabel\_3);

textFieldExpenseId = **new** JTextField();

textFieldExpenseId.setBounds(320, 75, 152, 25);

contentPane.add(textFieldExpenseId);

textFieldExpenseId.setColumns(10);

textFieldExpenseId.setEditable(**false**);

textFieldExpenseType = **new** JTextField();

textFieldExpenseType.setBounds(320, 104, 152, 25);

contentPane.add(textFieldExpenseType);

textFieldExpenseType.setColumns(10);

textFieldExpenseDescription = **new** JTextField();

textFieldExpenseDescription.setBounds(320, 133, 152, 25);

contentPane.add(textFieldExpenseDescription);

textFieldExpenseDescription.setColumns(10);

textFieldExpenseAmount = **new** JTextField();

textFieldExpenseAmount.setBounds(320, 162, 152, 25);

contentPane.add(textFieldExpenseAmount);

textFieldExpenseAmount.setColumns(10);

btnAddExpensesButton = **new** JButton("Add");

btnAddExpensesButton.setBounds(236, 209, 89, 23);

contentPane.add(btnAddExpensesButton);

btnAddExpensesButton.addActionListener(action);

btnUpdateExpensesButton = **new** JButton("Update");

btnUpdateExpensesButton.setBounds(236, 209, 101, 23);

contentPane.add(btnUpdateExpensesButton);

btnUpdateExpensesButton.setVisible(**false**);

btnUpdateExpensesButton.addActionListener(action);

btnDeleteExpensesButton = **new** JButton("Delete");

btnDeleteExpensesButton.setBounds(382, 209, 114, 23);

contentPane.add(btnDeleteExpensesButton);

btnDeleteExpensesButton.setVisible(**false**);

btnDeleteExpensesButton.addActionListener(action);

btnCancelButton = **new** JButton("Cancel");

btnCancelButton.setBounds(126, 209, 89, 23);

contentPane.add(btnCancelButton);

btnCancelButton.addActionListener(action);

btnCancelButton.setVisible(**false**);

btnClearButton = **new** JButton("Clear");

btnClearButton.setBounds(370, 209, 103, 23);

contentPane.add(btnClearButton);

btnClearButton.addActionListener(action);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select expense\_id,expense\_desc,expense\_amount from expenses";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Expenses Id");

tableModel.addColumn("Expenses Description");

tableModel.addColumn("Expenses Amount");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,250,640,196);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddExpensesButton.setVisible(**false**);

btnUpdateExpensesButton.setVisible(**true**);

btnDeleteExpensesButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select expense\_id,expense\_amount,expense\_type,expense\_desc from expenses where expense\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldExpenseId.setText(rs.getString(1));

textFieldExpenseAmount.setText(rs.getString(2));

textFieldExpenseType.setText(rs.getString(3));

textFieldExpenseDescription.setText(rs.getString(4));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

}

**void** clearText() {

textFieldExpenseId.setText("");

textFieldExpenseAmount.setText("");

textFieldExpenseType.setText("");

textFieldExpenseDescription.setText("");

}

**private** **class** SwingAction **extends** AbstractAction {

**public** SwingAction() {

putValue(***NAME***, "SwingAction");

putValue(***SHORT\_DESCRIPTION***, "Some short description");

}

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

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

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

txt = "insert into expenses values('"+textFieldExpenseId.getText()+"','"+textFieldExpenseType.getText()+"','"+textFieldExpenseDescription.getText()+"','"+textFieldExpenseAmount.getText()+"')";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Added Expense");

clearText();

}

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

txt = "delete from expenses where expense\_id ='"+textFieldExpenseId.getText()+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Expense Deleted");

clearText();

}

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

txt = "update expenses set expense\_id ='"+Integer.*parseInt*(textFieldExpenseId.getText())+"',expense\_amount='"+Integer.*parseInt*(textFieldExpenseAmount.getText())+"',expense\_type = '"+textFieldExpenseType.getText()+"',expense\_desc = '"+textFieldExpenseDescription.getText()+"' where expense\_id='"+Integer.*parseInt*(textFieldExpenseId.getText())+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Expense Updated");

clearText();

}

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

btnAddExpensesButton.setVisible(**true**);

btnUpdateExpensesButton.setVisible(**false**);

btnDeleteExpensesButton.setVisible(**false**);

btnCancelButton.setVisible(**false**);

clearText();

btnClearButton.setVisible(**true**);

}

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

clearText();

}

txt = "select expense\_id,expense\_desc,expense\_amount from expenses";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Expenses Id");

tableModel.addColumn("Expenses Description");

tableModel.addColumn("Expenses Amount");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,250,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddExpensesButton.setVisible(**false**);

btnUpdateExpensesButton.setVisible(**true**);

btnDeleteExpensesButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select expense\_id,expense\_amount,expense\_type,expense\_desc from expenses where expense\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldExpenseId.setText(rs.getString(1));

textFieldExpenseAmount.setText(rs.getString(2));

textFieldExpenseType.setText(rs.getString(3));

textFieldExpenseDescription.setText(rs.getString(4));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

}

}

}

# FLATS:

**package** ams;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.EventQueue;

**import** java.awt.Font;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenu;

**import** javax.swing.event.MenuListener;

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

**import** javax.swing.event.ListSelectionEvent;

**import** javax.swing.event.ListSelectionListener;

**import** javax.swing.event.MenuEvent;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**import** javax.swing.JLabel;

**import** javax.swing.JTextField;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.SwingConstants;

**import** javax.swing.JButton;

**import** javax.swing.AbstractAction;

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

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** javax.swing.Action;

**public** **class** Flats **extends** Menu{

**private** JPanel contentPane;

**static** Flats *frame*;

/\*\*

\* Launch the application.

\*/

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

*frame* = **new** Flats();

*frame*.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

JLabel lblNewLabel;

JLabel lblNewLabel\_1;

JLabel lblNewLabel\_2;

JLabel lblNewLabel\_3;

**private** JTextField textFieldFlatId;

**private** JTextField textFieldFlatNumber;

**private** JTextField textFieldApartmentId;

**private** JTextField textFieldFlatType;

JButton btnClearButton;

JButton btnCancelButton;

**private** JButton btnAddFlatButton;

**private** JButton btnUpdateFlatButton;

**private** JButton btnDeleteFlatButton;

**private** **final** Action action = **new** SwingAction();

**private** JTable table\_1;

**private** JScrollPane js = **new** JScrollPane(table\_1);

**public** Flats() {

setTitle("Apartment Managment System - Flats");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 695, 523);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

**if**(login.*userName*.isEmpty()) {

dispose();

}

JLabel lblNewLabel = **new** JLabel("Apartment Management System - Flats",SwingConstants.***CENTER***);

lblNewLabel.setForeground(Color.***BLACK***);

lblNewLabel.setBackground(**new** Color(227, 227, 227));

lblNewLabel.setOpaque(**true**);

lblNewLabel.setFont(**new** Font("Stencil", Font.***PLAIN***, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

lblNewLabel = **new** JLabel("Flat Id");

lblNewLabel.setBounds(163, 71, 120, 25);

contentPane.add(lblNewLabel);

lblNewLabel\_1 = **new** JLabel("Flat Number");

lblNewLabel\_1.setBounds(163, 100, 120, 25);

contentPane.add(lblNewLabel\_1);

lblNewLabel\_2 = **new** JLabel("Apartment Id");

lblNewLabel\_2.setBounds(163, 129, 120, 25);

contentPane.add(lblNewLabel\_2);

lblNewLabel\_3 = **new** JLabel("Flat Type");

lblNewLabel\_3.setBounds(163, 158, 120, 25);

contentPane.add(lblNewLabel\_3);

textFieldFlatId = **new** JTextField();

textFieldFlatId.setBounds(320, 71, 120, 25);

contentPane.add(textFieldFlatId);

textFieldFlatId.setColumns(10);

textFieldFlatId.setEditable(**false**);

textFieldFlatNumber = **new** JTextField();

textFieldFlatNumber.setBounds(320, 100, 120, 25);

contentPane.add(textFieldFlatNumber);

textFieldFlatNumber.setColumns(10);

textFieldApartmentId = **new** JTextField();

textFieldApartmentId.setBounds(320, 129, 120, 25);

contentPane.add(textFieldApartmentId);

textFieldApartmentId.setColumns(10);

textFieldFlatType = **new** JTextField();

textFieldFlatType.setBounds(320, 158, 120, 25);

contentPane.add(textFieldFlatType);

textFieldFlatType.setColumns(10);

btnAddFlatButton = **new** JButton("Add");

btnAddFlatButton.setBounds(237, 235, 89, 23);

contentPane.add(btnAddFlatButton);

btnAddFlatButton.setVisible(**true**);

btnAddFlatButton.addActionListener(action);

btnUpdateFlatButton = **new** JButton("Update");

btnUpdateFlatButton.setBounds(242, 235, 102, 23);

contentPane.add(btnUpdateFlatButton);

btnUpdateFlatButton.setVisible(**false**);

btnUpdateFlatButton.addActionListener(action);

btnDeleteFlatButton = **new** JButton("Delete");

btnDeleteFlatButton.setBounds(368, 235, 115, 23);

contentPane.add(btnDeleteFlatButton);

btnDeleteFlatButton.setVisible(**false**);

btnDeleteFlatButton.addActionListener(action);

btnCancelButton = **new** JButton("Cancel");

btnCancelButton.setBounds(121, 235, 89, 23);

contentPane.add(btnCancelButton);

btnCancelButton.addActionListener(action);

btnCancelButton.setVisible(**false**);

btnClearButton = **new** JButton("Clear");

btnClearButton.setBounds(368, 235, 103, 23);

contentPane.add(btnClearButton);

btnClearButton.addActionListener(action);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select flat\_id,flat\_no,flat\_type from flats";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Flat Id");

tableModel.addColumn("Flat Number");

tableModel.addColumn("Flat Type");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,290,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddFlatButton.setVisible(**false**);

btnUpdateFlatButton.setVisible(**true**);

btnDeleteFlatButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select flat\_id,apartment\_id,flat\_no,flat\_type from flats where flat\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldFlatId.setText(rs.getString(1));

textFieldApartmentId.setText(rs.getString(2));

textFieldFlatNumber.setText(rs.getString(3));

textFieldFlatType.setText(rs.getString(4));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

}

**void** clearText() {

textFieldFlatId.setText("");

textFieldFlatNumber.setText("");

textFieldApartmentId.setText("");

textFieldFlatType.setText("");

}

**private** **class** SwingAction **extends** AbstractAction {

**public** SwingAction() {

putValue(***NAME***, "SwingAction");

putValue(***SHORT\_DESCRIPTION***, "Some short description");

}

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

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

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

txt = "insert into flats values('"+textFieldFlatId.getText()+"','"+textFieldApartmentId.getText()+"','"+textFieldFlatNumber.getText()+"','"+textFieldFlatType.getText()+"')";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Added Flat");

clearText();

}

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

txt = "delete from flats where flat\_id ='"+textFieldFlatId.getText()+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Flat Deleted");

clearText();

}

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

txt = "update flats set flat\_id ='"+Integer.*parseInt*(textFieldFlatId.getText())+"',apartment\_id='"+Integer.*parseInt*(textFieldApartmentId.getText())+"',flat\_no = '"+Integer.*parseInt*(textFieldFlatNumber.getText())+"',flat\_type = '"+textFieldFlatType.getText()+"' where flat\_id='"+Integer.*parseInt*(textFieldFlatId.getText())+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Flat Updated");

clearText();

}

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

btnAddFlatButton.setVisible(**true**);

btnUpdateFlatButton.setVisible(**false**);

btnDeleteFlatButton.setVisible(**false**);

btnCancelButton.setVisible(**false**);

clearText();

btnClearButton.setVisible(**true**);

}

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

clearText();

}

txt = "select flat\_id,flat\_no,flat\_type from flats";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Flat Id");

tableModel.addColumn("Flat Number");

tableModel.addColumn("Flat Type");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,290,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddFlatButton.setVisible(**false**);

btnUpdateFlatButton.setVisible(**true**);

btnDeleteFlatButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select flat\_id,apartment\_id,flat\_no,flat\_type from flats where flat\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldFlatId.setText(rs.getString(1));

textFieldApartmentId.setText(rs.getString(2));

textFieldFlatNumber.setText(rs.getString(3));

textFieldFlatType.setText(rs.getString(4));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

JOptionPane.*showMessageDialog*(**null**,"Invalid Input. Please try again");

}

}

}

}

# RESIDENTS:

**package** ams;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.EventQueue;

**import** java.awt.Font;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.event.ListSelectionEvent;

**import** javax.swing.event.ListSelectionListener;

**import** javax.swing.event.MenuEvent;

**import** javax.swing.event.MenuListener;

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

**import** javax.swing.JLabel;

**import** javax.swing.JTextField;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.SwingConstants;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenu;

**import** javax.swing.AbstractAction;

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

**import** java.awt.event.KeyEvent;

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** java.text.SimpleDateFormat;

**import** javax.swing.Action;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**public** **class** Residents **extends** Menu {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

Residents frame = **new** Residents();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

JLabel lblNewLabel;

JLabel lblNewLabel\_1;

JLabel lblNewLabel\_2;

JLabel lblNewLabel\_3;

JLabel lblNewLabel\_4;

JButton btnClearButton;

JButton btnCancelButton;

**private** JTextField textFieldResidentId;

**private** JTextField textFieldResidentName;

**private** JTextField textFieldResidentType;

**private** JTextField textFieldPhoneNumber;

**private** JButton btnAddResidentsButton;

**private** JButton btnUpdateResidentsButton;

**private** JButton btnDeleteResidentsButton;

JComboBox comboBoxFlatNo;

**private** **final** Action action = **new** SwingAction();

**private** JTable table\_1;

**private** JScrollPane js = **new** JScrollPane(table\_1);

**public** Residents() {

setTitle("Apartment Management System - Residents");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 695, 523);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

**if**(login.*userName*.isEmpty()) {

dispose();

}

JLabel lblNewLabel = **new** JLabel("Apartment Management System - Residents",SwingConstants.***CENTER***);

lblNewLabel.setForeground(Color.***BLACK***);

lblNewLabel.setBackground(**new** Color(227, 227, 227));

lblNewLabel.setOpaque(**true**);

lblNewLabel.setFont(**new** Font("Stencil", Font.***PLAIN***, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

lblNewLabel = **new** JLabel("Resident Id");

lblNewLabel.setBounds(163, 79, 120, 25);

contentPane.add(lblNewLabel);

lblNewLabel\_1 = **new** JLabel("Flat Number");

lblNewLabel\_1.setBounds(163, 195, 120, 25);

contentPane.add(lblNewLabel\_1);

lblNewLabel\_2 = **new** JLabel("Resident Name");

lblNewLabel\_2.setBounds(163, 108, 120, 25);

contentPane.add(lblNewLabel\_2);

lblNewLabel\_3 = **new** JLabel("Resident Type");

lblNewLabel\_3.setBounds(163, 166, 120, 25);

contentPane.add(lblNewLabel\_3);

lblNewLabel\_4 = **new** JLabel("Phone Number");

lblNewLabel\_4.setBounds(163, 137, 120, 25);

contentPane.add(lblNewLabel\_4);

textFieldResidentId = **new** JTextField();

textFieldResidentId.setBounds(320, 79, 120, 25);

contentPane.add(textFieldResidentId);

textFieldResidentId.setColumns(10);

textFieldResidentId.setEditable(**false**);

comboBoxFlatNo = **new** JComboBox();

comboBoxFlatNo.setBounds(320, 195, 120, 25);

contentPane.add(comboBoxFlatNo);

textFieldResidentName = **new** JTextField();

textFieldResidentName.setBounds(320, 108, 120, 25);

contentPane.add(textFieldResidentName);

textFieldResidentName.setColumns(10);

textFieldResidentType = **new** JTextField();

textFieldResidentType.setBounds(320, 166, 120, 25);

contentPane.add(textFieldResidentType);

textFieldResidentType.setColumns(10);

textFieldPhoneNumber = **new** JTextField();

textFieldPhoneNumber.setBounds(320, 137, 120, 25);

contentPane.add(textFieldPhoneNumber);

textFieldPhoneNumber.setColumns(10);

textFieldPhoneNumber.addKeyListener(**new** java.awt.event.KeyAdapter() {

**public** **void** keyTyped(java.awt.event.KeyEvent evt) {

**if**(textFieldPhoneNumber.getText().length()>=10&&!(evt.getKeyChar()==KeyEvent.***VK\_DELETE***||evt.getKeyChar()==KeyEvent.***VK\_BACK\_SPACE***)||!(evt.getKeyChar()>= '0' &&evt.getKeyChar()<= '9')) {

//getToolkit().beep();

evt.consume();

}

}

});

btnAddResidentsButton = **new** JButton("Add");

btnAddResidentsButton.setBounds(224, 260, 89, 23);

contentPane.add(btnAddResidentsButton);

btnAddResidentsButton.addActionListener(action);

btnUpdateResidentsButton = **new** JButton("Update");

btnUpdateResidentsButton.setBounds(224, 260, 101, 23);

contentPane.add(btnUpdateResidentsButton);

btnUpdateResidentsButton.setVisible(**false**);

btnUpdateResidentsButton.addActionListener(action);

btnDeleteResidentsButton = **new** JButton("Delete");

btnDeleteResidentsButton.setBounds(350, 260, 114, 23);

contentPane.add(btnDeleteResidentsButton);

btnDeleteResidentsButton.setVisible(**false**);

btnDeleteResidentsButton.addActionListener(action);

btnCancelButton = **new** JButton("Cancel");

btnCancelButton.setBounds(124, 260, 89, 23);

contentPane.add(btnCancelButton);

btnCancelButton.addActionListener(action);

btnCancelButton.setVisible(**false**);

btnClearButton = **new** JButton("Clear");

btnClearButton.setBounds(350, 260, 103, 23);

contentPane.add(btnClearButton);

btnClearButton.addActionListener(action);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select flat\_no from flats";

ResultSet rs1 = stmt.executeQuery(txt);

**while**(rs1.next()) {

comboBoxFlatNo.addItem(rs1.getInt(1));

}

txt = "select resident\_id,resident\_name,flat\_no,phone\_number from residents";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Resident Id");

tableModel.addColumn("Resident Name");

tableModel.addColumn("Flat Id");

tableModel.addColumn("Resident Phone Number");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,294,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddResidentsButton.setVisible(**false**);

btnUpdateResidentsButton.setVisible(**true**);

btnDeleteResidentsButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select resident\_id,flat\_no,resident\_name,resident\_type,phone\_number from residents where resident\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldResidentId.setText(rs.getString(1));

comboBoxFlatNo.setSelectedItem(rs.getInt(2));

textFieldResidentName.setText(rs.getString(3));

textFieldResidentType.setText(rs.getString(4));

textFieldPhoneNumber.setText(rs.getString(5));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

}

**void** clearText() {

textFieldResidentId.setText("");

comboBoxFlatNo.setToolTipText("");

textFieldResidentName.setText("");

textFieldResidentType.setText("");

textFieldPhoneNumber.setText("");

}

**private** **class** SwingAction **extends** AbstractAction {

**public** SwingAction() {

putValue(***NAME***, "SwingAction");

putValue(***SHORT\_DESCRIPTION***, "Some short description");

}

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

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

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

txt = "insert into residents values('"+textFieldResidentId.getText()+"','"+comboBoxFlatNo.getSelectedItem()+"','"+textFieldResidentName.getText()+"','"+textFieldResidentType.getText()+"','"+textFieldPhoneNumber.getText()+"')";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Added Resident");

clearText();

}

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

txt = "delete from residents where payment\_id ='"+textFieldResidentId.getText()+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Resident Deleted");

clearText();

}

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

txt = "update residents set resident\_id ='"+Integer.*parseInt*(textFieldResidentId.getText())+"',flat\_no='"+comboBoxFlatNo.getSelectedItem()+"',resident\_name = '"+textFieldResidentName.getText()+"',resident\_type = '"+textFieldResidentType.getText()+"',phone\_number = '"+textFieldPhoneNumber.getText()+"' where resident\_id='"+Integer.*parseInt*(textFieldResidentId.getText())+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Resident Updated");

clearText();

}

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

btnAddResidentsButton.setVisible(**true**);

btnUpdateResidentsButton.setVisible(**false**);

btnDeleteResidentsButton.setVisible(**false**);

btnCancelButton.setVisible(**false**);

clearText();

btnClearButton.setVisible(**true**);

}

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

clearText();

}

txt = "select resident\_id,resident\_name,flat\_no,phone\_number from residents";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Resident Id");

tableModel.addColumn("Resident Name");

tableModel.addColumn("Flat Id");

tableModel.addColumn("Resident Phone Number");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,294,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddResidentsButton.setVisible(**false**);

btnUpdateResidentsButton.setVisible(**true**);

btnDeleteResidentsButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select resident\_id,flat\_no,resident\_name,resident\_type,phone\_number from residents where resident\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldResidentId.setText(rs.getString(1));

comboBoxFlatNo.setSelectedItem(rs.getInt(2));

textFieldResidentName.setText(rs.getString(3));

textFieldResidentType.setText(rs.getString(4));

textFieldPhoneNumber.setText(rs.getString(5));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

}

}

}

# CHARGES:

**package** ams;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.EventQueue;

**import** java.awt.Font;

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

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** javax.swing.AbstractAction;

**import** javax.swing.Action;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.SwingConstants;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.event.ListSelectionEvent;

**import** javax.swing.event.ListSelectionListener;

**import** javax.swing.event.MenuEvent;

**import** javax.swing.event.MenuListener;

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

**import** ams.Charges.SwingAction;

**import** javax.swing.JComboBox;

**public** **class** Charges **extends** Menu {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

Charges frame = **new** Charges();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

JLabel lblNewLabel;

**private** JLabel lblNewLabel\_1;

**private** JLabel lblNewLabel\_2;

**private** JLabel lblNewLabel\_3;

**private** JTextField textFieldChargesId;

**private** JTextField textFieldChargesType;

**private** JTextField textFieldChargesDescription;

**private** JTextField textFieldChargesAmount;

**private** **final** Action action = **new** SwingAction();

**private** JButton btnAddChargesButton;

**private** JButton btnUpdateChargesButton;

**private** JButton btnDeleteChargesButton;

**private** JTable table\_1;

**private** JScrollPane js = **new** JScrollPane(table\_1);

**private** JLabel lblNewLabel\_4;

JButton btnViewChargesButton;

**private** JButton btnPayChargeButton;

**private** JButton btnCancelButton;

**private** JButton btnClearButton;

JComboBox comboBoxFlatNo;

**public** Charges() {

setTitle("Apartment Management System - Charges");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 695, 523);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Apartment Management System - Charges",SwingConstants.***CENTER***);

lblNewLabel.setForeground(Color.***BLACK***);

lblNewLabel.setBackground(**new** Color(227, 227, 227));

lblNewLabel.setOpaque(**true**);

lblNewLabel.setFont(**new** Font("Stencil", Font.***PLAIN***, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

lblNewLabel = **new** JLabel("Charge Id");

lblNewLabel.setBounds(163, 63, 120, 25);

contentPane.add(lblNewLabel);

lblNewLabel\_1 = **new** JLabel("Charge Type");

lblNewLabel\_1.setBounds(163, 92, 120, 25);

contentPane.add(lblNewLabel\_1);

lblNewLabel\_2 = **new** JLabel("Charge description");

lblNewLabel\_2.setBounds(163, 121, 120, 25);

contentPane.add(lblNewLabel\_2);

lblNewLabel\_3 = **new** JLabel("Charge Amount");

lblNewLabel\_3.setBounds(163, 150, 120, 25);

contentPane.add(lblNewLabel\_3);

textFieldChargesId = **new** JTextField();

textFieldChargesId.setBounds(320, 63, 152, 25);

contentPane.add(textFieldChargesId);

textFieldChargesId.setColumns(10);

textFieldChargesId.setEditable(**false**);

textFieldChargesType = **new** JTextField();

textFieldChargesType.setBounds(320, 92, 152, 25);

contentPane.add(textFieldChargesType);

textFieldChargesType.setColumns(10);

textFieldChargesDescription = **new** JTextField();

textFieldChargesDescription.setBounds(320, 121, 152, 25);

contentPane.add(textFieldChargesDescription);

textFieldChargesDescription.setColumns(10);

textFieldChargesAmount = **new** JTextField();

textFieldChargesAmount.setBounds(320, 150, 152, 25);

contentPane.add(textFieldChargesAmount);

textFieldChargesAmount.setColumns(10);

lblNewLabel\_4 = **new** JLabel("Flat Number");

lblNewLabel\_4.setBounds(163, 179, 120, 25);

contentPane.add(lblNewLabel\_4);

lblNewLabel\_4.setVisible(**false**);

comboBoxFlatNo = **new** JComboBox();

comboBoxFlatNo.setBounds(320, 179, 152, 25);

contentPane.add(comboBoxFlatNo);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt = "select flat\_no from flats";

ResultSet rs1 = stmt.executeQuery(txt);

**while**(rs1.next()) {

comboBoxFlatNo.addItem(rs1.getInt(1));

}

comboBoxFlatNo.setSelectedItem(**null**);

}

**catch**(Exception e) {

e.printStackTrace();

}

**if**(login.*userType*.equals("user")) {

btnViewChargesButton = **new** JButton("View Charges");

btnViewChargesButton.setBounds(10, 64, 133, 23);

contentPane.add(btnViewChargesButton);

btnViewChargesButton.addActionListener(action);

lblNewLabel.setVisible(**false**);

lblNewLabel\_1.setVisible(**false**);

lblNewLabel\_2.setVisible(**false**);

lblNewLabel\_3.setVisible(**false**);

textFieldChargesId.setVisible(**false**);

textFieldChargesType.setVisible(**false**);

textFieldChargesDescription.setVisible(**false**);

textFieldChargesAmount.setVisible(**false**);

comboBoxFlatNo.setVisible(**false**);

textFieldChargesType.setEditable(**false**);

textFieldChargesDescription.setEditable(**false**);

textFieldChargesAmount.setEditable(**false**);

comboBoxFlatNo.setEditable(**false**);

btnPayChargeButton = **new** JButton("Pay");

btnPayChargeButton.setBounds(228, 244, 89, 23);

contentPane.add(btnPayChargeButton);

btnPayChargeButton.addActionListener(action);

btnPayChargeButton.setVisible(**false**);

}

**if**(login.*userType*.equals("admin")) {

textFieldChargesId.setVisible(**true**);

textFieldChargesType.setVisible(**true**);

textFieldChargesDescription.setVisible(**true**);

textFieldChargesAmount.setVisible(**true**);

comboBoxFlatNo.setVisible(**true**);

lblNewLabel.setVisible(**true**);

lblNewLabel\_1.setVisible(**true**);

lblNewLabel\_2.setVisible(**true**);

lblNewLabel\_3.setVisible(**true**);

lblNewLabel\_4.setVisible(**true**);

btnAddChargesButton = **new** JButton("Add");

btnAddChargesButton.setBounds(224, 221, 89, 23);

contentPane.add(btnAddChargesButton);

btnAddChargesButton.addActionListener(action);

btnCancelButton = **new** JButton("Cancel");

btnCancelButton.setBounds(81, 221, 89, 23);

contentPane.add(btnCancelButton);

btnCancelButton.addActionListener(action);

btnCancelButton.setVisible(**false**);

btnUpdateChargesButton = **new** JButton("Update");

btnUpdateChargesButton.setBounds(224, 221, 89, 23);

contentPane.add(btnUpdateChargesButton);

btnUpdateChargesButton.addActionListener(action);

btnUpdateChargesButton.setVisible(**false**);

btnDeleteChargesButton = **new** JButton("Delete");

btnDeleteChargesButton.setBounds(359, 221, 101, 23);

contentPane.add(btnDeleteChargesButton);

btnDeleteChargesButton.addActionListener(action);

btnDeleteChargesButton.setVisible(**false**);

btnClearButton = **new** JButton("Clear");

btnClearButton.setBounds(359, 221, 122, 23);

contentPane.add(btnClearButton);

btnClearButton.addActionListener(action);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt;

txt = "select charges\_id,charges\_type,charges\_amount,flat\_no from charges where status='unpaid'";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("charges Id");

tableModel.addColumn("charges Type");

tableModel.addColumn("charges Amount");

tableModel.addColumn("Flat Number");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,250,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

**if**(login.*userType*.equals("admin")) {

btnAddChargesButton.setVisible(**false**);

btnUpdateChargesButton.setVisible(**true**);

btnDeleteChargesButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

}

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select charges\_id,charges\_amount,charges\_type,charges\_desc,flat\_no from charges where charges\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldChargesId.setText(rs.getString(1));

textFieldChargesAmount.setText(rs.getString(2));

textFieldChargesType.setText(rs.getString(3));

textFieldChargesDescription.setText(rs.getString(4));

comboBoxFlatNo.setSelectedItem(rs.getInt(5));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception e){

}

}

}

**void** clearText() {

textFieldChargesId.setText("");

textFieldChargesAmount.setText("");

textFieldChargesType.setText("");

textFieldChargesDescription.setText("");

comboBoxFlatNo.setToolTipText("");

}

**public** **class** SwingAction **extends** AbstractAction {

**public** SwingAction() {

putValue(***NAME***, "SwingAction");

putValue(***SHORT\_DESCRIPTION***, "Some short description");

}

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

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

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

btnViewChargesButton.setVisible(**false**);

textFieldChargesId.setVisible(**true**);

textFieldChargesType.setVisible(**true**);

textFieldChargesDescription.setVisible(**true**);

textFieldChargesAmount.setVisible(**true**);

comboBoxFlatNo.setVisible(**true**);

lblNewLabel.setVisible(**true**);

lblNewLabel\_1.setVisible(**true**);

lblNewLabel\_2.setVisible(**true**);

lblNewLabel\_3.setVisible(**true**);

lblNewLabel\_4.setVisible(**true**);

btnPayChargeButton.setVisible(**true**);

}

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

**try** {

txt = "insert into payments (payment\_amount,payment\_date,payment\_type,flat\_no,charge\_id) values("+Integer.*parseInt*(textFieldChargesAmount.getText())+",sysdate,'card','"+comboBoxFlatNo.getSelectedItem()+"','"+textFieldChargesId.getText()+"')";

stmt.executeQuery(txt);

//txt = "delete from charges where charges\_id ='"+textFieldChargesId.getText()+"'";

txt = "update charges set status='paid' where charges\_id = "+textFieldChargesId.getText()+"";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Charge Paid");

}

**catch**(Exception ex) {

JOptionPane.*showMessageDialog*(**null**,"Invalid");

}

}

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

txt = "insert into charges(charges\_id,flat\_no,charges\_type,charges\_desc,charges\_amount) values('"+textFieldChargesId.getText()+"','"+comboBoxFlatNo.getSelectedItem()+"','"+textFieldChargesType.getText()+"','"+textFieldChargesDescription.getText()+"','"+textFieldChargesAmount.getText()+"')";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Added Charges");

clearText();

}

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

txt = "delete from charges where charges\_id ='"+textFieldChargesId.getText()+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Charges Deleted");

clearText();

}

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

txt = "update charges set charges\_id ='"+Integer.*parseInt*(textFieldChargesId.getText())+"',flat\_no = '"+comboBoxFlatNo.getSelectedItem()+"',charges\_amount='"+Integer.*parseInt*(textFieldChargesAmount.getText())+"',charges\_type = '"+textFieldChargesType.getText()+"',charges\_desc = '"+textFieldChargesDescription.getText()+"' where charges\_id='"+Integer.*parseInt*(textFieldChargesId.getText())+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Charges Updated");

clearText();

}

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

btnAddChargesButton.setVisible(**true**);

btnUpdateChargesButton.setVisible(**false**);

btnDeleteChargesButton.setVisible(**false**);

btnCancelButton.setVisible(**false**);

clearText();

btnClearButton.setVisible(**true**);

}

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

clearText();

}

**if**(login.*userType*.equals("user")){

txt = "select charges\_id,charges\_type,charges\_amount,flat\_no from charges where flat\_no = "+login.*flatNo*+" and status='unpaid'";

}

**else** {

txt = "select charges\_id,charges\_type,charges\_amount,flat\_no from charges where status='unpaid'";

}

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("charges Id");

tableModel.addColumn("charges Type");

tableModel.addColumn("charges Amount");

tableModel.addColumn("Flat Number");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,250,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

**if**(login.*userType*.equals("admin")) {

btnAddChargesButton.setVisible(**false**);

btnUpdateChargesButton.setVisible(**true**);

btnDeleteChargesButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

}

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select charges\_id,charges\_amount,charges\_type,charges\_desc,flat\_no from charges where charges\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldChargesId.setText(rs.getString(1));

textFieldChargesAmount.setText(rs.getString(2));

textFieldChargesType.setText(rs.getString(3));

textFieldChargesDescription.setText(rs.getString(4));

comboBoxFlatNo.setSelectedItem(rs.getInt(5));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

}

}

}

# COMMITTEE:

**package** ams;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.EventQueue;

**import** java.awt.Font;

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

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

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** java.util.ArrayList;

**import** javax.swing.AbstractAction;

**import** javax.swing.Action;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.SwingConstants;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.event.ListSelectionEvent;

**import** javax.swing.event.ListSelectionListener;

**import** javax.swing.event.MenuEvent;

**import** javax.swing.event.MenuListener;

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

**import** ams.Committee.SwingAction;

**public** **class** Committee **extends** Menu {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

Committee frame = **new** Committee();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

JLabel lblNewLabel;

JLabel lblNewLabel\_1;

JLabel lblNewLabel\_2;

JLabel lblNewLabel\_3;

JButton btnClearButton;

JButton btnCancelButton;

JComboBox comboBoxFlatNo;

JComboBox comboBoxMemberType;

**private** JTextField textFieldMemberName;

**private** JTextField textFieldMemberId;

**private** JButton btnAddMemberButton;

**private** JButton btnUpdateMemberButton;

**private** JButton btnDeleteMemberButton;

**private** **final** Action action = **new** SwingAction();

**private** JTable table\_1;

**private** JScrollPane js = **new** JScrollPane(table\_1);

ArrayList<Integer> array\_flats = **new** ArrayList<Integer>();

**public** Committee() {

setTitle("Apartment Managment System - Committee");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 695, 523);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

**if**(login.*userName*.isEmpty()) {

dispose();

}

JLabel lblNewLabel = **new** JLabel("Apartment Management System - Committee",SwingConstants.***CENTER***);

lblNewLabel.setForeground(Color.***BLACK***);

lblNewLabel.setBackground(**new** Color(227, 227, 227));

lblNewLabel.setOpaque(**true**);

lblNewLabel.setFont(**new** Font("Stencil", Font.***PLAIN***, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

lblNewLabel = **new** JLabel("Member Id");

lblNewLabel.setBounds(163, 87, 120, 25);

contentPane.add(lblNewLabel);

lblNewLabel\_1 = **new** JLabel("Member Name");

lblNewLabel\_1.setBounds(163, 145, 120, 25);

contentPane.add(lblNewLabel\_1);

lblNewLabel\_2 = **new** JLabel("Flat Id");

lblNewLabel\_2.setBounds(163, 116, 120, 25);

contentPane.add(lblNewLabel\_2);

lblNewLabel\_3 = **new** JLabel("Member Type");

lblNewLabel\_3.setBounds(163, 174, 120, 25);

contentPane.add(lblNewLabel\_3);

comboBoxFlatNo = **new** JComboBox();

comboBoxFlatNo.setBounds(320, 116, 120, 25);

contentPane.add(comboBoxFlatNo);

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

@Override

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

// **TODO** Auto-generated method stub

**if**(comboBoxFlatNo.getSelectedIndex()>0) {

**int** s = (Integer) comboBoxFlatNo.getSelectedItem();

**if**(array\_flats.contains(s)) {

**try** {

//System.out.println(s);

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select resident\_name from residents where flat\_no='"+s+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldMemberName.setText(rs.getString(1));

}

}

**catch**(Exception ex) {

//System.out.println("not there");

}

}

**else** {

textFieldMemberName.setText("");

}

}

}

});

textFieldMemberName = **new** JTextField();

textFieldMemberName.setBounds(320, 145, 120, 25);

contentPane.add(textFieldMemberName);

textFieldMemberName.setColumns(10);

comboBoxMemberType = **new** JComboBox();

comboBoxMemberType.setBounds(320, 174, 120, 25);

contentPane.add(comboBoxMemberType);

textFieldMemberId = **new** JTextField();

textFieldMemberId.setBounds(320, 87, 120, 25);

contentPane.add(textFieldMemberId);

textFieldMemberId.setColumns(10);

textFieldMemberId.setEditable(**false**);

btnAddMemberButton = **new** JButton("Add");

btnAddMemberButton.setBounds(237, 239, 89, 23);

contentPane.add(btnAddMemberButton);

btnAddMemberButton.addActionListener(action);

btnUpdateMemberButton = **new** JButton("Update");

btnUpdateMemberButton.setBounds(237, 239, 102, 23);

contentPane.add(btnUpdateMemberButton);

btnUpdateMemberButton.setVisible(**false**);

btnUpdateMemberButton.addActionListener(action);

btnDeleteMemberButton = **new** JButton("Delete");

btnDeleteMemberButton.setBounds(349, 239, 115, 23);

contentPane.add(btnDeleteMemberButton);

btnDeleteMemberButton.setVisible(**false**);

btnDeleteMemberButton.addActionListener(action);

btnCancelButton = **new** JButton("Cancel");

btnCancelButton.setBounds(125, 239, 89, 23);

contentPane.add(btnCancelButton);

btnCancelButton.addActionListener(action);

btnCancelButton.setVisible(**false**);

btnClearButton = **new** JButton("Clear");

btnClearButton.setBounds(351, 239, 103, 23);

contentPane.add(btnClearButton);

btnClearButton.addActionListener(action);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select flat\_no from flats";

ResultSet rs1 = stmt.executeQuery(txt);

**while**(rs1.next()) {

comboBoxFlatNo.addItem(rs1.getInt(1));

}

rs1.close();

txt = "select member\_type from members";

ResultSet rs2 = stmt.executeQuery(txt);

**while**(rs2.next()) {

comboBoxMemberType.addItem(rs2.getString(1));

}

rs2.close();

txt = "select flat\_no from residents";

ResultSet rs3 = stmt.executeQuery(txt);

**while**(rs3.next()) {

array\_flats.add(rs3.getInt(1));

}

rs3.close();

txt = "select member\_id,member\_name,member\_type from committee";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Member Id");

tableModel.addColumn("Member Name");

tableModel.addColumn("Member Type");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,290,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddMemberButton.setVisible(**false**);

btnUpdateMemberButton.setVisible(**true**);

btnDeleteMemberButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select member\_id,member\_name,flat\_no,member\_type from committee where member\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldMemberId.setText(rs.getString(1));

textFieldMemberName.setText(rs.getString(2));

comboBoxFlatNo.setSelectedItem(rs.getInt(3));

comboBoxMemberType.setSelectedItem(rs.getString(4));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

}

**void** clearText() {

textFieldMemberId.setText("");

comboBoxFlatNo.setSelectedItem(**null**);

textFieldMemberName.setText("");

comboBoxMemberType.setSelectedIndex(-1);

}

**public** **class** SwingAction **extends** AbstractAction {

**public** SwingAction() {

putValue(***NAME***, "SwingAction");

putValue(***SHORT\_DESCRIPTION***, "Some short description");

}

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

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

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

txt = "insert into committee values('"+textFieldMemberId.getText()+"','"+textFieldMemberName.getText()+"','"+comboBoxFlatNo.getSelectedItem()+"','"+comboBoxMemberType.getSelectedItem()+"')";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Added Member");

clearText();

}

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

txt = "delete from committee where member\_id ='"+textFieldMemberId.getText()+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Member Deleted");

clearText();

}

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

txt = "update committee set member\_id ='"+Integer.*parseInt*(textFieldMemberId.getText())+"',flat\_no='"+comboBoxFlatNo.getSelectedItem()+"',member\_name = '"+textFieldMemberName.getText()+"',member\_type = '"+comboBoxMemberType.getSelectedItem()+"' where member\_id='"+Integer.*parseInt*(textFieldMemberId.getText())+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Member Updated");

clearText();

}

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

btnAddMemberButton.setVisible(**true**);

btnUpdateMemberButton.setVisible(**false**);

btnDeleteMemberButton.setVisible(**false**);

btnCancelButton.setVisible(**false**);

clearText();

btnClearButton.setVisible(**true**);

}

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

clearText();

}

txt = "select member\_id,member\_name,member\_type from committee";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Member Id");

tableModel.addColumn("Member Name");

tableModel.addColumn("Member Type");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,290,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddMemberButton.setVisible(**false**);

btnUpdateMemberButton.setVisible(**true**);

btnDeleteMemberButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select member\_id,member\_name,flat\_no,member\_type from committee where member\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldMemberId.setText(rs.getString(1));

textFieldMemberName.setText(rs.getString(2));

comboBoxFlatNo.setSelectedItem(rs.getInt(3));

comboBoxMemberType.setSelectedItem(rs.getString(4));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

ex.printStackTrace();

}

}

}

}

# PAYMENTS:

**package** ams;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.EventQueue;

**import** java.awt.Font;

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

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** java.text.SimpleDateFormat;

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

**import** javax.swing.AbstractAction;

**import** javax.swing.Action;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.SwingConstants;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.event.ListSelectionEvent;

**import** javax.swing.event.ListSelectionListener;

**import** javax.swing.event.MenuEvent;

**import** javax.swing.event.MenuListener;

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

**import** com.github.lgooddatepicker.components.DateTimePicker;

**import** ams.Payments.SwingAction;

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

**public** **class** Payments **extends** Menu {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

Payments frame = **new** Payments();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

**void** clearText() {

textFieldPaymentsId.setText("");

textFieldPaymentsAmount.setText("");

textFieldPaymentsType.setText("");

comboBoxFlatNo.setSelectedItem(**null**);

paymentDate.setDateTimeStrict(**null**);

textFieldPaymentsAmount.setText("");

}

/\*\*

\* Create the frame.

\*/

JLabel lblNewLabel;

**private** JLabel lblNewLabel\_1;

**private** JLabel lblNewLabel\_2;

**private** JLabel lblNewLabel\_3;

**private** JTextField textFieldPaymentsId;

**private** JTextField textFieldPaymentsType;

**private** JTextField textFieldPaymentsAmount;

**private** **final** Action action = **new** SwingAction();

**private** JButton btnAddPaymentsButton;

**private** JButton btnUpdatePaymentsButton;

**private** JButton btnDeletePaymentsButton;

**private** JTable table\_1;

**private** JScrollPane js = **new** JScrollPane(table\_1);

**private** JLabel lblNewLabel\_4;

JComboBox comboBoxFlatNo;

DateTimePicker paymentDate;

**private** JButton btnClearButton;

**private** JButton btnCancelButton;

**public** Payments() {

setTitle("Apartment Management System - Payments");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 695, 523);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

**if**(login.*userName*.isEmpty()) {

dispose();

}

JLabel lblNewLabel = **new** JLabel("Apartment Management System - Payments",SwingConstants.***CENTER***);

lblNewLabel.setForeground(Color.***BLACK***);

lblNewLabel.setBackground(**new** Color(227, 227, 227));

lblNewLabel.setOpaque(**true**);

lblNewLabel.setFont(**new** Font("Stencil", Font.***PLAIN***, 20));

lblNewLabel.setBounds(0, 0, 679, 60);

getContentPane().add(lblNewLabel);

lblNewLabel = **new** JLabel("Payment Id");

lblNewLabel.setBounds(163, 81, 120, 25);

contentPane.add(lblNewLabel);

lblNewLabel\_1 = **new** JLabel("Payment Type");

lblNewLabel\_1.setBounds(163, 110, 120, 25);

contentPane.add(lblNewLabel\_1);

lblNewLabel\_2 = **new** JLabel("Payment Date");

lblNewLabel\_2.setBounds(163, 139, 120, 25);

contentPane.add(lblNewLabel\_2);

lblNewLabel\_3 = **new** JLabel("Payment Amount");

lblNewLabel\_3.setBounds(163, 168, 120, 25);

contentPane.add(lblNewLabel\_3);

textFieldPaymentsId = **new** JTextField();

textFieldPaymentsId.setBounds(320, 81, 120, 25);

contentPane.add(textFieldPaymentsId);

textFieldPaymentsId.setColumns(10);

textFieldPaymentsId.setEditable(**false**);

textFieldPaymentsType = **new** JTextField();

textFieldPaymentsType.setBounds(320, 110, 120, 25);

contentPane.add(textFieldPaymentsType);

textFieldPaymentsType.setColumns(10);

paymentDate = **new** DateTimePicker();

paymentDate.setBounds(320, 139, 250, 25);

getContentPane().add(paymentDate);

textFieldPaymentsAmount = **new** JTextField();

textFieldPaymentsAmount.setBounds(320, 168, 120, 25);

contentPane.add(textFieldPaymentsAmount);

textFieldPaymentsAmount.setColumns(10);

lblNewLabel\_4 = **new** JLabel("Flat Number");

lblNewLabel\_4.setBounds(163, 197, 120, 25);

contentPane.add(lblNewLabel\_4);

comboBoxFlatNo = **new** JComboBox();

comboBoxFlatNo.setBounds(320, 197, 120, 25);

contentPane.add(comboBoxFlatNo);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt = "select flat\_no from flats";

ResultSet rs1 = stmt.executeQuery(txt);

**while**(rs1.next()) {

comboBoxFlatNo.addItem(rs1.getInt(1));

}

comboBoxFlatNo.setSelectedItem(**null**);

}

**catch**(Exception e) {

e.printStackTrace();

}

**if**(login.*userType*.equals("admin")) {

btnAddPaymentsButton = **new** JButton("Add");

btnAddPaymentsButton.setBounds(252, 233, 89, 23);

contentPane.add(btnAddPaymentsButton);

btnAddPaymentsButton.addActionListener(action);

btnUpdatePaymentsButton = **new** JButton("Update");

btnUpdatePaymentsButton.setBounds(255, 233, 101, 23);

contentPane.add(btnUpdatePaymentsButton);

btnUpdatePaymentsButton.addActionListener(action);

btnDeletePaymentsButton = **new** JButton("Delete");

btnDeletePaymentsButton.setBounds(377, 233, 114, 23);

contentPane.add(btnDeletePaymentsButton);

btnDeletePaymentsButton.addActionListener(action);

btnClearButton = **new** JButton("Clear");

btnClearButton.addActionListener(action);

btnClearButton.setBounds(377, 233, 89, 23);

contentPane.add(btnClearButton);

btnCancelButton = **new** JButton("Cancel");

btnCancelButton.setBounds(138, 233, 89, 23);

contentPane.add(btnCancelButton);

btnCancelButton.addActionListener(action);

btnUpdatePaymentsButton.setVisible(**false**);

btnDeletePaymentsButton.setVisible(**false**);

btnCancelButton.setVisible(**false**);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select payment\_id,payment\_amount,payment\_date from payments";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Payment Id");

tableModel.addColumn("Payment Amount");

tableModel.addColumn("Payment Date");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,286,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddPaymentsButton.setVisible(**false**);

btnUpdatePaymentsButton.setVisible(**true**);

btnDeletePaymentsButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select payment\_id,payment\_amount,payment\_type,payment\_date,flat\_no from payments where payment\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldPaymentsId.setText(rs.getString(1));

textFieldPaymentsAmount.setText(rs.getString(2));

textFieldPaymentsType.setText(rs.getString(3));

paymentDate.setDateTimeStrict(rs.getTimestamp(4).toLocalDateTime());

comboBoxFlatNo.setSelectedItem(rs.getInt(5));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

**else** **if**(login.*userType*.equals("user")) {

textFieldPaymentsId.setEditable(**false**);

textFieldPaymentsType.setEditable(**false**);

textFieldPaymentsAmount.setEditable(**false**);

comboBoxFlatNo.setEditable(**false**);

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

txt = "select payment\_id,payment\_amount,payment\_date from payments where flat\_no ="+login.*flatNo*+"";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Payment Id");

tableModel.addColumn("Payment Amount");

tableModel.addColumn("Payment Date");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,290,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select payment\_id,payment\_amount,payment\_type,payment\_date,flat\_no from payments where payment\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldPaymentsId.setText(rs.getString(1));

textFieldPaymentsAmount.setText(rs.getString(2));

textFieldPaymentsType.setText(rs.getString(3));

paymentDate.setDateTimeStrict(rs.getTimestamp(4).toLocalDateTime());

comboBoxFlatNo.setSelectedItem(rs.getString(5));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

**public** **class** SwingAction **extends** AbstractAction {

**public** SwingAction() {

putValue(***NAME***, "SwingAction");

putValue(***SHORT\_DESCRIPTION***, "Some short description");

}

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

**try** {

MyDbConnection connect = **new** MyDbConnection();

Connection con = connect.getConnection();

Statement stmt=con.createStatement();

String txt="";

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

txt = "insert into payments values('"+textFieldPaymentsId.getText()+"','"+textFieldPaymentsAmount.getText()+"',to\_date('"+paymentDate.getDateTimeStrict().format(DateTimeFormatter.*ofPattern*("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi'),'"+textFieldPaymentsType.getText()+"','"+comboBoxFlatNo.getSelectedItem()+"')";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Added Payment");

clearText();

}

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

txt = "delete from payments where payment\_id ='"+textFieldPaymentsId.getText()+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Payment Deleted");

clearText();

}

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

txt = "update payments set payment\_id ='"+Integer.*parseInt*(textFieldPaymentsId.getText())+"',payment\_amount='"+Integer.*parseInt*(textFieldPaymentsAmount.getText())+"',payment\_type = '"+textFieldPaymentsType.getText()+"',payment\_date = to\_date('"+paymentDate.getDateTimeStrict().format(DateTimeFormatter.*ofPattern*("dd-MMM-yy HH:mm"))+"','dd-mon-yy hh24:mi'),flat\_no = '"+comboBoxFlatNo.getSelectedItem()+"' where payment\_id='"+Integer.*parseInt*(textFieldPaymentsId.getText())+"'";

stmt.executeQuery(txt);

JOptionPane.*showMessageDialog*(**null**,"Payment Updated");

clearText();

}

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

btnAddPaymentsButton.setVisible(**true**);

btnUpdatePaymentsButton.setVisible(**false**);

btnDeletePaymentsButton.setVisible(**false**);

btnCancelButton.setVisible(**false**);

clearText();

btnClearButton.setVisible(**true**);

}

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

clearText();

}

txt = "select payment\_id,payment\_amount,payment\_date from payments";

ResultSet rs = stmt.executeQuery(txt);

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

table\_1 = **new** JTable(tableModel);

tableModel.addColumn("Payment Id");

tableModel.addColumn("Payment Amount");

tableModel.addColumn("Payment Date");

**int** i=0;

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

tableModel.insertRow(i,**new** Object[] { rs.getString(1),rs.getString(2),rs.getString(3) });

System.***out***.println(rs.getString(2));

i++;

}

table\_1.setVisible(**true**);

js.setBounds(20,286,640,150);

getContentPane().add(js);

js.setViewportView(table\_1);

ListSelectionModel model = table\_1.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

**public** **void** valueChanged(ListSelectionEvent e) {

**if**(!model.isSelectionEmpty()) {

**int** selectedRow = model.getMinSelectionIndex();

btnAddPaymentsButton.setVisible(**false**);

btnUpdatePaymentsButton.setVisible(**true**);

btnDeletePaymentsButton.setVisible(**true**);

btnCancelButton.setVisible(**true**);

btnClearButton.setVisible(**false**);

//JOptionPane.showMessageDialog(null,"Selected Row "+table\_1.getModel().getValueAt(selectedRow,1));

**try** {

String txt = "select payment\_id,payment\_amount,payment\_type,payment\_date,flat\_no from payments where payment\_id = '"+table\_1.getModel().getValueAt(selectedRow,0)+"'";

ResultSet rs = stmt.executeQuery(txt);

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

textFieldPaymentsId.setText(rs.getString(1));

textFieldPaymentsAmount.setText(rs.getString(2));

textFieldPaymentsType.setText(rs.getString(3));

paymentDate.setDateTimeStrict(rs.getTimestamp(3).toLocalDateTime());

comboBoxFlatNo.setSelectedItem(rs.getInt(5));

}

}

**catch**(Exception exe) {

exe.printStackTrace();

}

}

}

});

}

**catch**(Exception ex) {

System.***out***.println(ex);

}

}

}

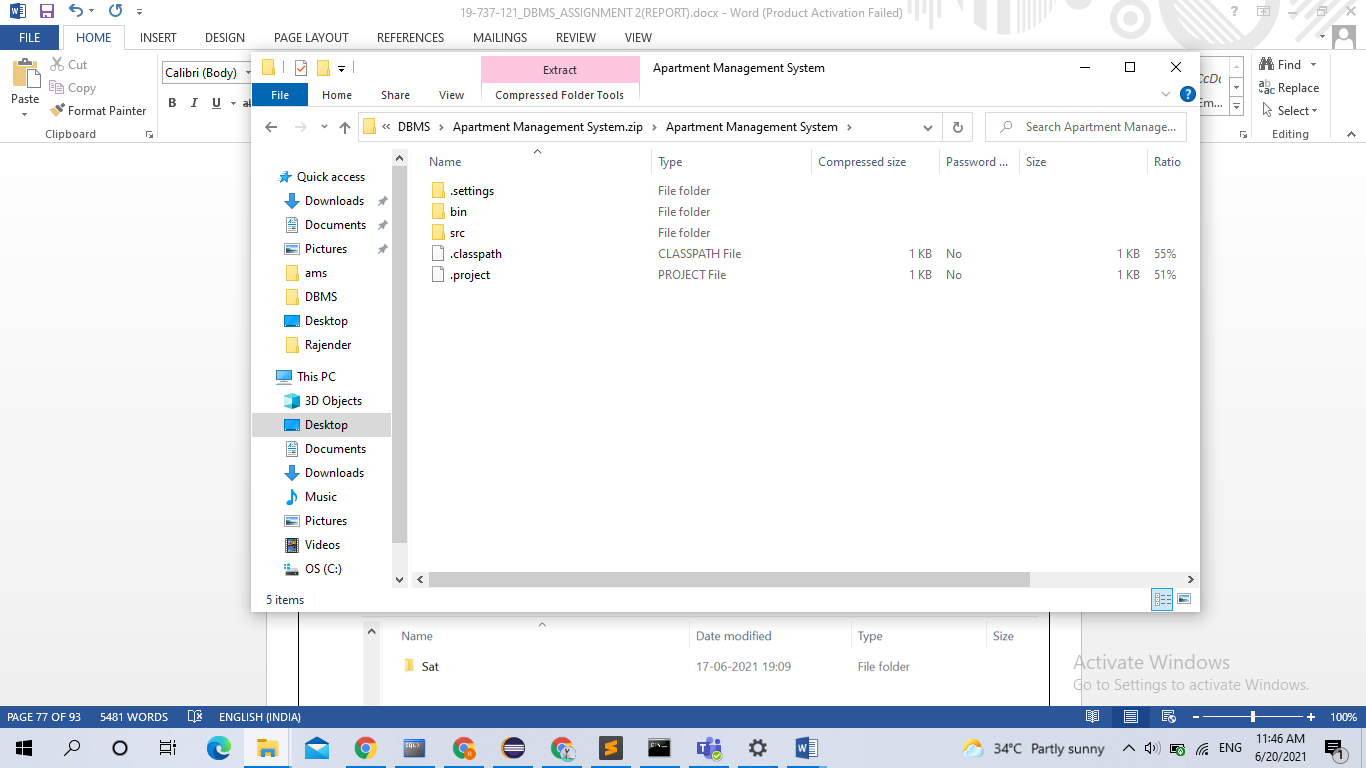
}

**Github links and folder structure**:

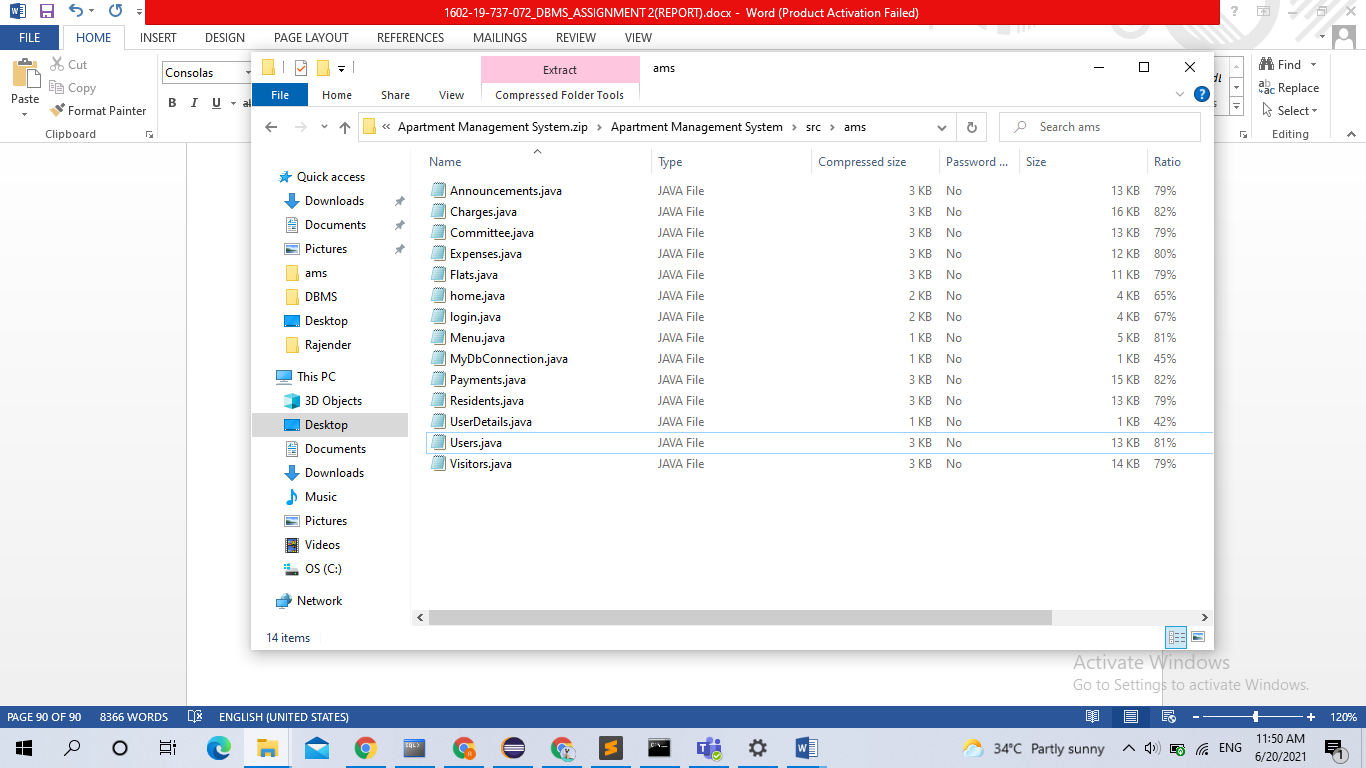
Link:

https://github.com/Harshavardhan0913/Apartment-Management-System.git

Folder structure:

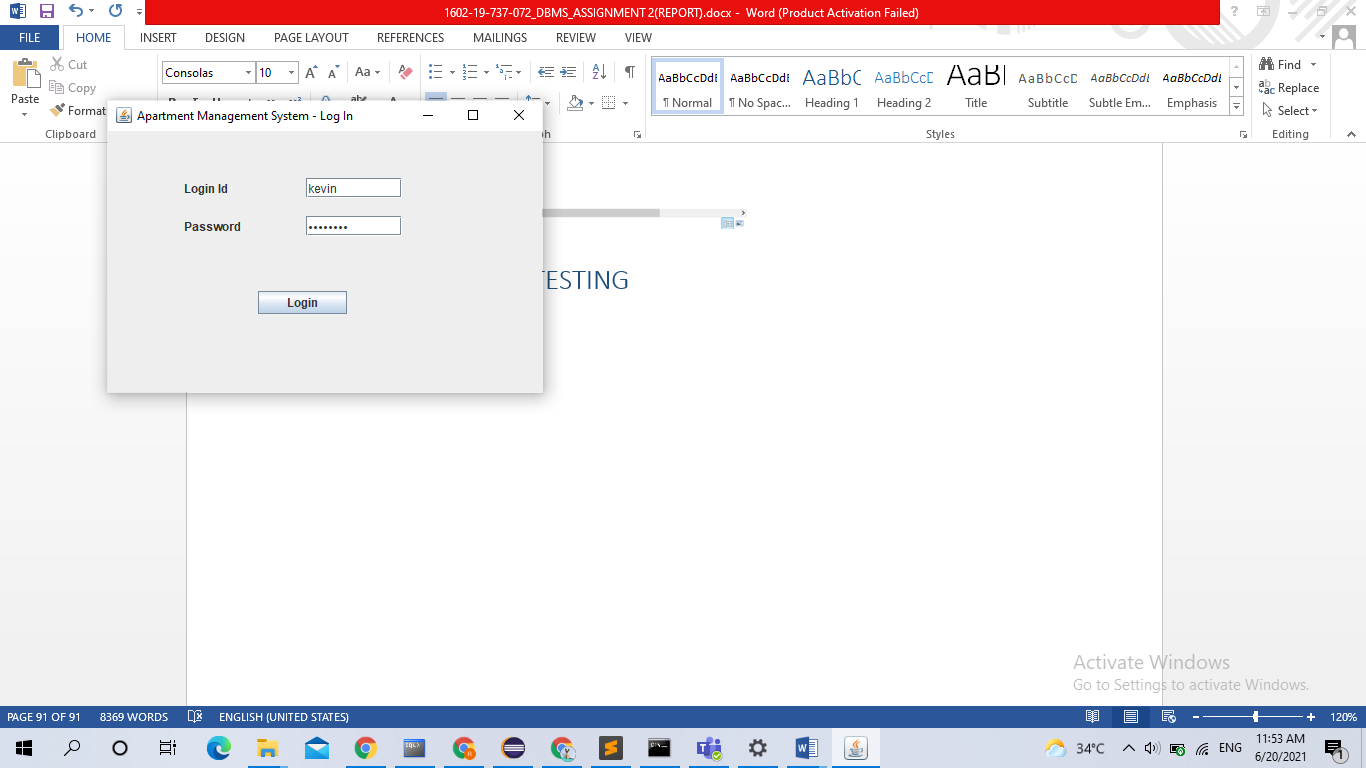
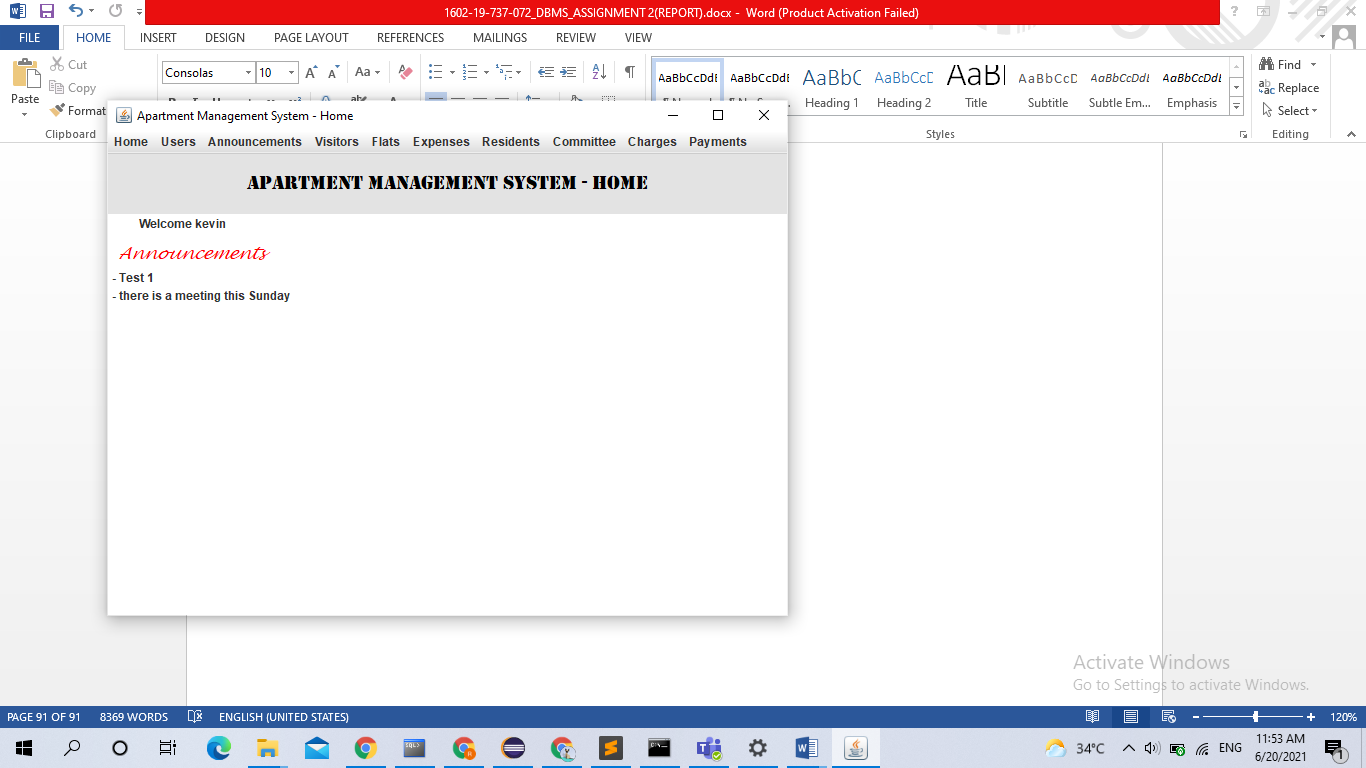
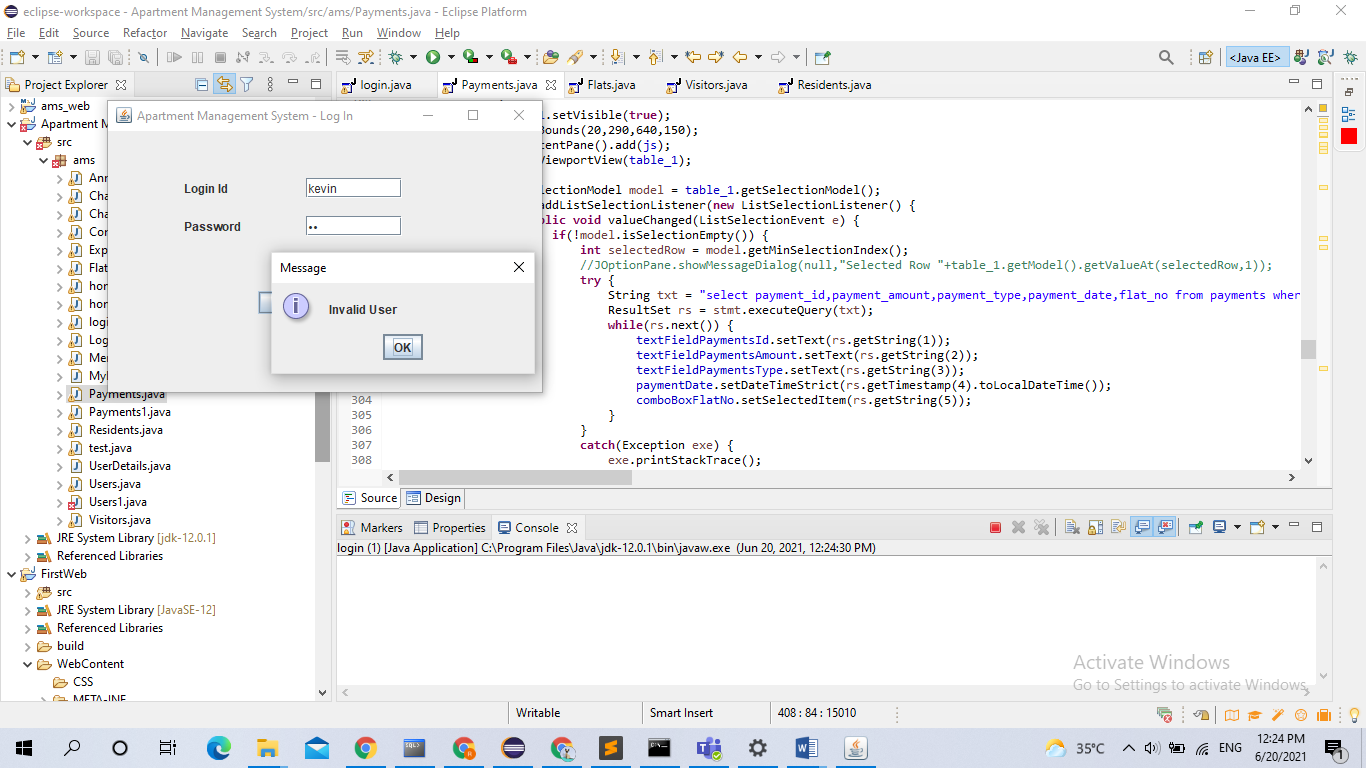
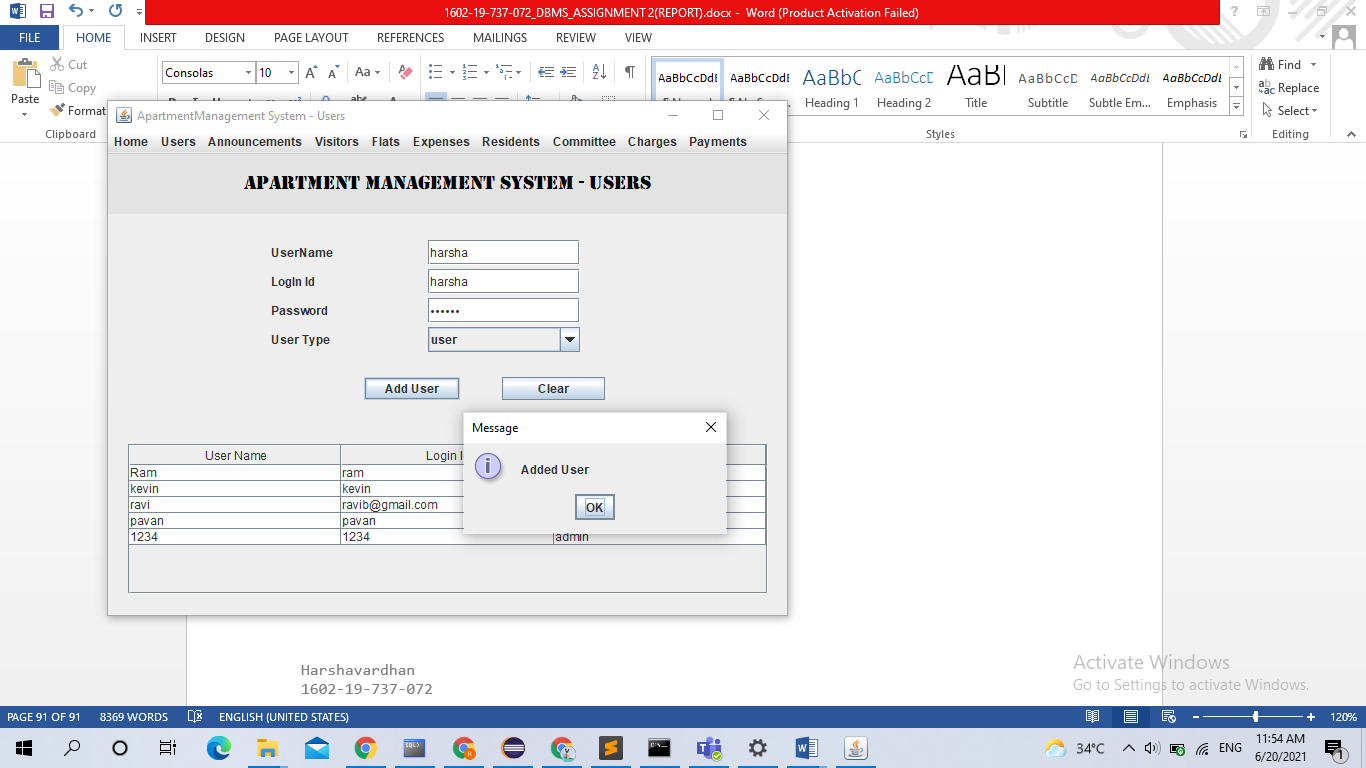
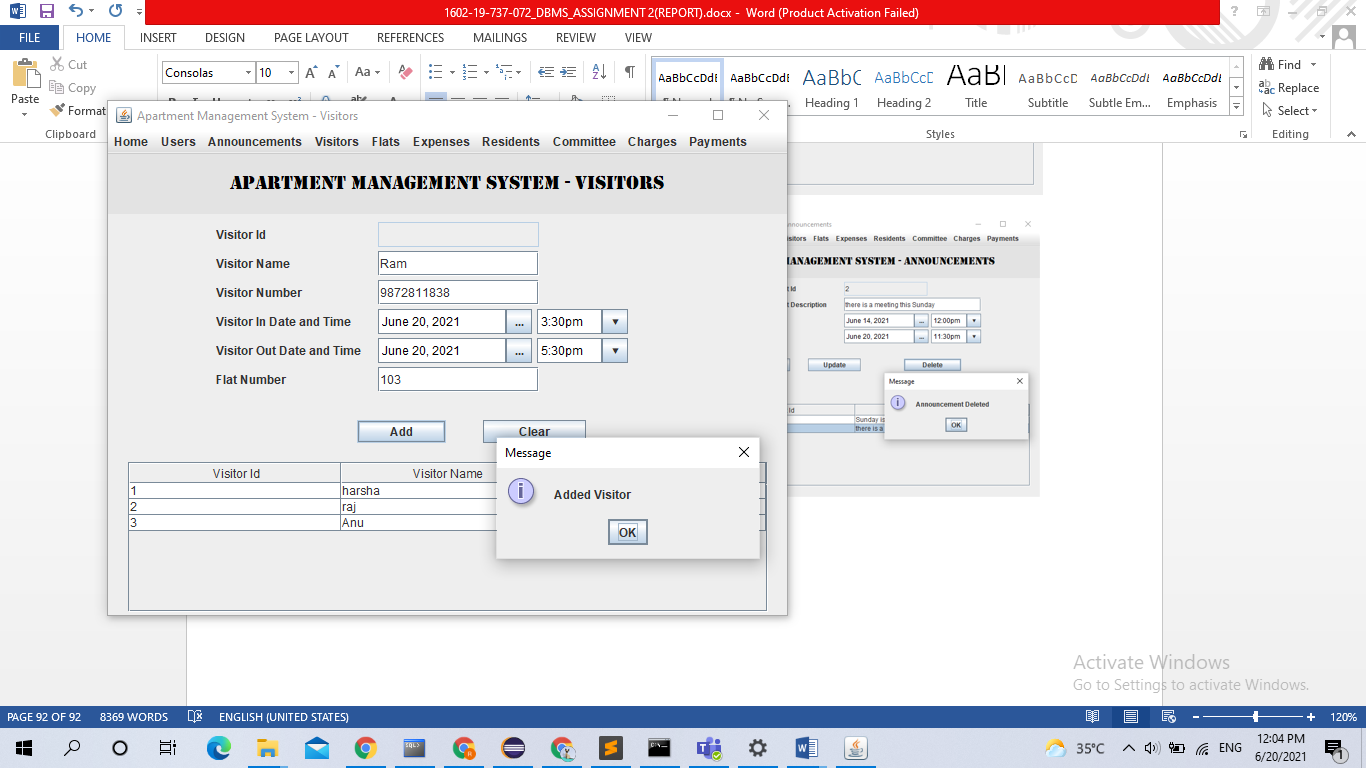
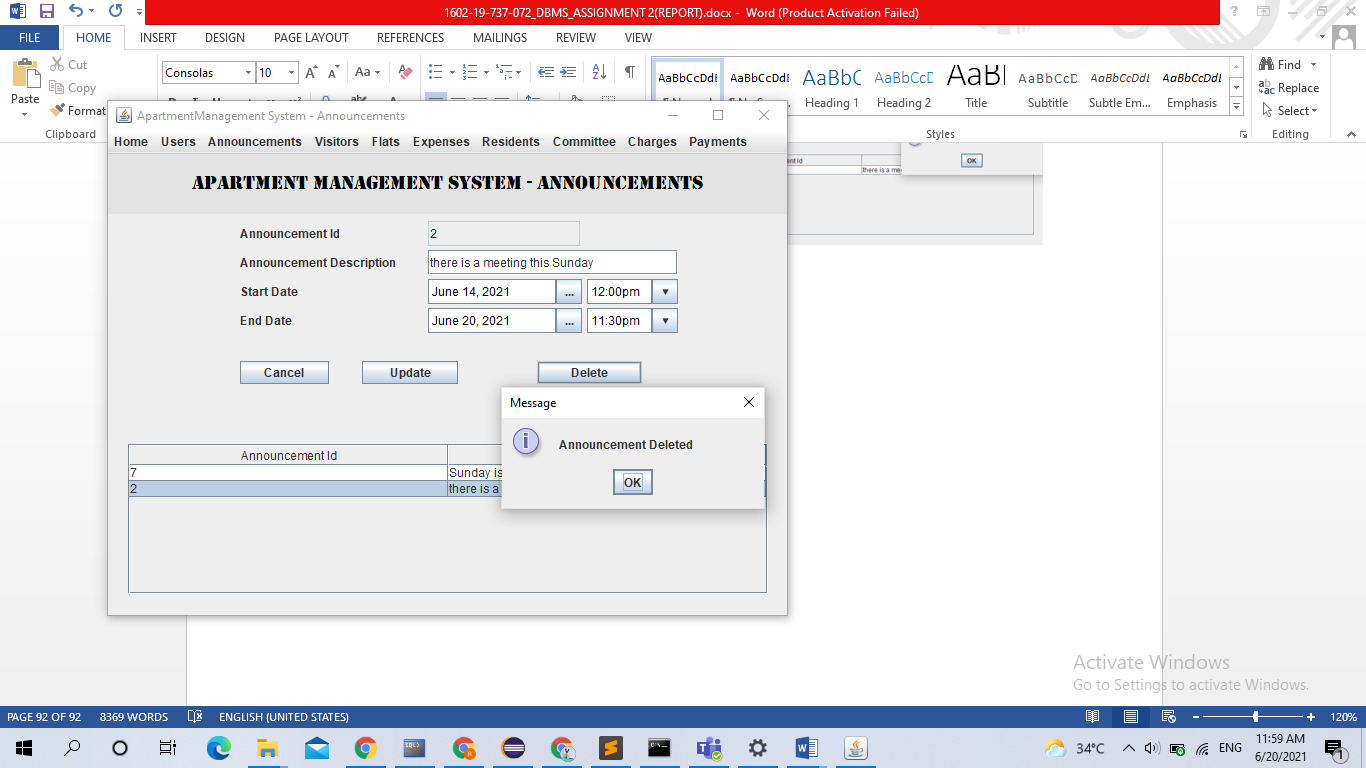
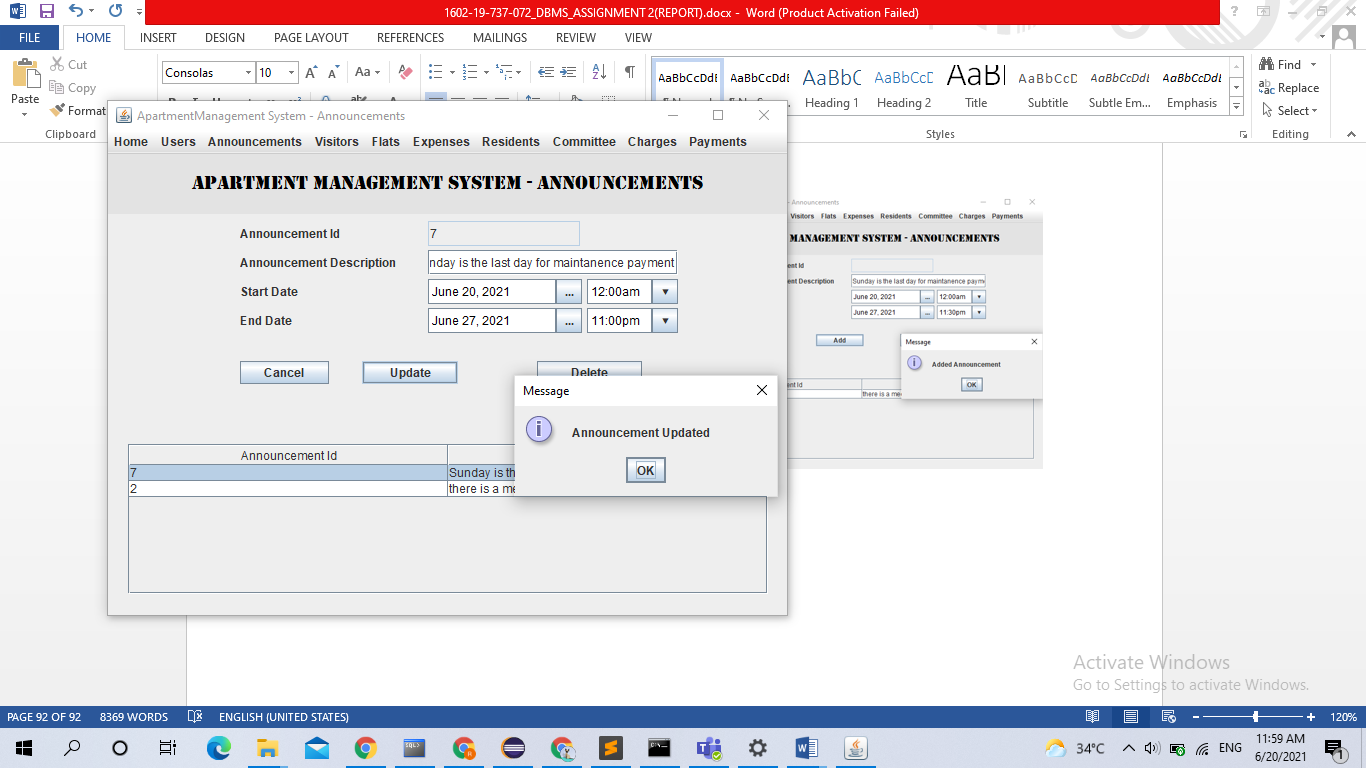
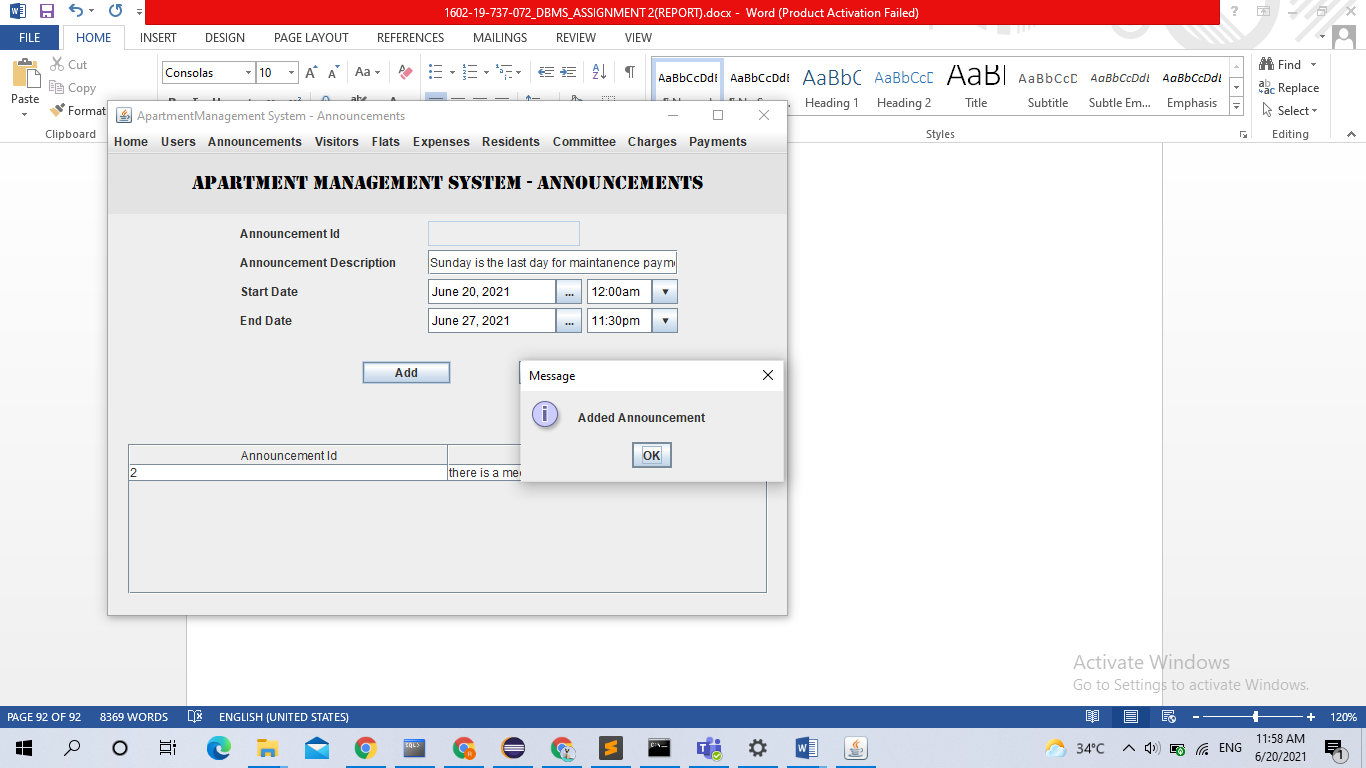
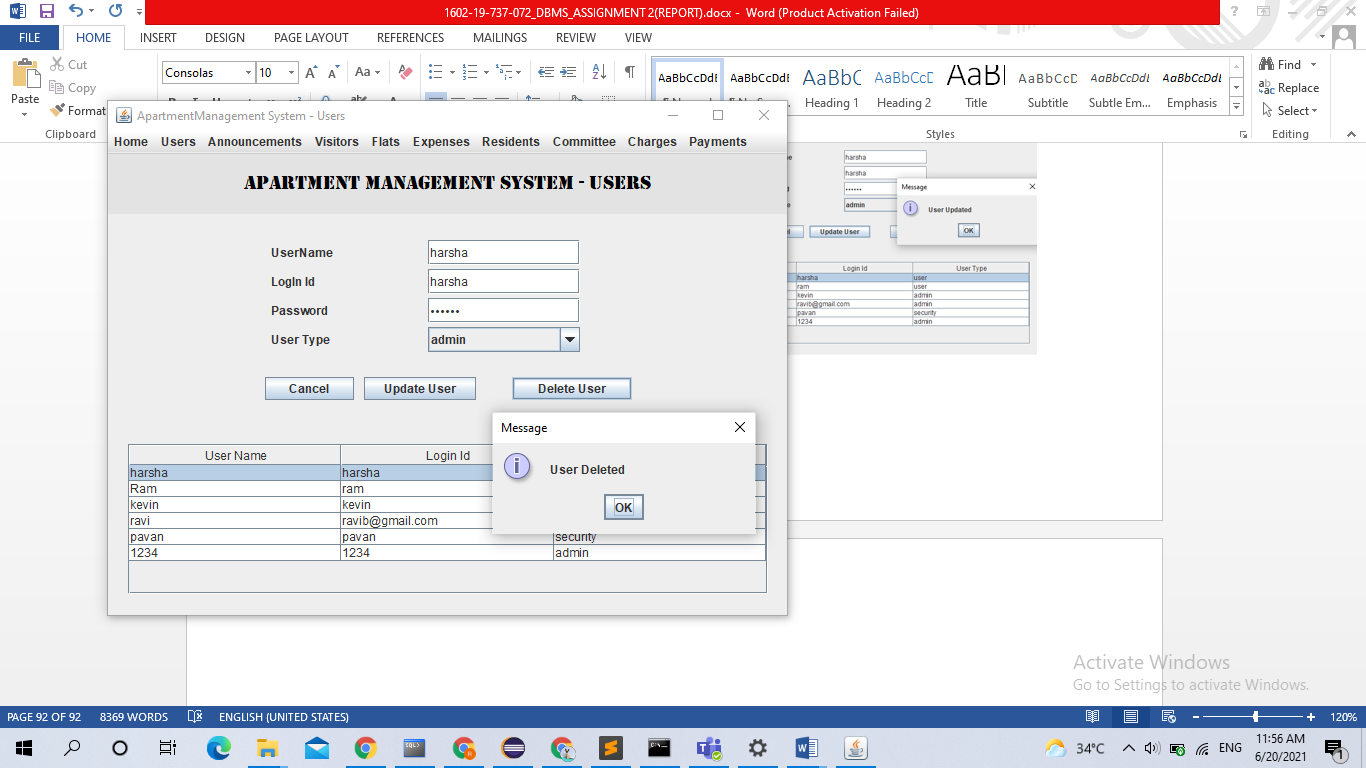
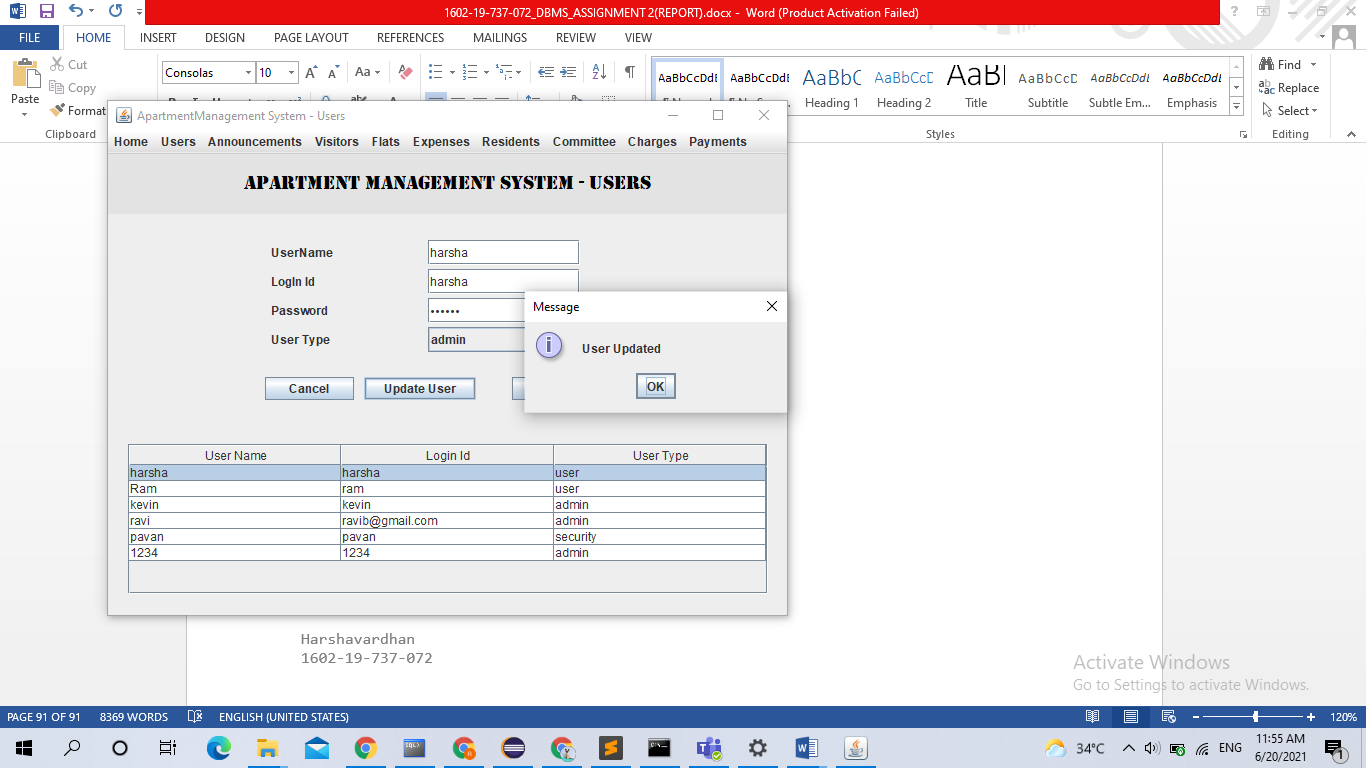
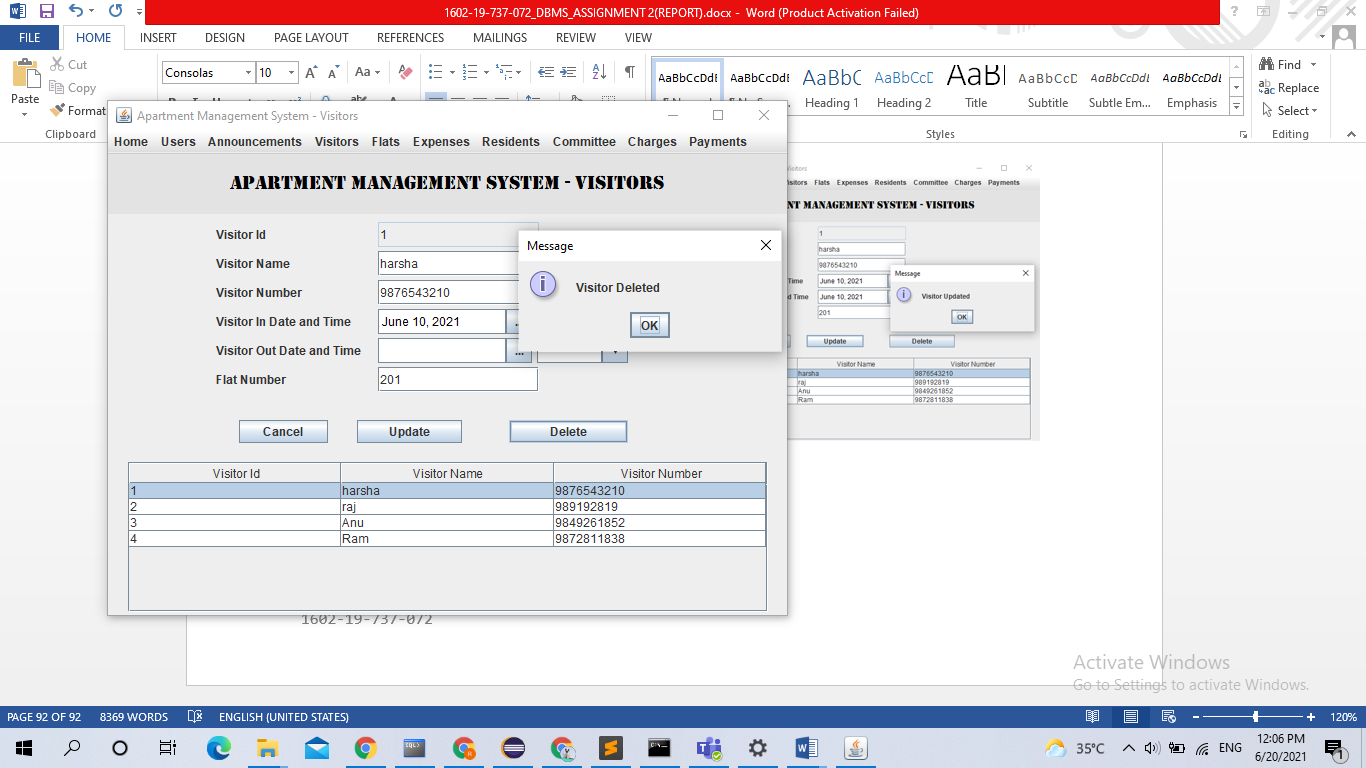
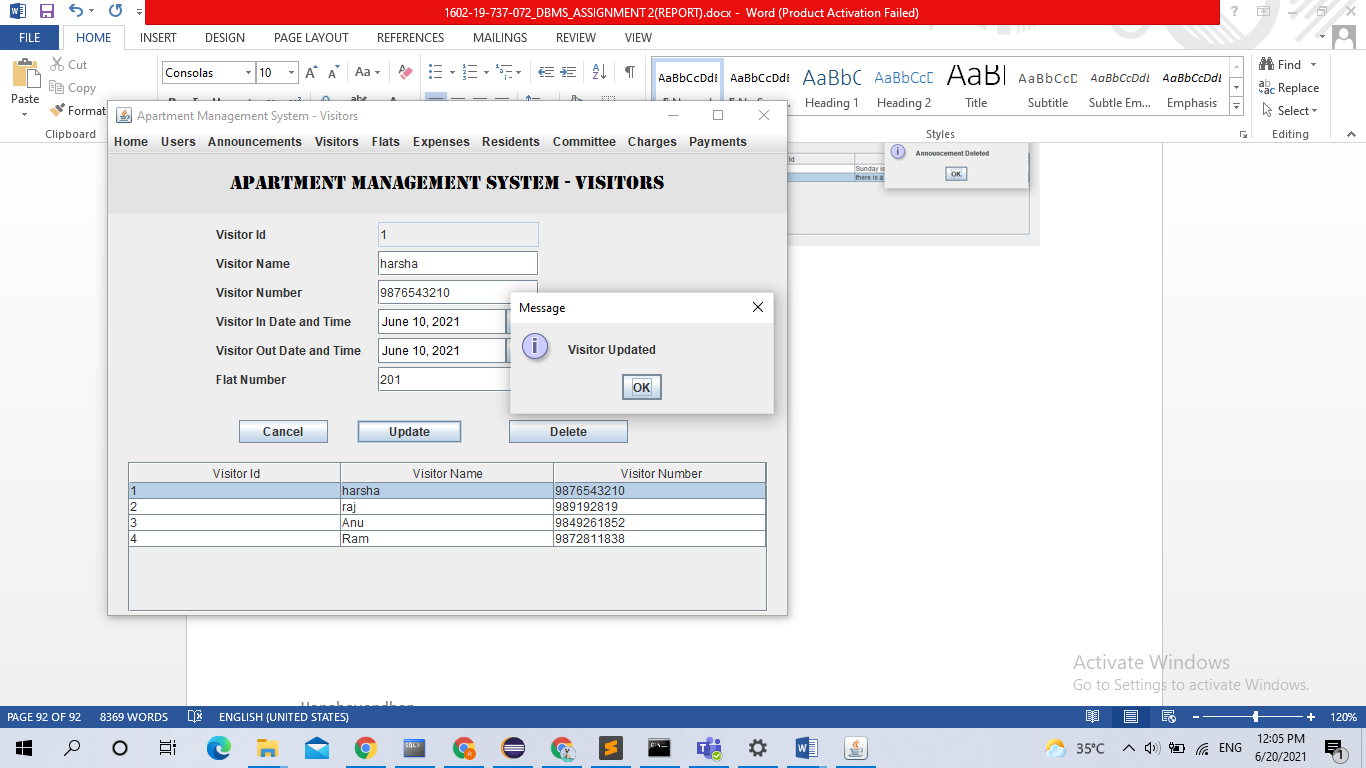
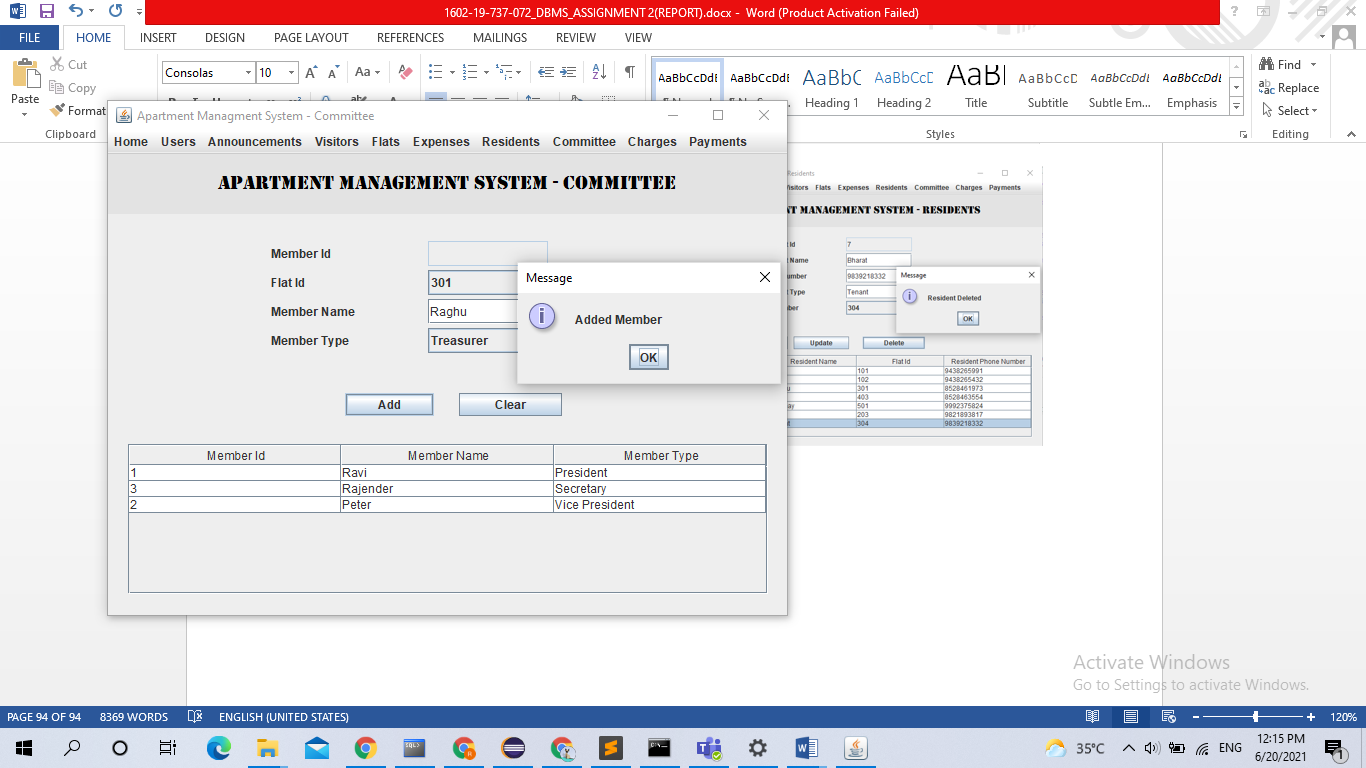
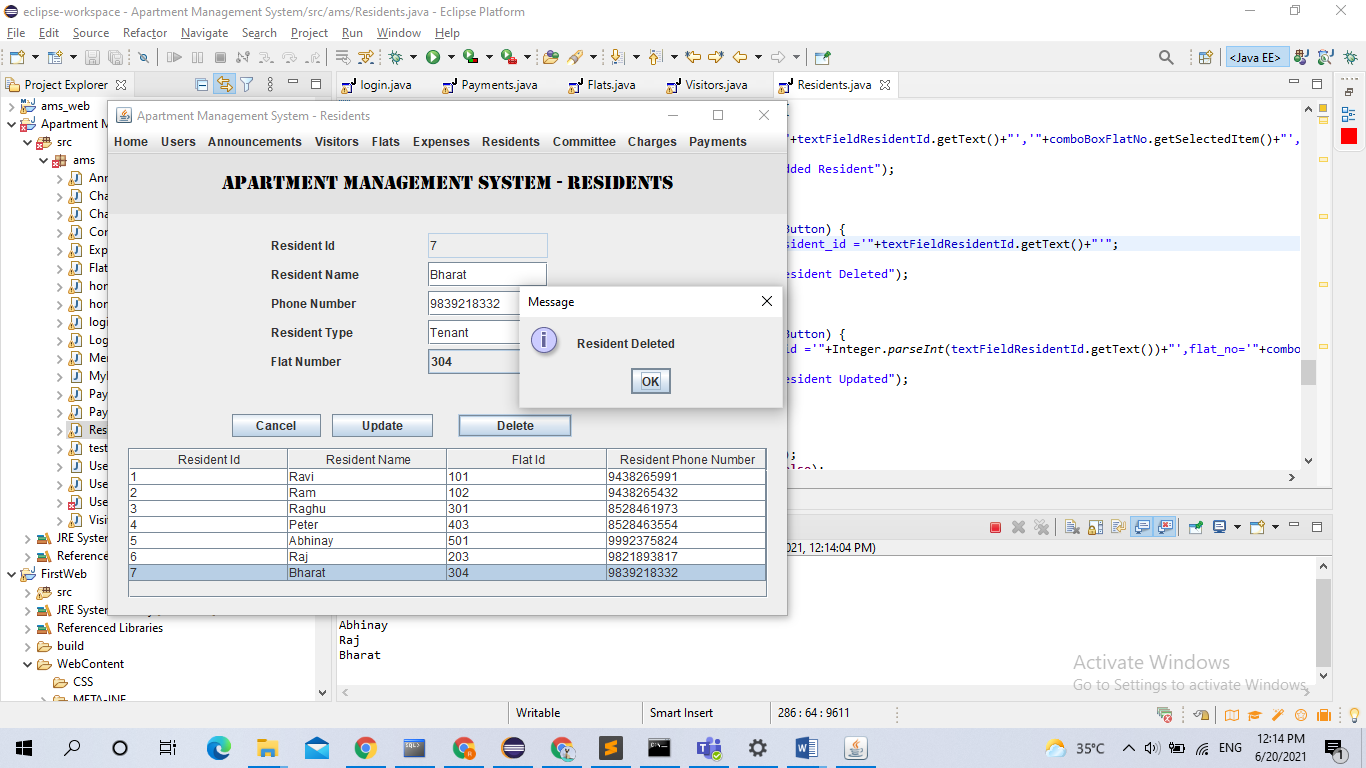
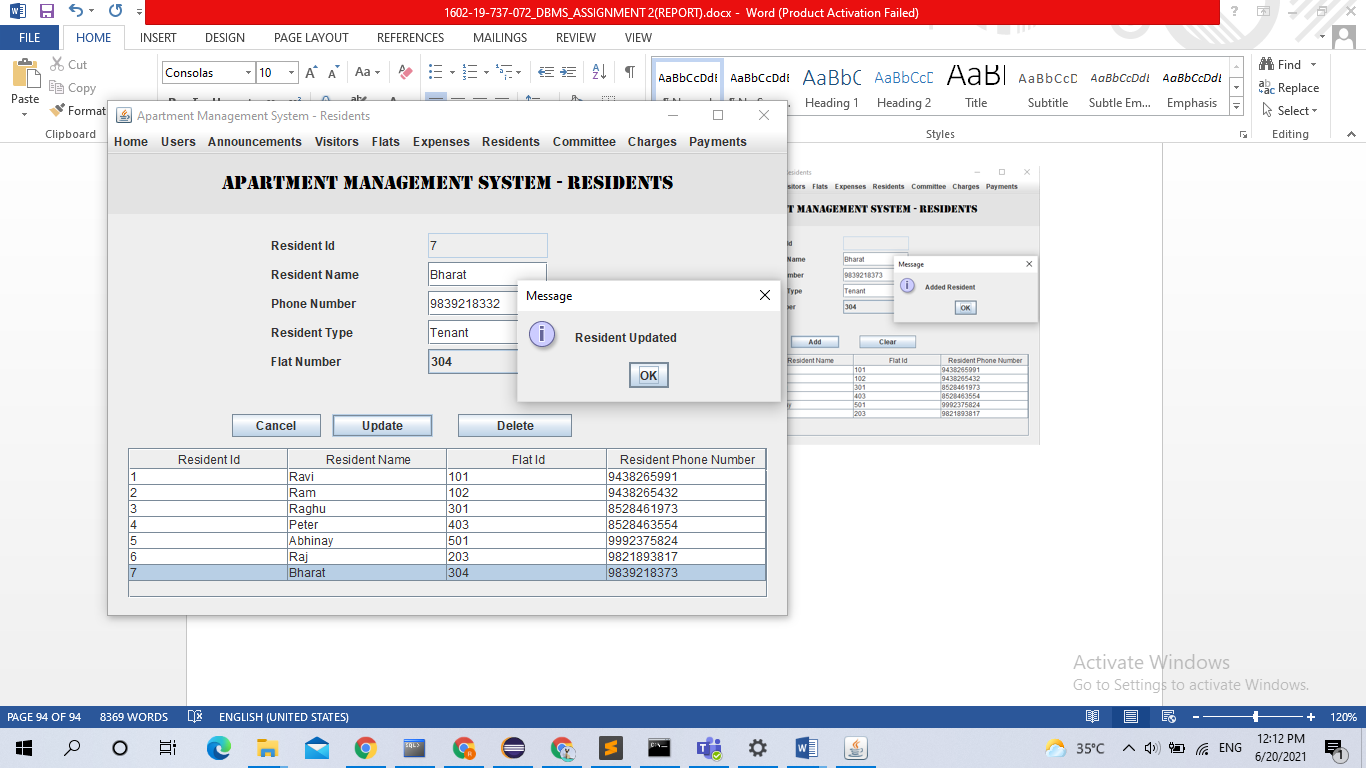
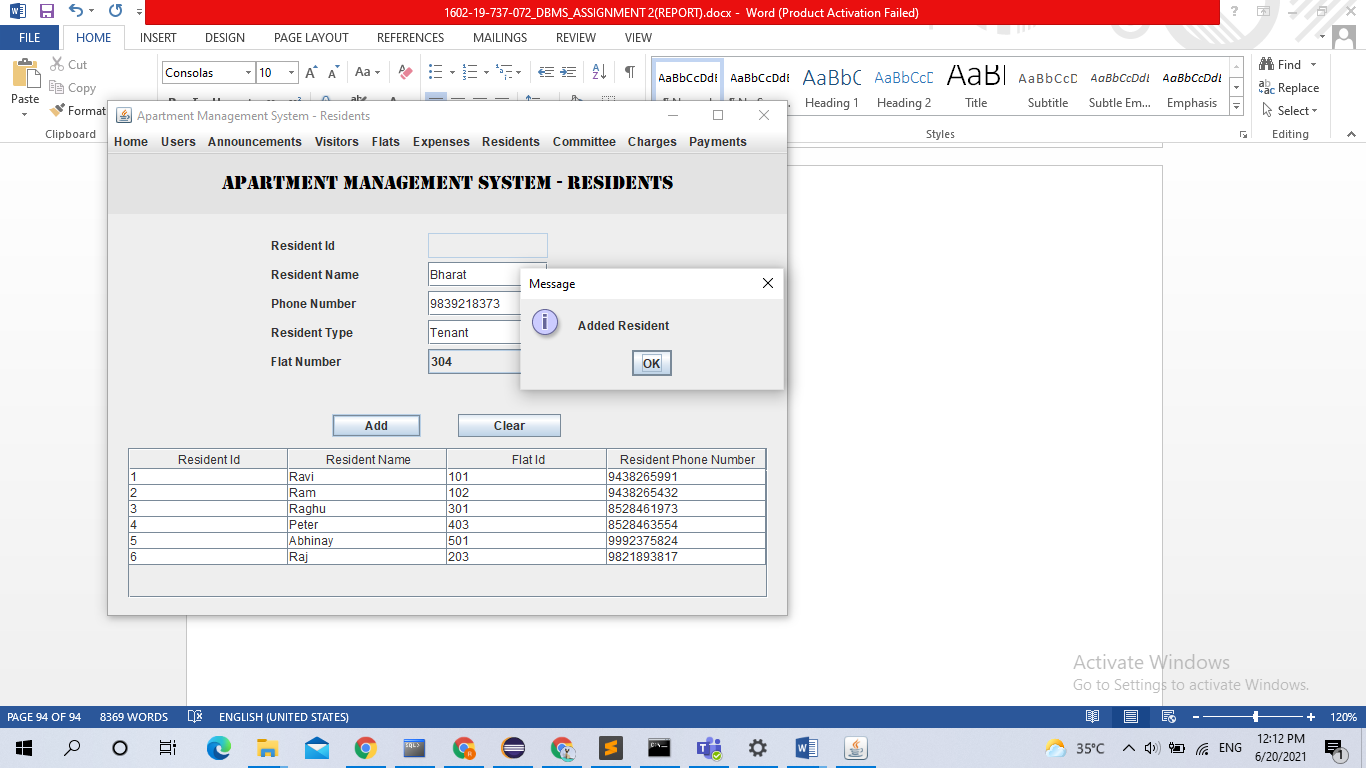
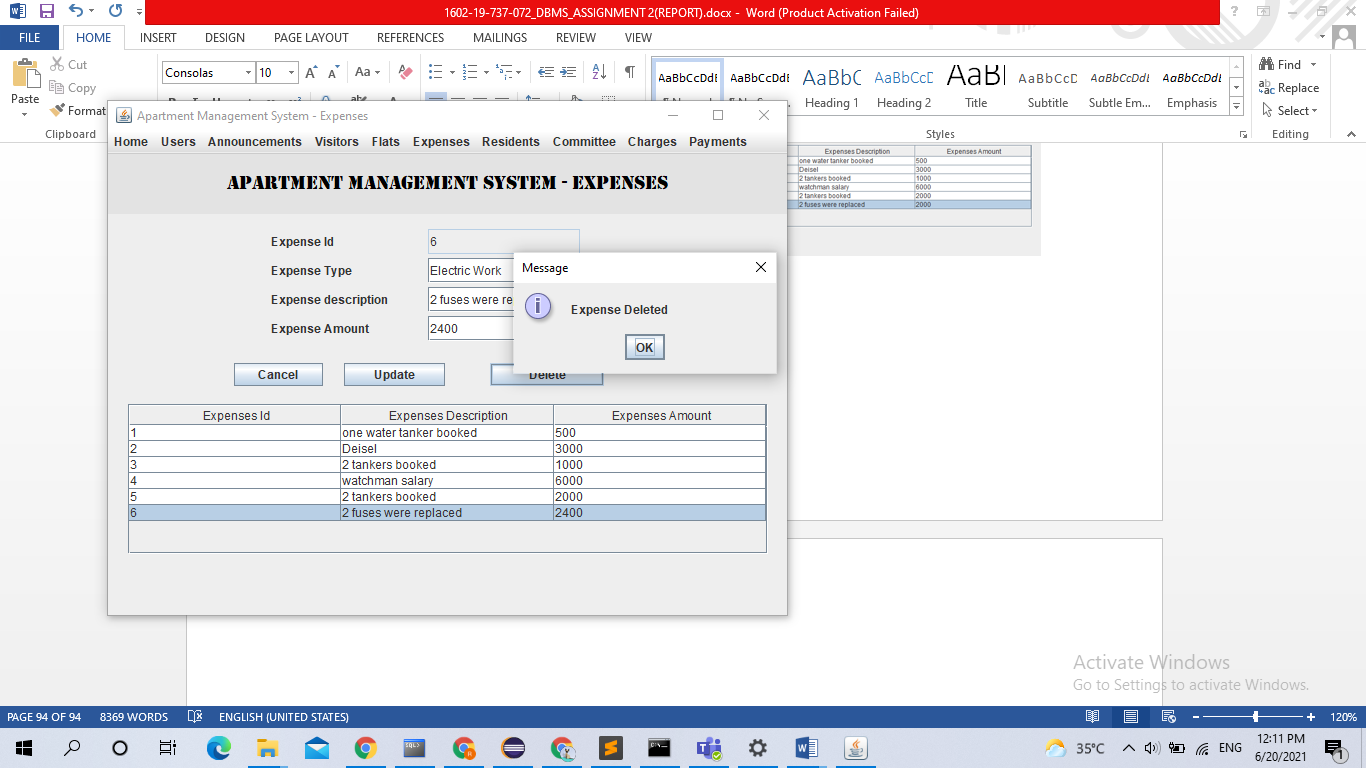
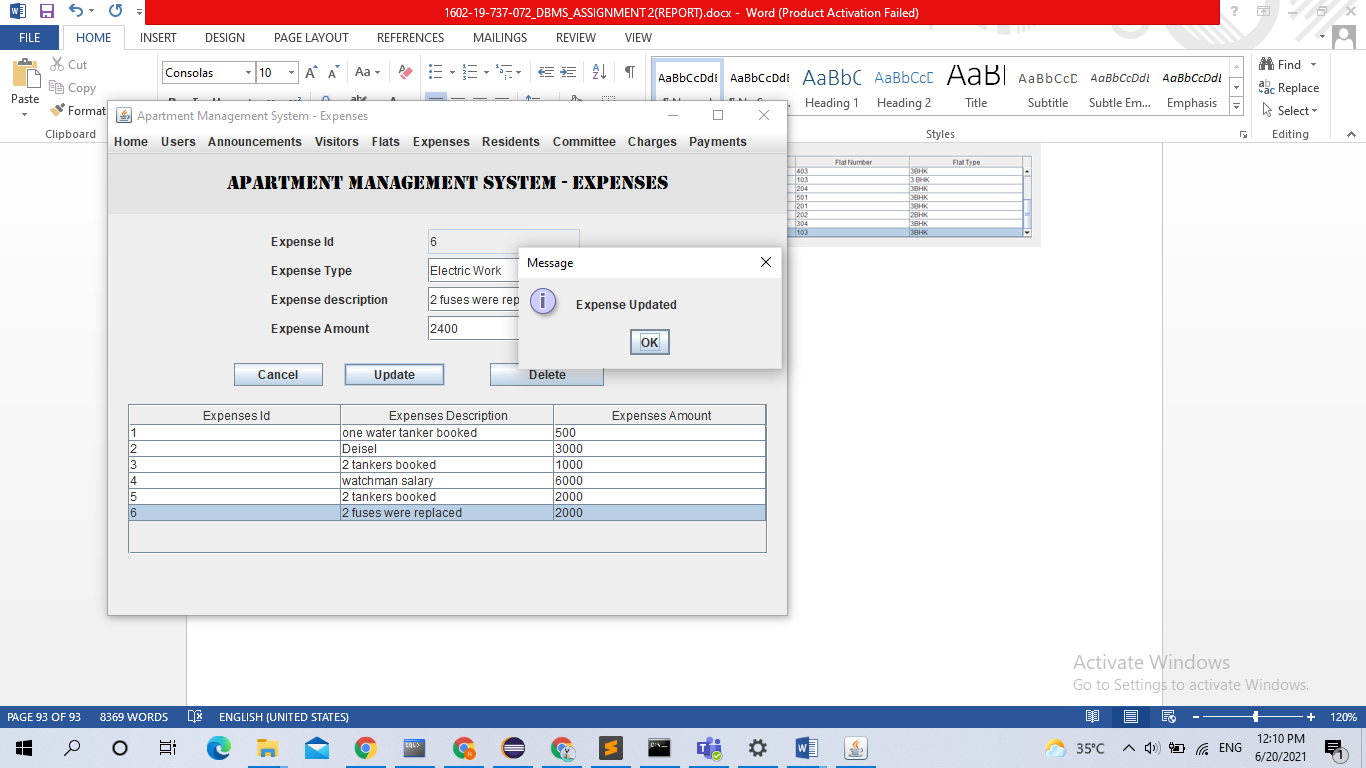
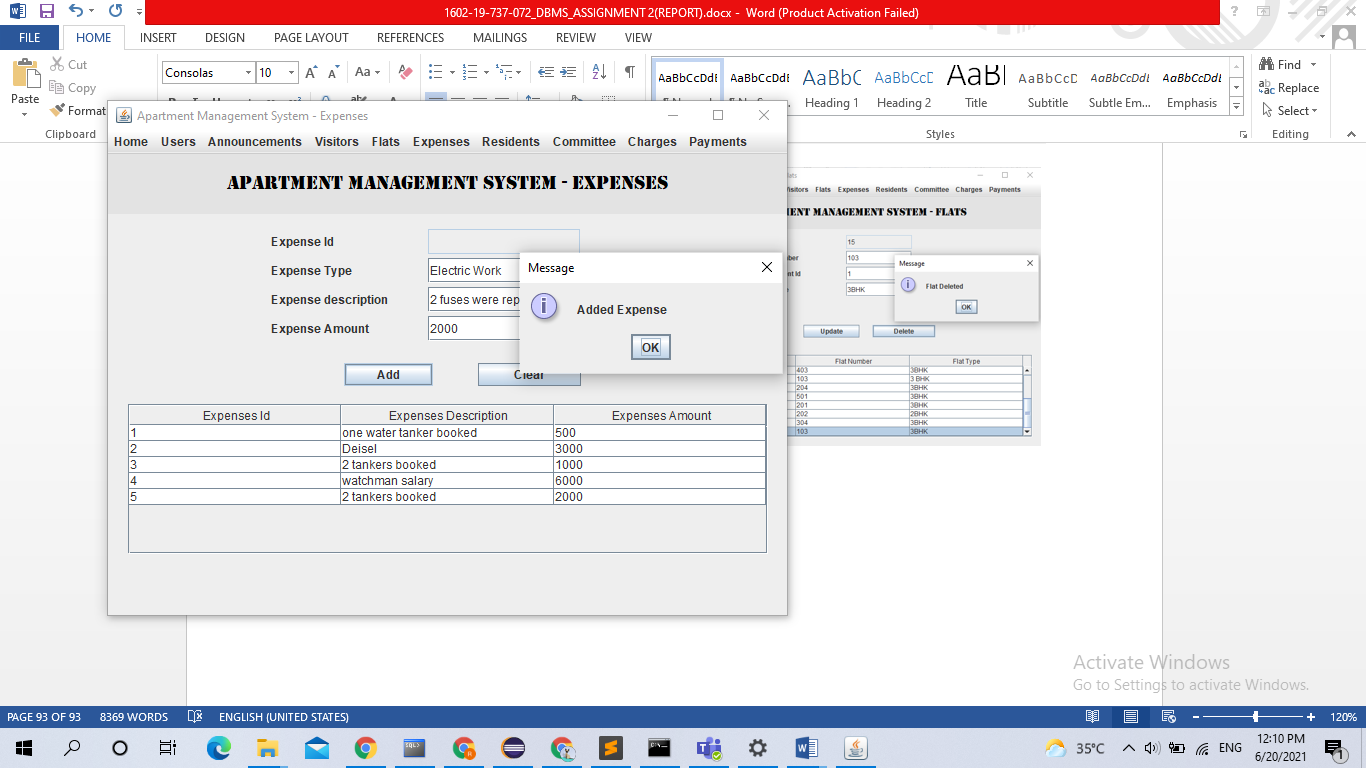
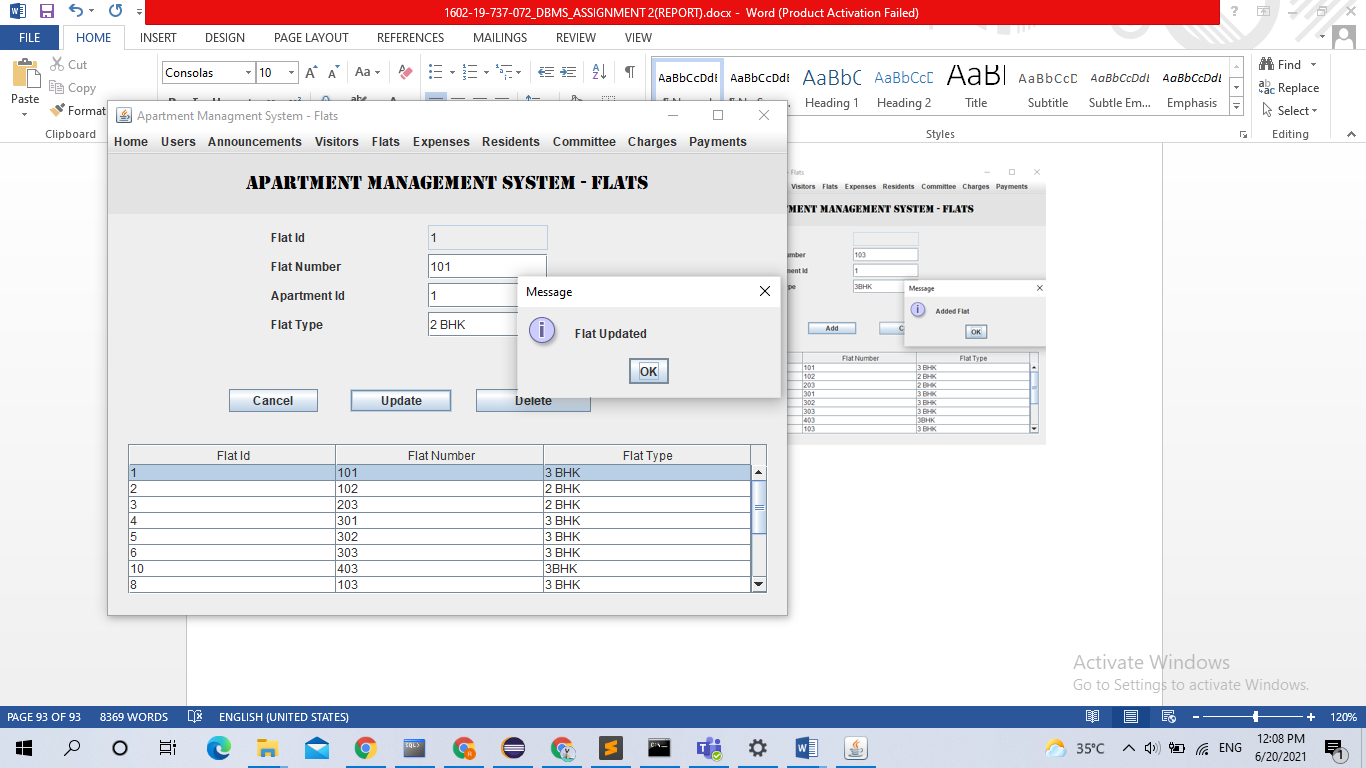
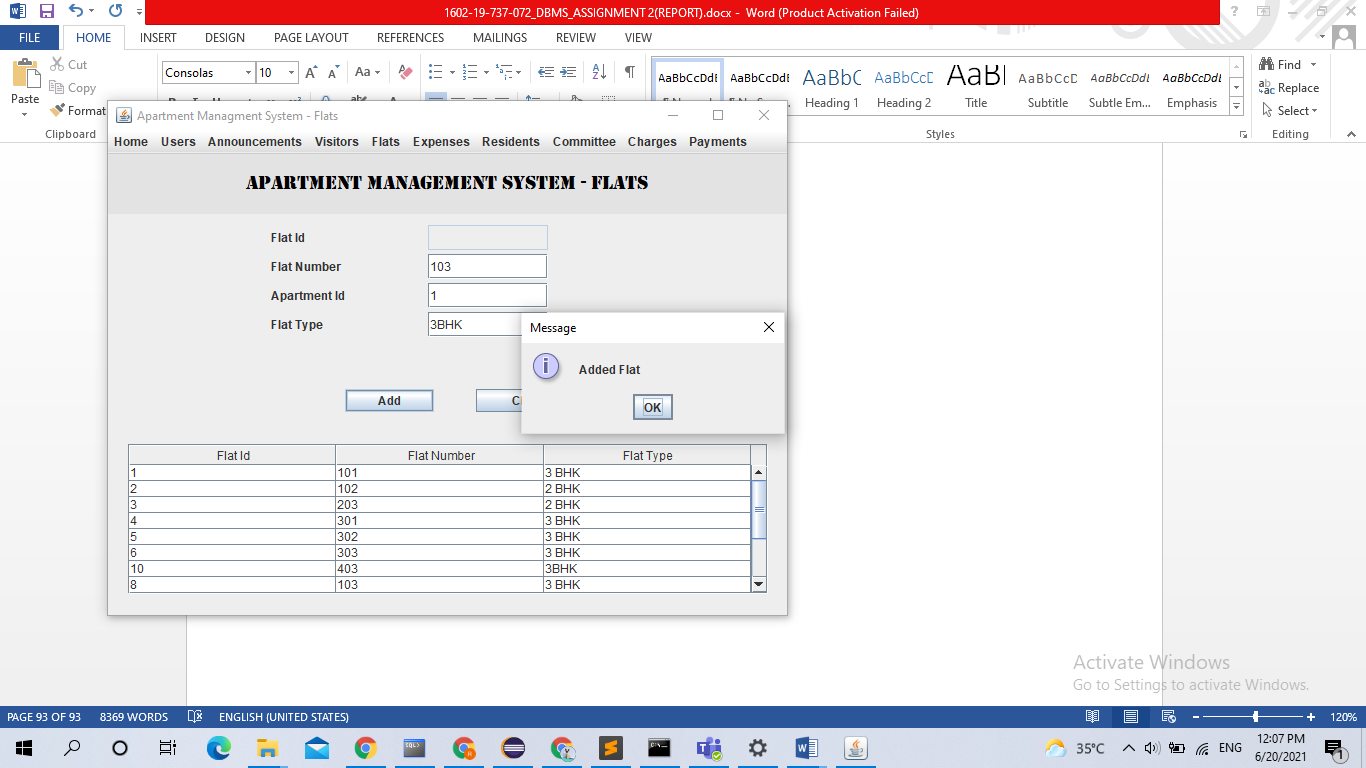
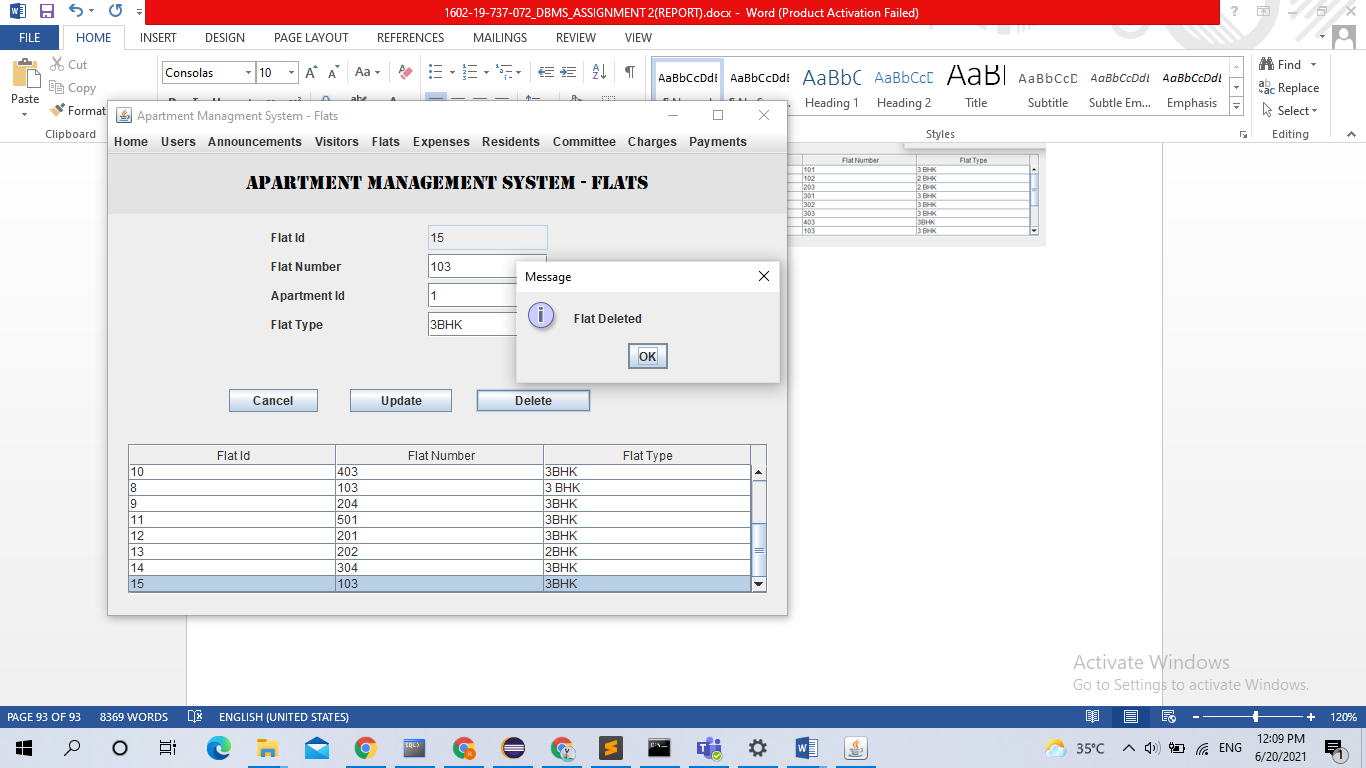
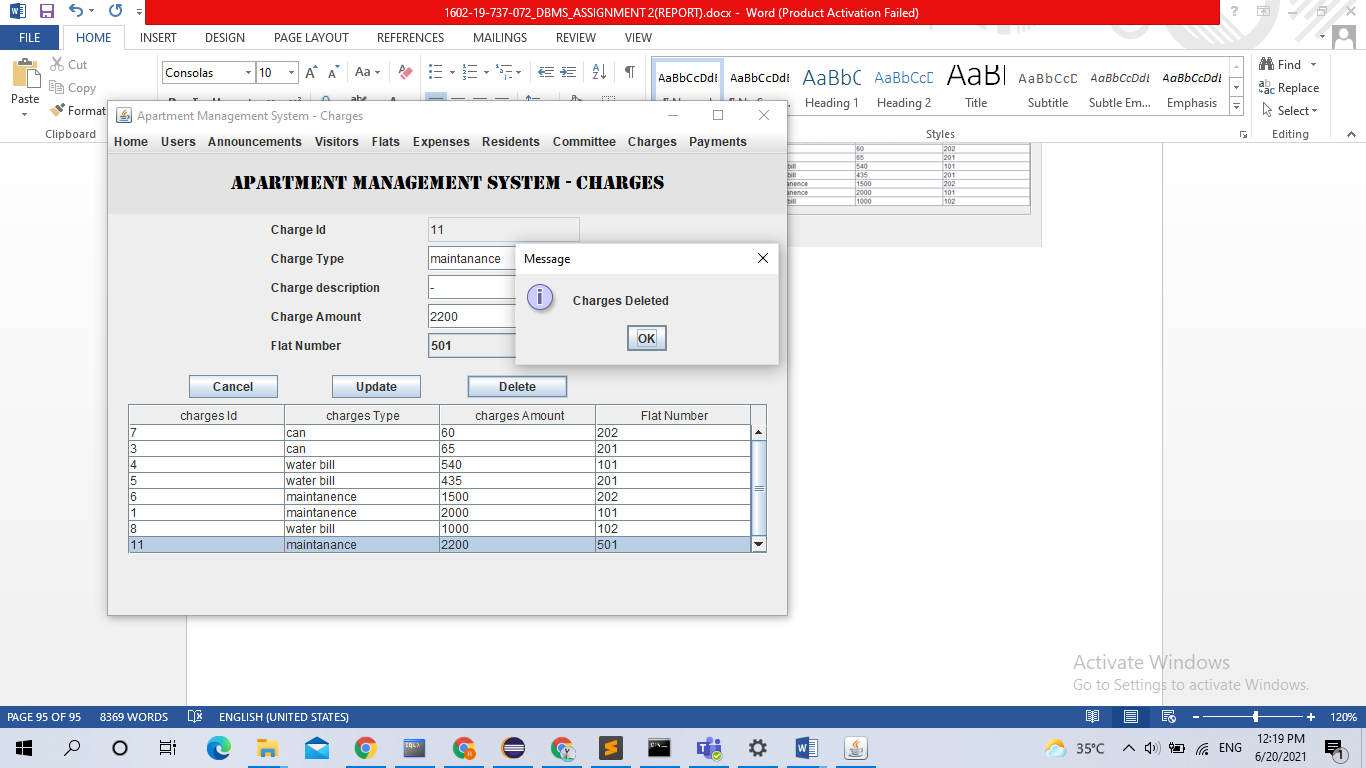
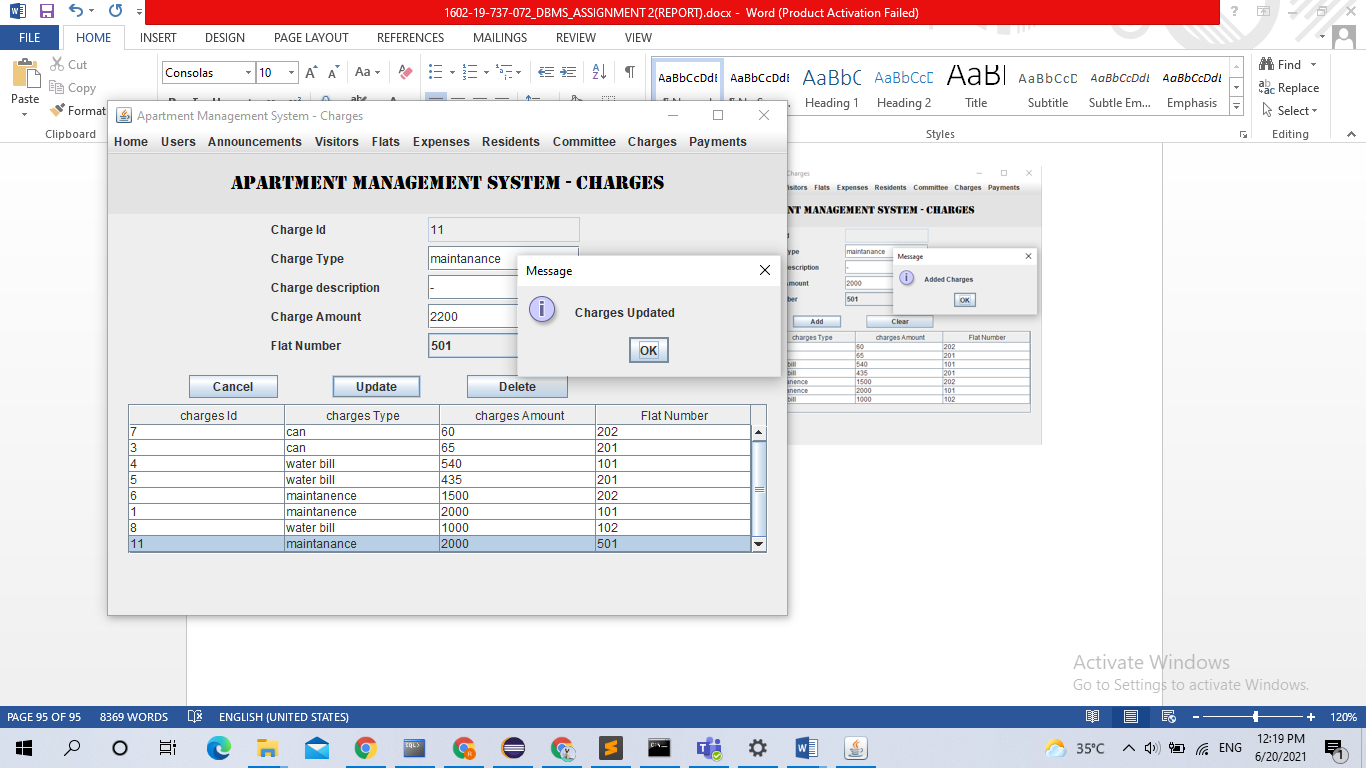
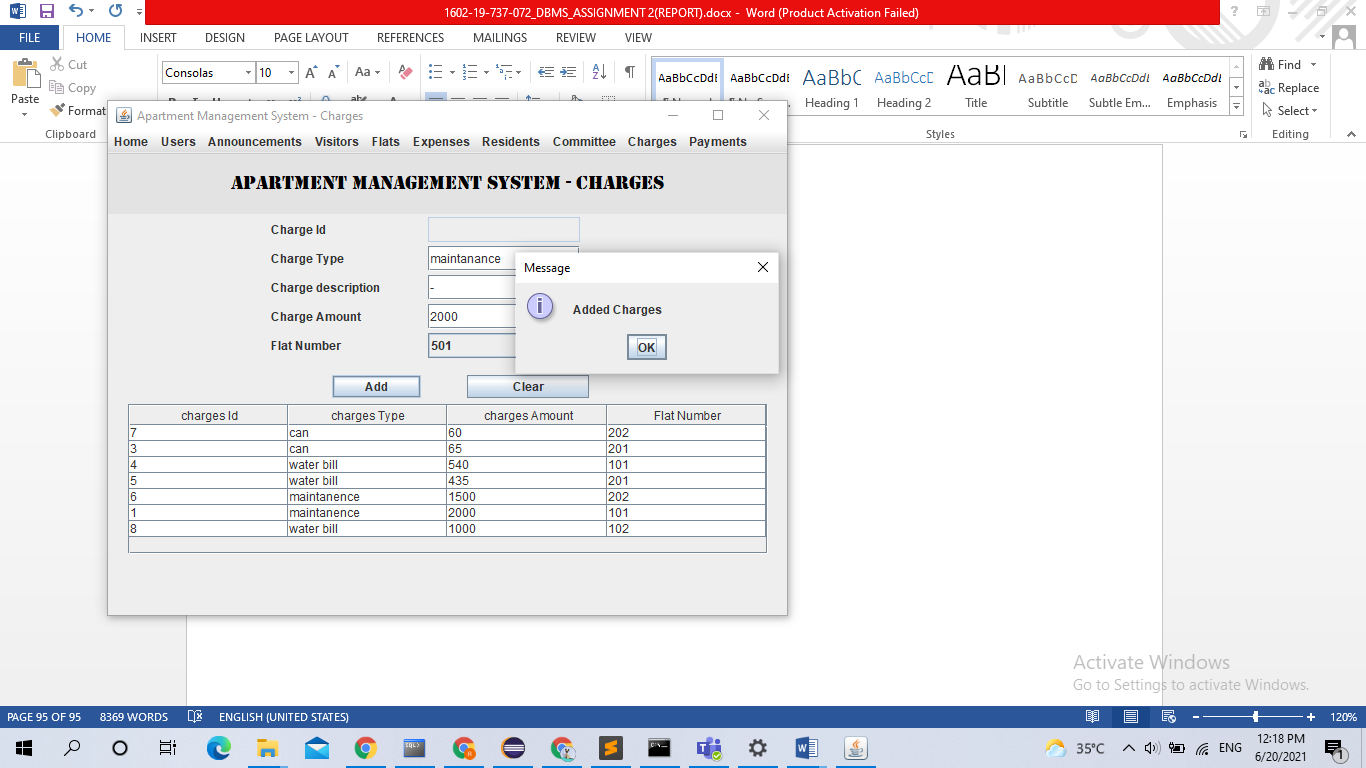
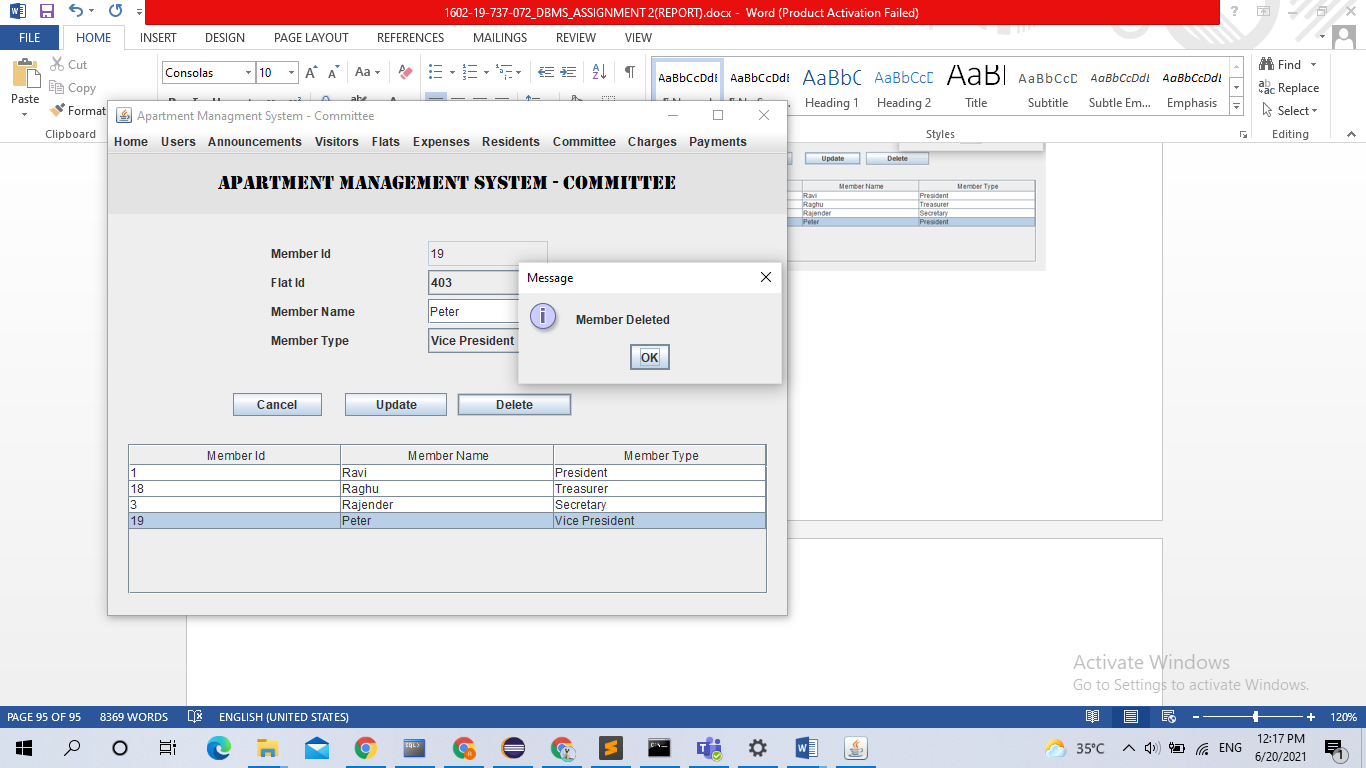
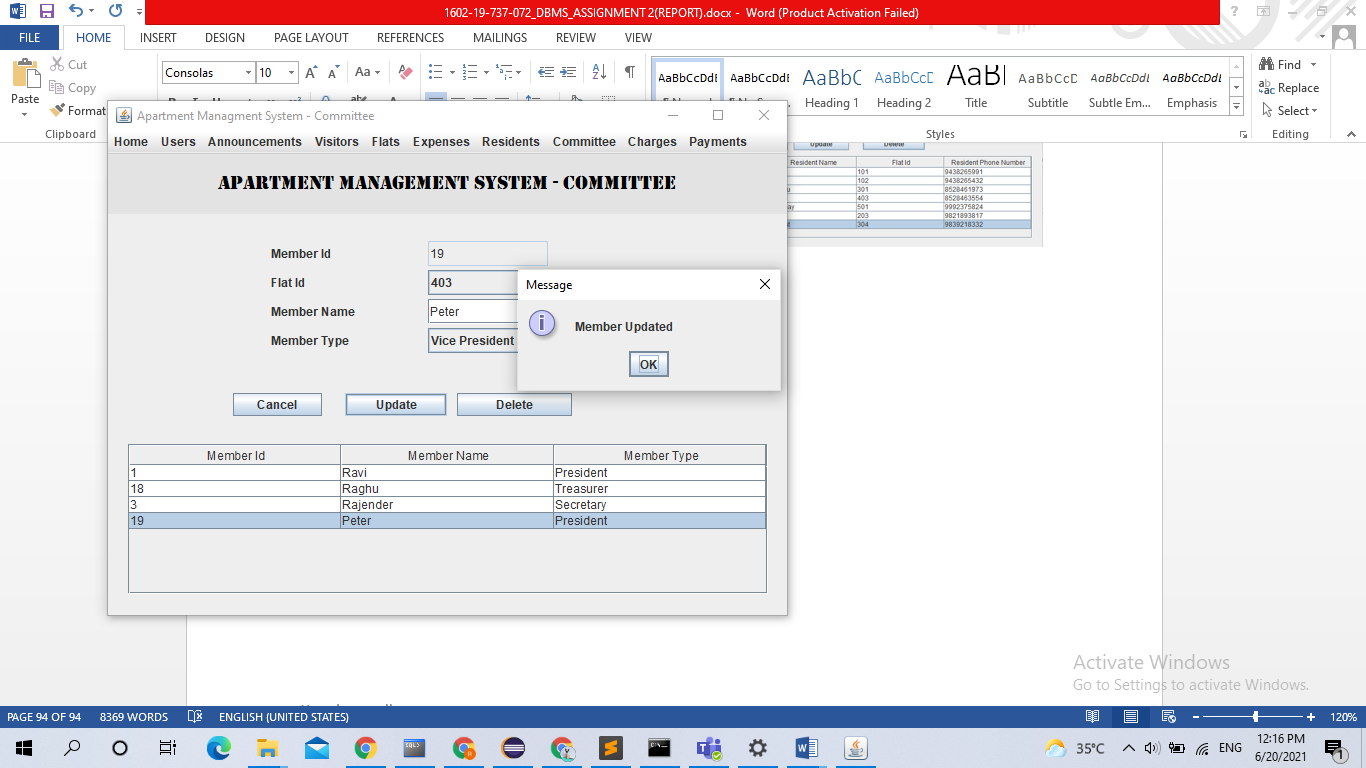
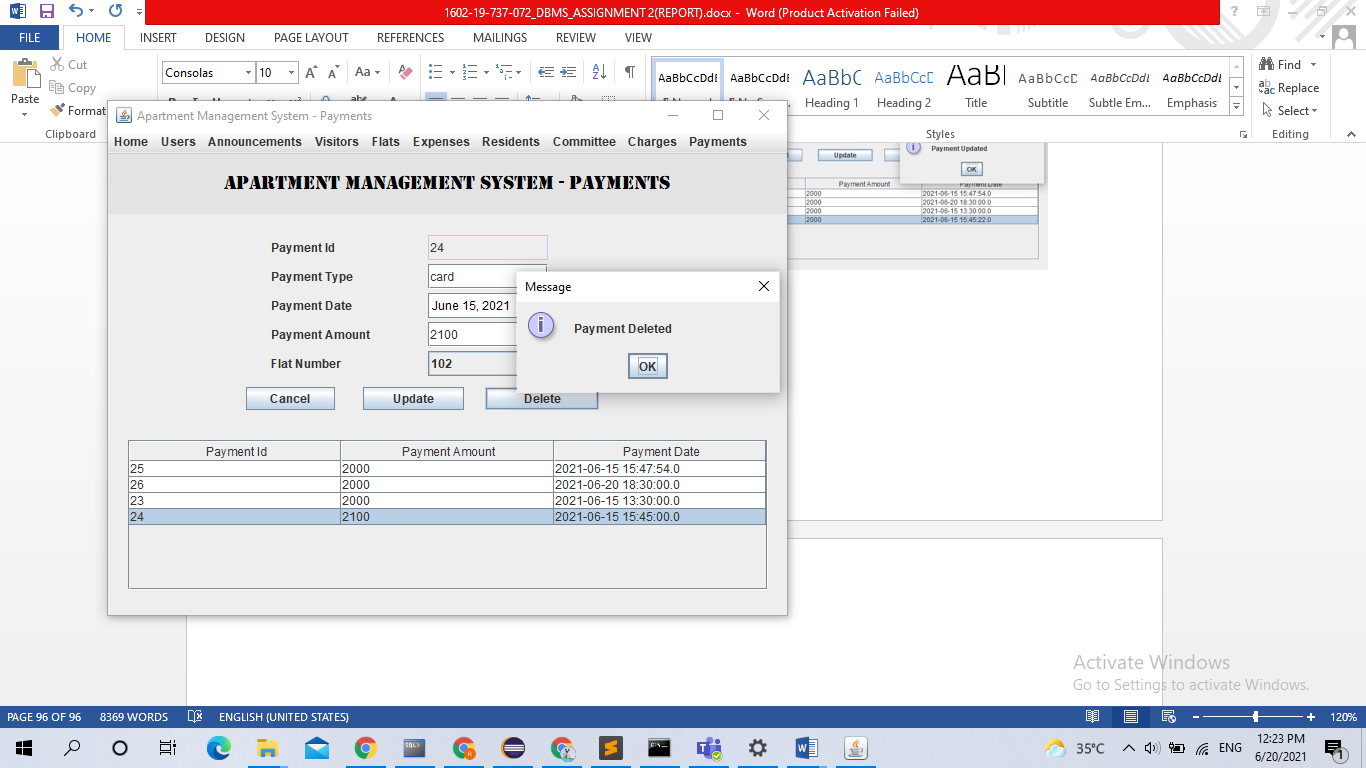
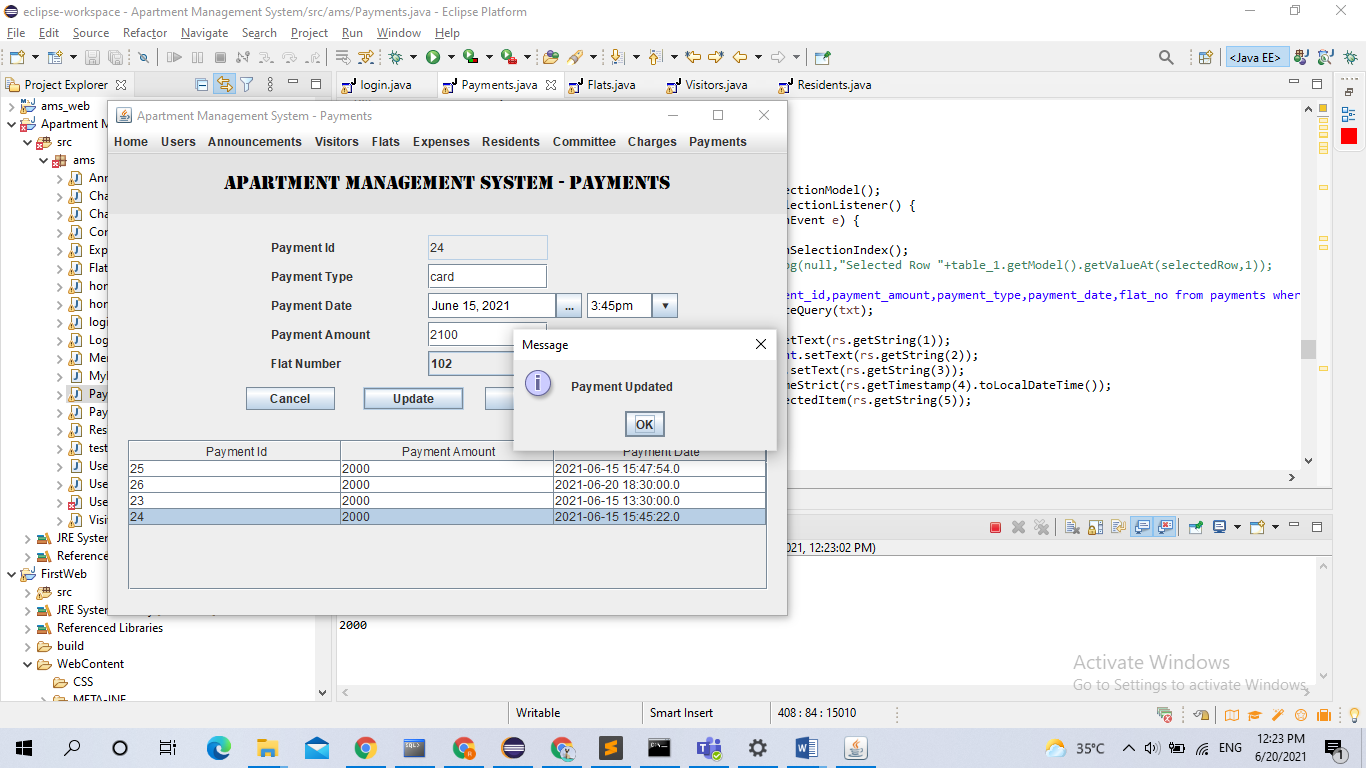






# TESTING

OUTPUT SCREENSHOTS:

## DISCUSSION AND FUTURE WORK

So far this project has helped us manage all the residents in an apartment and the expenses and charges for all the residents and also have an eye on who is entering and leaving the apartment.

In future we try to include different apartments and have a set of people to take care of more than one apartment.

# REFERENCES:

1. Abraham Silberschatz, Henry F Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw-Hill International Edition, 2010.

[https://docs.oracle.com/cd/E11882\_01/server.112/e40540/intro.htm#CNCPT001](https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm)