

//ITMD 510

//Java Final Project Spring 2020

//Done by Ahmed Ali Khan and Yazeed Alqahtani

//Date of Submission: 10th May 2020

```
package application;
```

```
import javafx.collections.FXCollections;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.application.Application;
```

```
import javafx.stage.Stage;
```

```
import javafx.scene.Scene;
```

```
import javafx.scene.layout.BorderPane;
```

```
import javafx.fxml.FXMLLoader;
```

```
import javafx.scene.layout.AnchorPane;
```

```
public class Main extends Application {
```

```
    public static Stage stage; // set global stage object!!!
```

```
    @Override
```

```
    public void start(Stage primaryStage) {
```

```
        try {
```

```
            stage = primaryStage;
```

```
            AnchorPane root = (AnchorPane)
```

```
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
```

```
            Scene scene = new Scene(root);
```

```
            scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
```

```
            stage.setTitle("Login View");
```

```
            stage.setScene(scene);
```

```
            stage.show();
```

```

        } catch (Exception e) {
            System.out.println("Error occurred while inflating view: " + e);
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}

/* JavaFX CSS - Leave this comment until you have at least create one rule which uses -fx-Property */
package controllers;

import java.io.IOException;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.swing.JOptionPane;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.ChoiceBox;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;

```

```
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import models.DaoModel;
```

```
public class AddCust
{
    @FXML
    public TextField NewCustId;

    @FXML
    private TextField NewCustName;

    @FXML
    private TextField NewCustUser;

    @FXML
    private TextField NewCustPass;

    @FXML
    private TextField NewCustPhone;

    @FXML
    private TextField NewCustEmail;

    @FXML
    private TextField NewCustCurrBalance;

    @FXML
    public javafx.scene.control.Button closebutton;
```

```

public void initData() throws SQLException, IOException
{
    // TODO Auto-generated method stub

    ResultSet rs = null;

    DaoModel db1 = new DaoModel();

    ArrayList<String> CustData = new ArrayList<String>();

    CustData.add(NewCustId.getText());
    CustData.add(NewCustName.getText());
    CustData.add(NewCustUser.getText());
    CustData.add(NewCustPass.getText());
    CustData.add(NewCustPhone.getText());
    CustData.add(NewCustEmail.getText());
    CustData.add(NewCustCurrBalance.getText());


    db1.InsertCustRec(CustData);

    System.out.println(CustData);


    Alert alert = new Alert(AlertType.INFORMATION);
    alert.setTitle("Success!");
    alert.setHeaderText(null);
    alert.setContentText("Customer succesfully added to the database!");
    alert.showAndWait();

    NewCustId.clear();
    NewCustName.clear();
    NewCustUser.clear();
    NewCustPass.clear();
    NewCustPhone.clear();
    NewCustEmail.clear();

```

```

        NewCustCurrBalance.clear();
    }

    @FXML

    public void closeButtonAction(){
        // get a handle to the stage
        Stage stage = (Stage) closebutton.getScene().getWindow();
        // do what you have to do
        stage.close();
    }
}

package controllers;

import java.io.IOException;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.swing.JOptionPane;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.ChoiceBox;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;

```

```
import javafx.scene.control.Alert.AlertType;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import models.DaoModel;


public class AddEmp
{
    @FXML
    public TextField NewEmpId;


    @FXML
    private TextField NewEmpName;


    @FXML
    private TextField NewEmpUser;


    @FXML
    private TextField NewEmpPass;


    @FXML
    private TextField NewEmpPhone;


    @FXML
    private TextField NewEmpEmail;


    @FXML
    public javafx.scene.control.Button closebutton;


    public void initData() throws SQLException, IOException
    {
        // TODO Auto-generated method stub
    }
}
```

```

        ResultSet rs = null;

        DaoModel db1 = new DaoModel();

        ArrayList<String> EmpData = new ArrayList<String>();

        EmpData.add(NewEmpId.getText());

        EmpData.add(NewEmpName.getText());

        EmpData.add(NewEmpUser.getText());

        EmpData.add(NewEmpPass.getText());

        EmpData.add(NewEmpPhone.getText());

        EmpData.add(NewEmpEmail.getText());


        db1.InsertEmpRec(EmpData);

        System.out.println(EmpData);


        Alert alert = new Alert(AlertType.INFORMATION);
        alert.setTitle("Success!");
        alert.setHeaderText(null);
        alert.setContentText("Employee succesfully added to the database!");
        alert.showAndWait();

        NewEmpId.clear();

        NewEmpName.clear();

        NewEmpUser.clear();

        NewEmpPass.clear();

        NewEmpPhone.clear();

        NewEmpEmail.clear();

    }

```

@FXML

```

public void closeButtonAction(){

    // get a handle to the stage

    Stage stage = (Stage) closebutton.getScene().getWindow();

    // do what you have to do

```

```

        stage.close();
    }
}

package controllers;

import java.io.IOException;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.swing.JOptionPane;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.ChoiceBox;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import models.DaoModel;

public class AddRoom
{
    @FXML

```



```
public TextField NewRoomId;
```

```
@FXML
```

```
private TextField NewRoomType;
```

```
@FXML
```

```
private TextField NewRoomsAvailable;
```

```
@FXML
```

```
private TextField NewRoomPrice;
```

```
@FXML
```

```
private TextField NewDining;
```

```
@FXML
```

```
private TextField NewGymTiming;
```

```
@FXML
```

```
    public javafx.scene.control.Button closebutton;
```

```
public void initData() throws SQLException, IOException
```

```
{
```

```
    // TODO Auto-generated method stub
```

```
    ResultSet rs = null;
```

```
    DaoModel db1 = new DaoModel();
```

```
    ArrayList<String> RoomData = new ArrayList<String>();
```

```
    RoomData.add(NewRoomId.getText());
```

```
    RoomData.add(NewRoomType.getText());
```

```
    RoomData.add(NewRoomsAvailable.getText());
```

```
    RoomData.add(NewRoomPrice.getText());
```

```
    RoomData.add(NewDining.getText());
```

```

RoomData.add(NewGymTiming.getText());

db1.InsertRoomRec(RoomData);

System.out.println(RoomData);


Alert alert = new Alert(AlertType.INFORMATION);
alert.setTitle("Success!");
alert.setHeaderText(null);
alert.setContentText("Room succesfully added to the database!");
alert.showAndWait();

NewRoomId.clear();
NewRoomType.clear();
NewRoomIsAvailable.clear();
NewRoomPrice.clear();
NewDining.clear();
NewGymTiming.clear();
}

```

@FXML

```

    public void closeButtonAction(){
        // get a handle to the stage
        Stage stage = (Stage) closebutton.getScene().getWindow();
        // do what you have to do
        stage.close();
    }
}

```

```

package controllers;

import models.DBConnect;

import java.io.IOException;

import java.sql.SQLException;

```

```
import javax.swing.JOptionPane;

import javafx.application.Application;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.MenuItem;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;

import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;

public class Admin extends Application implements AdminAbstract{

    public static String query;

    public int counter=0;

    @FXML
    public javafx.scene.control.Button closebutton;

    @Override
    public void start(Stage primaryStage)
    {
        if (counter==1)
        {
```

```

        try
        {
            System.out.println("Inside Admin");
        }
        catch (Exception e)
        {
            System.out.println("Error occured while inflating view: " + e);
        }
    }
}

```

```

    public void intialize1() throws IOException{
        FXMLLoader loader = new
FXMLLoader(getClass().getResource("/views/AdminView.fxml"));
        Parent root = loader.load();
        //Show scene 2 in new window
        application.Main.stage.setScene(new Scene(root));
        application.Main.stage.setTitle("Administrator Page");
        application.Main.stage.show();
    }

```

```

    public void GoAdminViewRoom() throws IOException, SQLException{
        counter=1;
        System.out.println(counter);
        //Stage stage = primaryStage;
        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/Display.fxml"));
        Scene scene1 = new Scene(root);
        application.Main.stage.setTitle("Display of records");
        application.Main.stage.setScene(scene1);
        application.Main.stage.show();
    }

```

```
public void GoAdminViewEmp() throws IOException, SQLException{  
    AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/DisplayEmpRec.fxml"));  
    Scene scene1 = new Scene(root);  
    application.Main.stage.setTitle("Display of records");  
    application.Main.stage.setScene(scene1);  
    application.Main.stage.show();  
}
```

```
public void GoAdminViewCust() throws IOException, SQLException{  
    AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/DisplayCustRec.fxml"));  
    Scene scene1 = new Scene(root);  
    application.Main.stage.setTitle("Display of records");  
    application.Main.stage.setScene(scene1);  
    application.Main.stage.show();  
}
```

```
public void GoAdminViewBooked() throws IOException, SQLException{  
    AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/DisplayBookedRec.fxml"));  
    Scene scene1 = new Scene(root);  
    application.Main.stage.setTitle("Display of records");  
    application.Main.stage.setScene(scene1);  
    application.Main.stage.show();  
}
```

```
public void GoAdminAddRoom() throws IOException, SQLException{  
    AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/AddRoom.fxml"));  
    Scene scene1 = new Scene(root);
```

```
        application.Main.stage.setTitle("Adding records for room");  
        application.Main.stage.setScene(scene1);  
        application.Main.stage.show();  
    }
```

```
    public void GoAdminAddEmp() throws IOException, SQLException{  
        AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/AddEmp.fxml"));  
        Scene scene1 = new Scene(root);  
        application.Main.stage.setTitle("Adding records for employees");  
        application.Main.stage.setScene(scene1);  
        application.Main.stage.show();  
    }
```

```
    public void GoAdminAddCust() throws IOException, SQLException{  
        AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/AddCust.fxml"));  
        Scene scene1 = new Scene(root);  
        application.Main.stage.setTitle("Adding records for customers");  
        application.Main.stage.setScene(scene1);  
        application.Main.stage.show();  
    }
```

```
    public void GoAdminDelRoom() throws IOException, SQLException{  
        AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/DelRoom.fxml"));  
        Scene scene1 = new Scene(root);  
        application.Main.stage.setTitle("Deleting record from rooms");  
        application.Main.stage.setScene(scene1);  
        application.Main.stage.show();  
    }
```

```
public void GoAdminDelEmp() throws IOException, SQLException{  
    AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/DelEmp.fxml"));  
    Scene scene1 = new Scene(root);  
    application.Main.stage.setTitle("Deleting record from employees");  
    application.Main.stage.setScene(scene1);  
    application.Main.stage.show();  
}
```

```
public void GoAdminDelCust() throws IOException, SQLException{  
    AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/DelCust.fxml"));  
    Scene scene1 = new Scene(root);  
    application.Main.stage.setTitle("Deleting record from customers");  
    application.Main.stage.setScene(scene1);  
    application.Main.stage.show();  
}
```

```
public void GoAdminUpdateRoom() throws IOException, SQLException{  
    AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/UpdateRoom.fxml"));  
    Scene scene1 = new Scene(root);  
    application.Main.stage.setTitle("Updating record in rooms");  
    application.Main.stage.setScene(scene1);  
    application.Main.stage.show();  
}
```

```
public void GoAdminUpdateEmp() throws IOException, SQLException{  
    AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/UpdateEmp.fxml"));  
    Scene scene1 = new Scene(root);  
    application.Main.stage.setTitle("Updating record in employees");
```

```

        application.Main.stage.setScene(scene1);
        application.Main.stage.show();
    }

    public void GoAdminUpdateCust() throws IOException, SQLException{
        AnchorPane root = (AnchorPane)
FXMMLoader.load(getClass().getResource("/views/UpdateCust.fxml"));
        Scene scene1 = new Scene(root);
        application.Main.stage.setTitle("Updating record in customers");
        application.Main.stage.setScene(scene1);
        application.Main.stage.show();
    }

    @FXML
    public void closeButtonAction(){
        // get a handle to the stage
        Stage stage = (Stage) closebutton.getScene().getWindow();
        // do what you have to do
        stage.close();
    }
}

package controllers;
import java.io.IOException;
import java.sql.SQLException;
import javafx.stage.Stage;

public interface AdminAbstract {
    void start(Stage primaryStage);
    void intialize1() throws IOException;
}

package controllers;
```



```
import models.DBConnect;
import models.DaoModel;
import models.DisplayModel;

import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.swing.JOptionPane;

import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.control.Button;
import javafx.scene.control.MenuItem;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;

import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
```

```

public class Customer extends Application {

    public static String query;

    @FXML
    private TextField textArea;

    @FXML
    public javafx.scene.control.Button closebutton;

    ObservableList<DisplayModel> data =
        FXCollections.observableArrayList();

    @FXML
    private TableView<DisplayModel> tableView;

    @FXML
    private AnchorPane anchortable;

    @FXML
    private TableColumn<DisplayModel, String> col_roomid;

    @FXML
    private TableColumn<DisplayModel, String> col_roomtype;

    @FXML
    private TableColumn<DisplayModel, String> col_roomisavailable;

```

@FXML

private TableColumn<DisplayModel, String> col_roomprice;

@FXML

private TableColumn<DisplayModel, String> col_dining;

@FXML

private TableColumn<DisplayModel, String> col_gymtiming;

@Override

public void start(Stage primaryStage)

{

try

{

System.out.println("Inside Customer");

}

catch (Exception e)

{

System.out.println("Error occurred while inflating view: " + e);

}

}

public void initialize1() throws IOException{

FXMLLoader loader = new

FXMLLoader(getClass().getResource("/views/CustomerView.fxml"));

Parent root = loader.load();

//Show scene 2 in new window

application.Main.stage.setScene(new Scene(root));

application.Main.stage.setTitle("Customer Page");

application.Main.stage.show();

```
}
```

```
public void GoCustomerViewRoom() throws IOException, SQLException{
```

```
    //Stage stage = primaryStage;
```

```
    AnchorPane root = (AnchorPane)
```

```
FXMLLoader.load(getClass().getResource("/views/Display.fxml"));
```

```
    Scene scene1 = new Scene(root);
```

```
    application.Main.stage.setTitle("Display of records");
```

```
    application.Main.stage.setScene(scene1);
```

```
    application.Main.stage.show();
```

```
}
```

```
public void searchButtonClicked() throws IOException, SQLException{
```

```
    // TODO Auto-generated method stub
```

```
        col_roomid.setCellValueFactory(new  
Property<ValueFactory>("roomid"));
```

```
        col_roomtype.setCellValueFactory(new  
Property<ValueFactory>("roomtype"));
```

```
        col_roomisavailable.setCellValueFactory(new  
Property<ValueFactory>("roomisavailable"));
```

```
        col_roomprice.setCellValueFactory(new  
Property<ValueFactory>("roomprice"));
```

```
        col_dining.setCellValueFactory(new  
Property<ValueFactory>("dining"));
```

```
        col_gymtiming.setCellValueFactory(new  
Property<ValueFactory>("gymtiming"));
```

```
String SearchRoomId=textField.getText();
```

```
ResultSet rs = null;
```

```
DaoModel db1 = new DaoModel();
```

```
rs = db1.ShowSearchRoomRecords(SearchRoomId);
```

```

        try {
            while (rs.next()){
                data.add(new
DisplayModel(rs.getString("roomid"),rs.getString("roomtype"),

rs.getString("roomisavailable"),rs.getString("roomprice"),rs.getString("dining"),

rs.getString("gymtiming")));
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        tableView.getItems().setAll(data);

        System.out.println(col_roomisavailable.getCellData(0));

    }

    public void bookButtonClicked() throws IOException, SQLException{
        System.out.println(col_roomisavailable.getCellData(0));
        String choice=col_roomisavailable.getCellData(0).toString();

        if(choice.equals("YES")) {
            Alert alert = new Alert(AlertType.INFORMATION);
            alert.setTitle("Success!");
            alert.setHeaderText(null);
            alert.setContentText("Room has been booked in the database!");
            alert.showAndWait();
        }
        else {
            Alert alert = new Alert(AlertType.ERROR);

```

```

        alert.setTitle("Error!");
        alert.setHeaderText(null);
        alert.setContentText("Room could not be booked in the database!");
        alert.showAndWait();
    }
}

@FXML
public void closeButtonAction(){
    // get a handle to the stage
    Stage stage = (Stage) closebutton.getScene().getWindow();
    // do what you have to do
    stage.close();
}

}

package controllers;


import java.io.IOException;
import java.net.URL;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.ResourceBundle;


import javax.swing.JOptionPane;


import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;

```

```
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.ComboBox;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import javafx.util.StringConverter;

import models.DaoModel;
import models.DisplayModel;

public class DelCust implements Initializable
{
    @FXML
    public TextField DelCustId;

    @FXML
    public javafx.scene.control.Button closebutton;

    public void initialize(URL location, ResourceBundle resources) {

        application.Main.stage.setTitle("Deletion of Records");
        application.Main.stage.show();

    }
```

```

public void deleteButtonClicked() throws SQLException {

    DaoModel db1 = new DaoModel();

    System.out.println("Delete Button Clicked");

    ArrayList<String> CustData = new ArrayList<String>();

    String CustIdToBeDel=DelCustId.getText();

    CustData.add(DelCustId.getText());


    System.out.println(CustData + CustIdToBeDel);


    db1.DeleteCustRec(CustData, CustIdToBeDel);


    Alert alert = new Alert(AlertType.INFORMATION);
    alert.setTitle("Success!");
    alert.setHeaderText(null);
    alert.setContentText("Customer has been deleted from the database!");


    alert.showAndWait();

    DelCustId.clear();

}

@FXML

public void closeButtonAction(){

    // get a handle to the stage

    Stage stage = (Stage) closebutton.getScene().getWindow();

    // do what you have to do

    stage.close();

}

}

package controllers;

```



```
import java.io.IOException;
```

```
import java.net.URL;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import java.util.ArrayList;
```

```
import java.util.ResourceBundle;
```

```
import javax.swing.JOptionPane;
```

```
import javafx.collections.FXCollections;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.fxml.FXML;
```

```
import javafx.fxml.FXMLLoader;
```

```
import javafx.fxml.Initializable;
```

```
import javafx.scene.Parent;
```

```
import javafx.scene.Scene;
```

```
import javafx.scene.control.Alert;
```

```
import javafx.scene.control.Alert.AlertType;
```

```
import javafx.scene.control.Button;
```

```
import javafx.scene.control.ComboBox;
```

```
import javafx.scene.control.RadioButton;
```

```
import javafx.scene.control.TextField;
```

```
import javafx.scene.layout.AnchorPane;
```

```
import javafx.stage.Stage;
```

```
import javafx.util.StringConverter;
```

```
import models.DaoModel;
```

```
import models.DisplayModel;
```

```
public class DelEmp implements Initializable
```

```

{

    @FXML

    public TextField DelEmpId;


    @FXML

    public javafx.scene.control.Button closebutton;


    public void initialize(URL location,ResourceBundle resources) {


        application.Main.stage.setTitle("Deletion of Records");
        application.Main.stage.show();

    }


    public void deleteButtonClicked() throws SQLException {


        DaoModel db1 = new DaoModel();
        System.out.println("Delete Button Clicked");
        ArrayList<String> EmpData = new ArrayList<String>();
        String EmpIdToBeDel=DelEmpId.getText();
        EmpData.add(DelEmpId.getText());


        System.out.println(EmpData + EmpIdToBeDel);


        db1.DeleteEmpRec(EmpData, EmpIdToBeDel);


        Alert alert = new Alert(AlertType.INFORMATION);
        alert.setTitle("Success!");
        alert.setHeaderText(null);
        alert.setContentText("Employee has been deleted from the database!");
    }
}

```

```

        alert.showAndWait();

        DelEmpId.clear();

    }

    @FXML

    public void closeButtonAction(){

        // get a handle to the stage

        Stage stage = (Stage) closebutton.getScene().getWindow();

        // do what you have to do

        stage.close();

    }

}

package controllers;


import java.io.IOException;

import java.net.URL;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.ResourceBundle;


import javax.swing.JOptionPane;


import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

```

```

import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.ComboBox;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import javafx.util.StringConverter;

import models.DaoModel;
import models.DisplayModel;

public class DelRoom implements Initializable
{
    @FXML
    public TextField DelRoomId;

    @FXML
    public javafx.scene.control.Button closebutton;

    public void initialize(URL location,ResourceBundle resources) {

        application.Main.stage.setTitle("Deletion of Records");
        application.Main.stage.show();

    }

    public void deleteButtonClicked() throws SQLException {

        DaoModel db1 = new DaoModel();

```

```

        System.out.println("Delete Button Clicked");

        ArrayList<String> RoomData = new ArrayList<String>();

        String RoomIdToBeDel=DelRoomId.getText();

        RoomData.add(DelRoomId.getText());


        System.out.println(RoomData + RoomIdToBeDel);


        db1.DeleteRoomRec(RoomData, RoomIdToBeDel);


        Alert alert = new Alert(AlertType.INFORMATION);
        alert.setTitle("Success!");
        alert.setHeaderText(null);
        alert.setContentText("Room has been deleted from the database!");


        alert.showAndWait();

        DelRoomId.clear();

    }

    @FXML

    public void closeButtonAction(){

        // get a handle to the stage

        Stage stage = (Stage) closebutton.getScene().getWindow();

        // do what you have to do

        stage.close();

    }

}

package controllers;


import java.io.IOException;

import java.net.URL;

```

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ResourceBundle;
```

```
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.AnchorPane;
import models.DBConnect;
import models.DisplayModel;
import models.DaoModel;
```

```
import controllers.Admin;
```

```
public class Display extends Admin implements Initializable{
    DBConnect conn = null;
    Statement stmt = null;

    @FXML
    private TableView<DisplayModel> tableView;
```

@FXML

private AnchorPane anchortable;

@FXML

private TableColumn<DisplayModel, String> col_roomid;

@FXML

private TableColumn<DisplayModel, String> col_roomtype;

@FXML

private TableColumn<DisplayModel, String> col_roomisavailable;

@FXML

private TableColumn<DisplayModel, String> col_roomprice;

@FXML

private TableColumn<DisplayModel, String> col_dining;

@FXML

private TableColumn<DisplayModel, String> col_gymtiming;

Parent root = null;

ObservableList<DisplayModel> data =

FXCollections.observableArrayList();

FXMLLoader loader = new FXMLLoader(getClass().getResource("/views/Display.fxml"));

public void initialize2() throws IOException {

```
application.Main.stage.setScene(new Scene(loader.load()));  
System.out.println("Inside initialize2");
```

```
}
```

```
public void initialize(URL location,ResourceBundle resources)
```

```
{
```

```
    initCol();
```

```
    DBConnect handler = new DBConnect();
```

```
        try
```

```
        {
```

```
            loadData();
```

```
        } catch (SQLException e)
```

```
        {
```

```
            // TODO Auto-generated catch block
```

```
            e.printStackTrace();
```

```
        }
```

```
        application.Main.stage.setTitle("Display Page");
```

```
        application.Main.stage.show();
```

```
        System.out.println("Inside Initialize");
```

```
    }
```

```
private void loadData() throws SQLException
```

```
{
```

```
    // TODO Auto-generated method stub
```

```
    ResultSet rs = null;
```

```
    DaoModel db1 = new DaoModel();
```

```
    rs = db1.ShowAllRoomRecords();
```



```

        try
        {
            while(rs.next())
            {

                data.add(new
DisplayModel(rs.getString("roomid"),rs.getString("roomtype"),

                rs.getString("roomisavailable"),rs.getString("roomprice"),rs.getString("dining"),

                rs.getString("gymtiming")));

            }
        }catch (SQLException e)
        {

            e.printStackTrace();

        }
        tableView.getItems().setAll(data);
    }

    private void initCol()
    {

        // TODO Auto-generated method stub
        col_roomid.setCellValueFactory(new PropertyValueFactory<>("roomid"));
        col_roomtype.setCellValueFactory(new PropertyValueFactory<>("roomtype"));
        col_roomisavailable.setCellValueFactory(new
PropertyValueFactory<>("roomisavailable"));
        col_roomprice.setCellValueFactory(new PropertyValueFactory<>("roomprice"));
        col_dining.setCellValueFactory(new PropertyValueFactory<>("dining"));
        col_gymtiming.setCellValueFactory(new PropertyValueFactory<>("gymtiming"));

    }

}

```

```
package controllers;

import java.io.IOException;
import java.net.URL;


import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ResourceBundle;


import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.AnchorPane;
import models.DBConnect;
import models.DisplayModel;
import models.DisplayBookedRecModel;
import models.DaoModel;


import controllers.Admin;


public class DisplayBookedRec extends Admin implements Initializable{
    DBConnect conn = null;
```

```
Statement stmt = null;
```

```
@FXML
```

```
private TableView<DisplayBookedRecModel> tableView;
```

```
@FXML
```

```
private AnchorPane anchortable;
```

```
@FXML
```

```
private TableColumn<DisplayBookedRecModel, String> col_roomid;
```

```
@FXML
```

```
private TableColumn<DisplayBookedRecModel, String> col_custid;
```

```
Parent root = null;
```

```
ObservableList<DisplayBookedRecModel> data =
```

```
    FXCollections.observableArrayList();
```

```
FXMLLoader loader = new FXMLLoader(getClass().getResource("/views/DisplayBookedRec.fxml"));
```

```
public void initialize2() throws IOException {
```

```
    application.Main.stage.setScene(new Scene(loader.load()));
```

```
    System.out.println("Inside initialize2");
```

```
}
```

```
public void initialize(URL location, ResourceBundle resources)
```

```

{

    initCol();

    DBConnect handler = new DBConnect();


    try
    {
        loadData();
    } catch (SQLException e)
    {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }


    application.Main.stage.setTitle("Display Page");
    application.Main.stage.show();
    System.out.println("Inside Initialize");

}

```

```

private void loadData() throws SQLException
{
    // TODO Auto-generated method stub

    ResultSet rs = null;

    DaoModel db1 = new DaoModel();
    rs = db1.ShowAllBookedRecords();


    try
    {
        while(rs.next())
        {

```

```

                                data.add(new
DisplayBookedRecModel(rs.getString("roomid"),rs.getString("custid")));
                                }
                                }catch (SQLException e)
                                {
                                    e.printStackTrace();
                                }
                                tableView.getItems().setAll(data);
                                }

```

```

private void initCol()

```

```

{

```

```

    // TODO Auto-generated method stub

```

```

    col_roomid.setCellValueFactory(new PropertyValueFactory<>("roomid"));

```

```

    col_custid.setCellValueFactory(new PropertyValueFactory<>("custid"));

```

```

    }

```

```

}

```

```

package controllers;

```

```

import java.io.IOException;

```

```

import java.net.URL;

```

```

import java.sql.ResultSet;

```

```

import java.sql.SQLException;

```

```

import java.sql.Statement;

```

```

import java.util.ResourceBundle;

```

```

import javafx.collections.FXCollections;

```

```

import javafx.collections.ObservableList;

```

```

import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.AnchorPane;
import models.DBConnect;
import models.DisplayModel;
import models.DisplayCustRecModel;
import models.DaoModel;

import controllers.Admin;

public class DisplayCustRec extends Admin implements Initializable{

    DBConnect conn = null;
    Statement stmt = null;

    @FXML

    private TableView<DisplayCustRecModel> tableView;

    @FXML

    private AnchorPane anchortable;

    @FXML

    private TableColumn<DisplayCustRecModel, String> col_custid;

    @FXML

```

```
private TableColumn<DisplayCustRecModel, String> col_custname;
```

```
@FXML
```

```
private TableColumn<DisplayCustRecModel, String> col_custuser;
```

```
@FXML
```

```
private TableColumn<DisplayCustRecModel, String> col_custpass;
```

```
@FXML
```

```
private TableColumn<DisplayCustRecModel, String> col_custphone;
```

```
@FXML
```

```
private TableColumn<DisplayCustRecModel, String> col_custemail;
```

```
@FXML
```

```
private TableColumn<DisplayCustRecModel, String> col_custcurrbalance;
```

```
Parent root = null;
```

```
ObservableList<DisplayCustRecModel> data =
```

```
    FXCollections.observableArrayList();
```

```
FXMLLoader loader = new FXMLLoader(getClass().getResource("/views/DisplayCustRec.fxml"));
```

```
public void initialize2() throws IOException {
```

```
    application.Main.stage.setScene(new Scene(loader.load()));
```

```
    System.out.println("Inside initialize2");
```

```
}
```

```

public void initialize(URL location,ResourceBundle resources)
{
    initCol();
    DBConnect handler = new DBConnect();

    try
    {
        loadData();
    } catch (SQLException e)
    {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    application.Main.stage.setTitle("Display Page");
    application.Main.stage.show();
    System.out.println("Inside Initialize");

}

```

```

private void loadData() throws SQLException
{
    // TODO Auto-generated method stub
    ResultSet rs = null;
    DaoModel db1 = new DaoModel();
    rs = db1.ShowAllCustomerRecords();

    try
    {
        while(rs.next())
        {

```



```

        data.add(new
DisplayCustRecModel(rs.getString("custid"),rs.getString("custname"),

        rs.getString("custuser"),rs.getString("custpass"),rs.getString("custphone"),

        rs.getString("custemail"),rs.getString("custcurrbalance")));

    }

    }catch (SQLException e)

    {

        e.printStackTrace();

    }

    tableView.getItems().setAll(data);

}

private void initCol()

{

    // TODO Auto-generated method stub

    col_custid.setCellValueFactory(new PropertyValueFactory<>("custid"));

    col_custname.setCellValueFactory(new PropertyValueFactory<>("custname"));

    col_custuser.setCellValueFactory(new PropertyValueFactory<>("custuser"));

    col_custpass.setCellValueFactory(new PropertyValueFactory<>("custpass"));

    col_custphone.setCellValueFactory(new PropertyValueFactory<>("custphone"));

    col_custemail.setCellValueFactory(new PropertyValueFactory<>("custemail"));

    col_custcurrbalance.setCellValueFactory(new
PropertyValueFactory<>("custcurrbalance"));

    }

}

package controllers;

import java.io.IOException;

import java.net.URL;

```

```
import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;
```

```
import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.AnchorPane;

import models.DBConnect;

import models.DisplayModel;

import models.DisplayEmpRecModel;

import models.DaoModel;
```

```
import controllers.Admin;
```

```
public class DisplayEmpRec extends Admin implements Initializable{

    DBConnect conn = null;

    Statement stmt = null;

    @FXML
```

```
private TableView<DisplayEmpRecModel> tableView;
```

```
    @FXML
```

```
    private AnchorPane anchortable;
```

```
    @FXML
```

```
    private TableColumn<DisplayEmpRecModel, String> col_empid;
```

```
    @FXML
```

```
    private TableColumn<DisplayEmpRecModel, String> col_empname;
```

```
    @FXML
```

```
    private TableColumn<DisplayEmpRecModel, String> col_empuser;
```

```
    @FXML
```

```
    private TableColumn<DisplayEmpRecModel, String> col_emppass;
```

```
    @FXML
```

```
    private TableColumn<DisplayEmpRecModel, String> col_empphone;
```

```
    @FXML
```

```
    private TableColumn<DisplayEmpRecModel, String> col_empemail;
```

```
Parent root = null;
```

```
ObservableList<DisplayEmpRecModel> data =
```

```
    FXCollections.observableArrayList();
```

```
FXMLLoader loader = new FXMLLoader(getClass().getResource("/views/DisplayEmpRec.fxml"));
```

```

public void initialize2() throws IOException {

    application.Main.stage.setScene(new Scene(loader.load()));

    System.out.println("Inside Initialize2");

}

```

```

public void initialize(URL location,ResourceBundle resources)
{

    initCol();

    DBConnect handler = new DBConnect();

    try
    {

        loadData();

    } catch (SQLException e)
    {

        // TODO Auto-generated catch block

        e.printStackTrace();

    }

    application.Main.stage.setTitle("Display Page");

    application.Main.stage.show();

    System.out.println("Inside Initialize");

}

```

```

private void loadData() throws SQLException
{

    // TODO Auto-generated method stub

    ResultSet rs = null;

```

```

        DaoModel db1 = new DaoModel();
        rs = db1.ShowAllEmployeeRecords();

        try
        {
            while(rs.next())
            {

                data.add(new
DisplayEmpRecModel(rs.getString("empid"),rs.getString("empname"),

                rs.getString("empuser"),rs.getString("emppass"),rs.getString("empphone"),

                rs.getString("empemail")));

            }
        }catch (SQLException e)
        {

            e.printStackTrace();

        }
        tableView.getItems().setAll(data);
    }

    private void initCol()
    {

        // TODO Auto-generated method stub
        col_empid.setCellValueFactory(new PropertyValueFactory<>("empid"));
        col_empname.setCellValueFactory(new PropertyValueFactory<>("empname"));
        col_empuser.setCellValueFactory(new PropertyValueFactory<>("empuser"));
        col_emppass.setCellValueFactory(new PropertyValueFactory<>("emppass"));
        col_empphone.setCellValueFactory(new PropertyValueFactory<>("empphone"));
        col_empemail.setCellValueFactory(new PropertyValueFactory<>("empemail"));

    }

```

```
}  
  
package controllers;  
  
import models.DBConnect;  
import java.io.IOException;  
import java.sql.SQLException;  
  
import javax.swing.JOptionPane;  
  
import javafx.application.Application;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Alert;  
import javafx.scene.control.Alert.AlertType;  
import javafx.scene.control.Button;  
import javafx.scene.control.MenuItem;  
import javafx.scene.control.RadioButton;  
import javafx.scene.control.TextField;  
  
import javafx.scene.layout.AnchorPane;  
import javafx.stage.Stage;  
  
public class Employee extends Application {  
  
    public static String query;  
  
    @FXML  
    public javafx.scene.control.Button closebutton;
```

```

@Override

public void start(Stage primaryStage)
{
    try
    {
        System.out.println("Inside Employee");
    }
    catch (Exception e)
    {
        System.out.println("Error occured while inflating view: " + e);
    }
}

public void initialize1() throws IOException{
    FXMLLoader loader = new
FXMLLoader(getClass().getResource("/views/EmployeeView.fxml"));

    Parent root = loader.load();

    //Show scene 2 in new window
    application.Main.stage.setScene(new Scene(root));
    application.Main.stage.setTitle("Employee Page");
    application.Main.stage.show();
}

public void GoEmployeeViewRoom() throws IOException, SQLException{

    //Stage stage = primaryStage;

    AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/Display.fxml"));

    Scene scene1 = new Scene(root);

    application.Main.stage.setTitle("Display of records");

    application.Main.stage.setScene(scene1);

    application.Main.stage.show();
}

```

```
}
```

```
    public void GoEmployeeViewCust() throws IOException, SQLException{  
        AnchorPane root = (AnchorPane)  
FXMLLoader.load(getClass().getResource("/views/DisplayCustRec.fxml"));  
        Scene scene1 = new Scene(root);  
        application.Main.stage.setTitle("Display of records");  
        application.Main.stage.setScene(scene1);  
        application.Main.stage.show();  
    }
```

```
@FXML
```

```
    public void closeButtonAction(){  
        // get a handle to the stage  
        Stage stage = (Stage) closebutton.getScene().getWindow();  
        // do what you have to do  
        stage.close();  
    }  
}
```

```
package controllers;  
import models.DBConnect;  
import java.awt.Button;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.fxml.FXML;  
import java.io.IOException;  
import javafx.scene.control.*;  
import javafx.scene.control.Alert.AlertType;  
import javafx.stage.Stage;  
import java.util.ResourceBundle;  
import models.DaoModel;
```



```

public class LoginAccount {

    public String usernamevar;

    public String passwordvar;


    ObservableList<String> UserChoiceList =
FXCollections.observableArrayList("Administrator", "Employee", "Customer");


    @SuppressWarnings("rawtypes")

    @FXML

    private ChoiceBox UserChoice;


    @FXML

    protected ResourceBundle resources;


    @FXML

    private TextField usernamefield;


    @FXML

    private PasswordField passwordfield;


    @FXML

    public javafx.scene.control.Button closebutton;


    @FXML

    public void closeButtonAction(){

        // get a handle to the stage

        Stage stage = (Stage) closebutton.getScene().getWindow();

        // do what you have to do

        stage.close();
    }
}

```

```
}
```

```
@SuppressWarnings("unchecked")
```

```
@FXML
```

```
public void clearFields() {  
    usernamefield.clear();  
    passwordfield.clear();  
    UserChoice.setValue(null);
```

```
}
```

```
@SuppressWarnings("unchecked")
```

```
private void Sceneload() {  
    UserChoice.setItems(UserChoiceList);
```

```
}
```

```
public void initialize(){
```

```
    if(UserChoice!=null) {  
        Sceneload();}
```

```
}
```

```
public void LoginInitiate() throws IOException {
```

```
    String[] record = null;  
    System.out.println("In Login Initiate");  
    DaoModel dr = new DaoModel();  
    String user =usernamefield.getText();  
    System.out.println(user);  
    if(UserChoice.getValue()==null){  
        Alert alert = new Alert(AlertType.ERROR);  
        alert.setTitle("Error Dialog");
```

```

        alert.setHeaderText("Error!");

        alert.setContentText("Select the user type!");

        alert.showAndWait();

    }

    else if(UserChoice.getValue().equals("Administrator")) {

        record = dr.GoAdminView();

    }else if(UserChoice.getValue().equals("Customer")) {

        record = dr.GoCustView(user);

    }else if(UserChoice.getValue().equals("Employee")) {

        record = dr.GoEmpView(user);

    }

    if(!(UserChoice.getValue()==null)) {

        if(UserChoice.getValue().equals("Administrator") && user.equals(record[0]) &&
passwordfield.getText().equals(record[1])) {

            // for the operations

            Admin adm = new Admin();

            adm.intialize1();

        }

        else if(UserChoice.getValue().equals("Employee") && user.equals(record[0]) &&
passwordfield.getText().equals(record[1])) {

            // for the operations

            Employee emp = new Employee();

            emp.intialize1();

        }

        else if(UserChoice.getValue().equals("Customer") && user.equals(record[0]) &&
passwordfield.getText().equals(record[1])) {

            // for the operations

            Customer cust = new Customer();

            cust.intialize1();

        }
    }

```

```
        else {  
            Alert alert = new Alert(AlertType.ERROR);  
            alert.setTitle("Error Dialog");  
            alert.setHeaderText("Credentials Error!");  
            alert.setContentText("The username or password haven't been entered  
correctly. Please confirm and try again.");  
            alert.showAndWait();  
        }  
    }  
}
```

```
package controllers;
```

```
import java.io.IOException;
```

```
import java.net.URL;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import java.util.ArrayList;
```

```
import java.util.ResourceBundle;
```

```
import javax.swing.JOptionPane;
```

```
import javafx.collections.FXCollections;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.fxml.FXML;
```

```
import javafx.fxml.FXMLLoader;
```

```
import javafx.fxml.Initializable;
```

```
import javafx.scene.Parent;
```

```
import javafx.scene.Scene;
```

```
import javafx.scene.control.Alert;
```

```
import javafx.scene.control.Button;
import javafx.scene.control.ChoiceBox;
import javafx.scene.control.ComboBox;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import javafx.util.StringConverter;
```

```
import models.DaoModel;
import models.DisplayModel;
```

```
public class UpdateCust implements Initializable
{
```

```
    @FXML
```

```
    private TextField UpdateCustId;
```

```
    @FXML
```

```
    private TextField UpdateCustName;
```

```
    @FXML
```

```
    private TextField UpdateCustUser;
```

```
    @FXML
```

```
    private TextField UpdateCustPass;
```

```
    @FXML
```

```
    private TextField UpdateCustPhone;
```

```
    @FXML
```

```
private TextField UpdateCustEmail;
```

```
@FXML
```

```
private TextField UpdateCustCurrBalance;
```

```
@FXML
```

```
    public javafx.scene.control.Button closebutton;
```

```
public void initialize(URL location,ResourceBundle resources)
```

```
{
```

```
    application.Main.stage.setTitle("Update of Records");
```

```
    application.Main.stage.show();
```

```
    System.out.println("Inside initialize");
```

```
}
```

```
public void updateButtonClicked() throws SQLException, IOException
```

```
{
```

```
    DaoModel db1 = new DaoModel();
```

```
    System.out.println("Update Button Clicked");
```

```
    ArrayList<String> CustData = new ArrayList<String>();
```

```
    CustData.add(UpdateCustName.getText());
```

```
    CustData.add(UpdateCustUser.getText());
```

```
    CustData.add(UpdateCustPass.getText());
```

```
    CustData.add(UpdateCustPhone.getText());
```

```
    CustData.add(UpdateCustEmail.getText());
```

```
    CustData.add(UpdateCustCurrBalance.getText());
```

```
    System.out.println(CustData);
```

```
    String CustIdToBeUpdated=UpdateCustId.getText();
```

```
    db1.UpdateCustRec(CustData,CustIdToBeUpdated);
```

```

        Alert alert = new Alert(AlertType.INFORMATION);

        alert.setTitle("Success!");

        alert.setHeaderText(null);

        alert.setContentText("Customer has been updated in the database!");


        alert.showAndWait();

        UpdateCustId.clear();

        UpdateCustName.clear();

        UpdateCustUser.clear();

        UpdateCustPass.clear();

        UpdateCustPhone.clear();

        UpdateCustEmail.clear();

        UpdateCustCurrBalance.clear();
    }

    @FXML

    public void closeButtonAction(){

        // get a handle to the stage

        Stage stage = (Stage) closebutton.getScene().getWindow();

        // do what you have to do

        stage.close();

    }

}

package controllers;


import java.io.IOException;

import java.net.URL;

import java.sql.ResultSet;

import java.sql.SQLException;

```

```
import java.util.ArrayList;
import java.util.ResourceBundle;

import javax.swing.JOptionPane;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.ChoiceBox;
import javafx.scene.control.ComboBox;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import javafx.util.StringConverter;

import models.DaoModel;
import models.DisplayModel;

public class UpdateEmp implements Initializable
{
    @FXML
    private TextField UpdateEmpId;
```


@FXML

private TextField UpdateEmpName;

@FXML

private TextField UpdateEmpUser;

@FXML

private TextField UpdateEmpPass;

@FXML

private TextField UpdateEmpPhone;

@FXML

private TextField UpdateEmpEmail;

@FXML

public javafx.scene.control.Button closebutton;

public void initialize(URL location,ResourceBundle resources)

{

application.Main.stage.setTitle("Update of Records");

application.Main.stage.show();

System.out.println("Inside initialize");

}

public void updateButtonClicked() throws SQLException, IOException

{

DaoModel db1 = new DaoModel();

System.out.println("Update Button Clicked");

ArrayList<String> EmpData = new ArrayList<String>();

```
EmpData.add(UpdateEmpName.getText());
EmpData.add(UpdateEmpUser.getText());
EmpData.add(UpdateEmpPass.getText());
EmpData.add(UpdateEmpPhone.getText());
EmpData.add(UpdateEmpEmail.getText());
System.out.println(EmpData);
String EmpIdToBeUpdated=UpdateEmpId.getText();
db1.UpdateEmpRec(EmpData,EmpIdToBeUpdated);
```

```
Alert alert = new Alert(AlertType.INFORMATION);
alert.setTitle("Success!");
alert.setHeaderText(null);
alert.setContentText("Employee has been updated in the database!");
```

```
alert.showAndWait();
UpdateEmpId.clear();
UpdateEmpName.clear();
UpdateEmpUser.clear();
UpdateEmpPass.clear();
UpdateEmpPhone.clear();
UpdateEmpEmail.clear();
```

```
}
```

```
@FXML
```

```
public void closeButtonAction(){
    // get a handle to the stage
    Stage stage = (Stage) closebutton.getScene().getWindow();
    // do what you have to do
    stage.close();
}
```

```
}
```

```
package controllers;
```

```
import java.io.IOException;
```

```
import java.net.URL;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import java.util.ArrayList;
```

```
import java.util.ResourceBundle;
```

```
import javax.swing.JOptionPane;
```

```
import javafx.collections.FXCollections;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.fxml.FXML;
```

```
import javafx.fxml.FXMLLoader;
```

```
import javafx.fxml.Initializable;
```

```
import javafx.scene.Parent;
```

```
import javafx.scene.Scene;
```

```
import javafx.scene.control.Alert;
```

```
import javafx.scene.control.Button;
```

```
import javafx.scene.control.ChoiceBox;
```

```
import javafx.scene.control.ComboBox;
```

```
import javafx.scene.control.RadioButton;
```

```
import javafx.scene.control.TextField;
```

```
import javafx.scene.control.Alert.AlertType;
```

```
import javafx.scene.layout.AnchorPane;
```

```
import javafx.stage.Stage;
```

```
import javafx.util.StringConverter;
```

```
import models.DaoModel;
```

```

import models.DisplayModel;

public class UpdateRoom implements Initializable
{
    @FXML
    private TextField UpdateRoomId;

    @FXML
    private TextField UpdateRoomType;

    @FXML
    private TextField UpdateRoomIsAvailable;

    @FXML
    private TextField UpdateRoomPrice;

    @FXML
    private TextField UpdateDining;

    @FXML
    private TextField UpdateGymTiming;

    @FXML
    public javafx.scene.control.Button closebutton;

    public void initialize(URL location,ResourceBundle resources)
    {
        application.Main.stage.setTitle("Update of Records");
        application.Main.stage.show();
        System.out.println("Inside initialize");
    }
}

```

```

public void updateButtonClicked() throws SQLException, IOException
{

    DaoModel db1 = new DaoModel();

    System.out.println("Update Button Clicked");

    ArrayList<String> RoomData = new ArrayList<String>();

    RoomData.add(UpdateRoomType.getText());
    RoomData.add(UpdateRoomIsAvailable.getText());
    RoomData.add(UpdateRoomPrice.getText());
    RoomData.add(UpdateDining.getText());
    RoomData.add(UpdateGymTiming.getText());
    System.out.println(RoomData);
    String RoomIdToBeUpdated=UpdateRoomId.getText();
    db1.UpdateRoomRec(RoomData,RoomIdToBeUpdated);

    Alert alert = new Alert(AlertType.INFORMATION);
    alert.setTitle("Success!");
    alert.setHeaderText(null);
    alert.setContentText("Room has been updated in the database!");

    alert.showAndWait();
    UpdateRoomId.clear();
    UpdateRoomType.clear();
    UpdateRoomIsAvailable.clear();
    UpdateRoomPrice.clear();
    UpdateDining.clear();
    UpdateGymTiming.clear();

}

```

@FXML

```

        public void closeButtonAction(){

            // get a handle to the stage

            Stage stage = (Stage) closebutton.getScene().getWindow();

            // do what you have to do

            stage.close();

        }

    }

package models;

import java.sql.*;

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.util.ArrayList;


public class DaoModel { //extends Display

    DBConnect conn = null;

    Statement stmt = null;


    public DaoModel() {

        //create db object instance

        conn = new DBConnect();

    }


    //GoAdminView method initiates login procedure for admin

    public String[] GoAdminView() {

        ResultSet rs1 = null;

        String[] values = new String[2];

        try {

            //open a connection

            stmt = conn.connect().createStatement();

            System.out.println("Retrieving Records from the table...");

            String sql = "SELECT adminuser,adminpass from aak_padmin";

```

```

        System.out.println(sql);
        // execute the query
        rs1 = stmt.executeQuery(sql);
        rs1.next();
        System.out.println(rs1.getString("adminuser"));
        values[0] = rs1.getString("adminuser");
        values[1] = rs1.getString("adminpass");
        System.out.println("\n\nRetrieving Records successful");
        // close the connection
        conn.connect().close();
    }
    catch (SQLException se) {
        se.printStackTrace();
    }
    return values;
}

```

```

//GoCustView method initiates login procedure for customer
public String[] GoCustView(String CustIDInput) {
    ResultSet rs1 = null;
    String[] values = new String[2];
    try {
        //open a connection
        stmt = conn.connect().createStatement();
        System.out.println("Connected to the database.\n"+
                           "\nRetrieving Records from the table...");
        String sql = "SELECT custuser,custpass from aak_pcust WHERE custuser IN ('"+
CustIDInput + "')";
        System.out.println(sql);
        // execute the query

```

```

rs1 = stmt.executeQuery(sql);

while(rs1.next()) {
values[0] = rs1.getString("custuser");
values[1] = rs1.getString("custpass");}
System.out.println("\n\nRetrieving Records successful");
// close the connection
conn.connect().close();

}
catch (SQLException se) {
    se.printStackTrace();
}
return values;

}

```

```

//GoEmpView method initiates login procedure for employee
public String[] GoEmpView(String EmplIDInput) {
    ResultSet rs1 = null;
    String[] values = new String[2];
    try {
        //open a connection
        stmt = conn.connect().createStatement();
        System.out.println("Connected to the database.\n"+
                                "\nRetrieving Records from the table...");
        String sql = "SELECT empuser,emppass from aak_pemp WHERE empuser IN ('"+
EmplIDInput + "')";
        System.out.println(sql);

```



```

// execute the query
rs1 = stmt.executeQuery(sql);

while(rs1.next()) {
    values[0] = rs1.getString("empuser");
    values[1] = rs1.getString("emppass");
}

System.out.println("\n\nRetrieving Records successful");
// close the connection
conn.connect().close();

}

catch (SQLException se) {
    se.printStackTrace();
}

return values;
}

//ShowAllRoomRecords method is used for displaying all rooms
public ResultSet ShowAllRoomRecords() throws SQLException {
    // TODO Auto-generated method stub
    ResultSet rs = null;
    String sql = null;
    stmt = conn.connect().createStatement();
    sql = "SELECT * from aak_prooms";
    System.out.println("inside ShowAllRoomRecords");
    //query along quotes
    String sqlQuery = sql;
    // execute the query
    rs = stmt.executeQuery(sqlQuery);

```

```

        System.out.println("\n\nRetrieving Records successful");

        // close the connection
        conn.connect().close();

        return rs;
    }

    //ShowAllEmployeeRecords method is used for displaying all employees
    public ResultSet ShowAllEmployeeRecords() throws SQLException{
        // TODO Auto-generated method stub

        ResultSet rs = null;
        String sql = null;
        stmt = conn.connect().createStatement();
        sql = "SELECT * from aak_pemp";
        System.out.println("inside ShowAllEmployeeRecords");

        //query along quotes
        String sqlQuery = sql;
        // execute the query
        rs = stmt.executeQuery(sqlQuery);
        System.out.println("\n\nRetrieving Records successful");

        // close the connection
        conn.connect().close();

        return rs;
    }

```

```

    //ShowAllCustomerRecords method is used for displaying all customers
    public ResultSet ShowAllCustomerRecords() throws SQLException{
        // TODO Auto-generated method stub

        ResultSet rs = null;

```

```

String sql = null;

stmt = conn.connect().createStatement();

    sql = "SELECT * from aak_pcust";

    System.out.println("inside ShowAllCustomerRecords");

//query along quotes
String sqlQuery = sql;

// execute the query
rs = stmt.executeQuery(sqlQuery);

System.out.println("\n\nRetrieving Records successful");


// close the connection
conn.connect().close();


return rs;

}

```

//ShowAllBookedRecords method is used for displaying all booked rooms

```

public ResultSet ShowAllBookedRecords() throws SQLException{

    // TODO Auto-generated method stub

    ResultSet rs = null;

    String sql = null;

    stmt = conn.connect().createStatement();

        sql = "SELECT * from aak_pbooking";

        System.out.println("inside ShowAllBookedRecords");

//query along quotes
String sqlQuery = sql;

// execute the query
rs = stmt.executeQuery(sqlQuery);

System.out.println("\n\nRetrieving Records successful");


// close the connection

```

```

        conn.connect().close();

        return rs;
    }

    //InsertRoomRec method is used for adding another room to database
    public void InsertRoomRec(ArrayList<String> InsertRooms) throws SQLException
    {
        int i = 1;
        String query = "INSERT INTO aak_prooms (roomid, "
            + "roomtype,roomisavailable,roomprice,dining,gymtiming)
VALUES (?, ?, ?, ?, ?, ?)";
        try(PreparedStatement stmt = conn.connect().prepareStatement(query))
        {
            for (int index = 0; index < InsertRooms.size(); index++)
            {
                stmt.setString(i, InsertRooms.get(index));
                i++;
            }
            stmt.execute();
        } catch (Exception e) {
            // log info somewhere at least until it's properly tested/
            // you implement a better way of handling the error
            e.printStackTrace(System.err);
        }
    }
}

```

```

//InsertEmpRec method is used for adding another employee to database
public void InsertEmpRec(ArrayList<String> InsertEmps) throws SQLException
{
    int i = 1;

```

```

String query = "INSERT INTO aak_pemp (empid, "
                + "empname,empuser,emppass,empphone,empemail)
VALUES (?, ?, ?, ?, ?, ?)";

try(PreparedStatement stmt = conn.connect().prepareStatement(query))
{
    for (int index = 0; index < InsertEmps.size(); index++)
    {
        stmt.setString(i, InsertEmps.get(index));
        i++;
    }
    stmt.execute();
} catch (Exception e) {
    // log info somewhere at least until it's properly tested/
    // you implement a better way of handling the error
    e.printStackTrace(System.err);
}
}

```

```

//InsertCustRec method is used for adding another customer to database
public void InsertCustRec(ArrayList<String> InsertCusts) throws SQLException
{
    int i = 1;
    String query = "INSERT INTO aak_pcust (custid, "
                  +
"custname,custuser,custpass,custphone,custemail,custcurrbalance) VALUES (?, ?, ?, ?, ?, ?)";
    try(PreparedStatement stmt = conn.connect().prepareStatement(query))
    {
        for (int index = 0; index < InsertCusts.size(); index++)
        {
            stmt.setString(i, InsertCusts.get(index));
            i++;
        }
    }
}

```

```

        stmt.execute();
    } catch (Exception e) {
        // log info somewhere at least until it's properly tested/
        // you implement a better way of handling the error
        e.printStackTrace(System.err);
    }
}

//DeleteRoomRec method is used for deleting room from database
public void DeleteRoomRec(ArrayList<String> RoomData, String RoomIdToBeDel) throws
SQLException
{
    // TODO Auto-generated method stub

    String query = "DELETE FROM aak_rooms WHERE roomid = ?";
    PreparedStatement stmt = conn.connect().prepareStatement(query);
    stmt.setString(1, RoomIdToBeDel);
    System.out.println(query);
    System.out.println(stmt);
    stmt.execute();

}

//DeleteEmpRec method is used for deleting employee from database
public void DeleteEmpRec(ArrayList<String> EmpData, String EmpIdToBeDel) throws
SQLException
{
    // TODO Auto-generated method stub

    String query = "DELETE FROM aak_pemp WHERE empid = ?";
    PreparedStatement stmt = conn.connect().prepareStatement(query);
    stmt.setString(1, EmpIdToBeDel);

```

```

        System.out.println(query);

        System.out.println(stmt);

    stmt.execute();

}

//DeleteCustRec method is used for deleting customer from database
public void DeleteCustRec(ArrayList<String> CustData, String CustIdToBeDel) throws
SQLException
{

    // TODO Auto-generated method stub

    String query = "DELETE FROM aak_pcust WHERE custid = ?";
    PreparedStatement stmt = conn.connect().prepareStatement(query);
    stmt.setString(1, CustIdToBeDel);
    System.out.println(query);
    System.out.println(stmt);

    stmt.execute();

}

//UpdateRoomRec method is used for updating room in database
public void UpdateRoomRec(ArrayList<String> RoomData, String RoomIdToBeUpdated)
throws SQLException
{

    // TODO Auto-generated method stub

    int i = 1;

    String query = "UPDATE aak_prooms SET roomtype = ?, roomisavailable = ?,
roomprice = ?, dining = ?, gymtiming =? WHERE roomid = ?";
    try(PreparedStatement stmt = conn.connect().prepareStatement(query))
    {

        for (int index = 0; index < RoomData.size(); index++)

```

```

        {

            stmt.setString(i, RoomData.get(index));

            i++;

        }

        stmt.setString(6, RoomIdToBeUpdated);

        System.out.println(query);

        System.out.println(stmt);

        stmt.execute();

    }catch (Exception e)

    {

        // log info somewhere at least until it's properly tested/

        // you implement a better way of handling the error

        e.printStackTrace(System.err);

    }

}

```

//UpdateEmpRec method is used for updating employee in database

public void UpdateEmpRec(ArrayList<String> EmpData, String EmpIdToBeUpdated) throws
SQLException

```

{

    // TODO Auto-generated method stub

    int i = 1;

    String query = "UPDATE aak_pemp SET empname = ?, empuser = ?, emppass = ?,
empphone = ?, empemail =? WHERE empid = ?";

    try(PreparedStatement stmt = conn.connect().prepareStatement(query))

    {

        for (int index = 0; index < EmpData.size(); index++)

        {

            stmt.setString(i, EmpData.get(index));

            i++;

        }

    }

}

```



```

        stmt.setString(6, EmplIdToBeUpdated);

        System.out.println(query);

        System.out.println(stmt);

        stmt.execute();

    }catch (Exception e)
    {

        // log info somewhere at least until it's properly tested/
        // you implement a better way of handling the error
        e.printStackTrace(System.err);

    }

}

```

//UpdateCustRec method is used for updating customer in database

public void UpdateCustRec(ArrayList<String> CustData, String CustIdToBeUpdated) throws
SQLException

```

{

    // TODO Auto-generated method stub

    int i = 1;

    String query = "UPDATE aak_pcust SET custname = ?, custuser = ?, custpass = ?,
custphone = ?, custemail =?, custcurrbalance = ? WHERE custid = ?";

    try(PreparedStatement stmt = conn.connect().prepareStatement(query))
    {

        for (int index = 0; index < CustData.size(); index++)
        {

            stmt.setString(i, CustData.get(index));

            i++;

        }

        stmt.setString(7, CustIdToBeUpdated);

        System.out.println(query);

        System.out.println(stmt);

        stmt.execute();
    }
}

```

```

    }catch (Exception e)
    {
        // log info somewhere at least until it's properly tested/
        // you implement a better way of handling the error
        e.printStackTrace(System.err);
    }
}

```

//ShowSearchRoomRecords method is used for searching for a room with a specified ID in database

```

public ResultSet ShowSearchRoomRecords(String SearchRoomId) throws SQLException {
    // TODO Auto-generated method stub
    ResultSet rs = null;
    String sql = null;
    stmt = conn.connect().createStatement();
        sql = "SELECT * from aak_prooms WHERE roomid=";
        System.out.println("inside ShowSearchRoomRecords");
    //query along quotes
    String sqlQuery = sql + SearchRoomId;
    // execute the query
    System.out.println(sqlQuery);
    rs = stmt.executeQuery(sqlQuery);
    System.out.println(rs);
    System.out.println("\n\nRetrieving Records successful");
    // close the connection
    conn.connect().close();

    return rs;
}

```

```

}

package models;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnect
{
    //URL of the code database
    static final String DB_URL =
"jdbc:mysql://www.papademas.net:3307/510labs?autoReconnect=true&useSSL=false";

    //Credentials of database
    static final String USER = "db510", PASS = "510";

    public Connection connect() throws SQLException
    {
        return DriverManager.getConnection(DB_URL, USER, PASS);
    }
}

package models;

```

```

public class DisplayBookedRecModel
{

    private String roomid, custid;

    public String getRoomid() {
        return roomid;
    }
}

```

```

    }

    public void setRoomid(String roomid) {
        this.roomid = roomid;
    }

    public String getCustid() {
        return custid;
    }

    public void setCustid(String custid) {
        this.custid = custid;
    }

    public DisplayBookedRecModel(String roomid, String custid)
    {
        super();
        this.roomid=roomid;
        this.custid=custid;
    }
}

package models;

public class DisplayCustRecModel
{

    private String custid, custname,custuser,custpass,custemail,custphone,custcurrbalance;

    public String getCustid() {
        return custid;
    }

```

```
}
```

```
public void setCustid(String custid) {  
    this.custid = custid;  
}
```

```
public String getCustname() {  
    return custname;  
}
```

```
public void setCustname(String custname) {  
    this.custname = custname;  
}
```

```
public String getCustuser() {  
    return custuser;  
}
```

```
public void setCustuser(String custuser) {  
    this.custuser = custuser;  
}
```

```
public String getCustpass() {  
    return custpass;  
}
```

```
public void setCustpass(String custpass) {  
    this.custpass = custpass;  
}
```

```
public String getCustemail() {  
    return custemail;  
}
```

```
public void setCustemail(String custemail) {  
    this.custemail = custemail;  
}
```

```
public String getCustphone() {  
    return custphone;  
}
```

```
public void setCustphone(String custphone) {  
    this.custphone = custphone;  
}
```

```
public String getCustcurrbalance() {  
    return custcurrbalance;  
}
```

```
public void setCustcurrbalance(String custcurrbalance) {  
    this.custcurrbalance = custcurrbalance;  
}
```

```
public DisplayCustRecModel(String custid, String custname, String custuser, String custpass,  
String custphone, String custemail, String custcurrbalance)  
{  
    super();  
    this.custid=custid;  
    this.custname=custname;  
    this.custuser=custuser;  
    this.custpass=custpass;  
    this.custphone=custphone;  
    this.custemail=custemail;  
    this.custcurrbalance=custcurrbalance;  
}  
}
```

```
package models;
```

```
public class DisplayEmpRecModel  
{  
  
    private String empid, empname,empuser,emppass,empemail,empphone;  
  
    public String getEmpid() {  
        return empid;  
    }  
}
```

```
public void setEmpid(String empid) {  
    this.empid = empid;  
}
```

```
public String getEmpname() {  
    return empname;  
}
```

```
public void setEmpname(String empname) {  
    this.empname = empname;  
}
```

```
public String getEmpuser() {  
    return empuser;  
}
```

```
public void setEmpuser(String empuser) {  
    this.empuser = empuser;  
}
```

```
public String getEmppass() {  
    return emppass;  
}
```

```
public void setEmppass(String emppass) {  
    this.emppass = emppass;  
}
```

```
public String getEmpemail() {  
    return empemail;  
}
```



```

        public void setEmpemail(String empemail) {
            this.empemail = empemail;
        }

        public String getEmppphone() {
            return empphone;
        }

        public void setEmppphone(String empphone) {
            this.empphone = empphone;
        }

        public DisplayEmpRecModel(String empid, String empname, String empuser, String
emppass, String empphone, String empemail)
        {
            super();
            this.empid=empid;
            this.empname=empname;
            this.empuser=empuser;
            this.emppass=emppass;
            this.empphone=empphone;
            this.empemail=empemail;
        }
    }

package models;

public class DisplayModel
{

    private String roomid, roomtype,roomisavailable,roomprice,dining,gymtiming;

```

```
public String getRoomid() {  
    return roomid;  
}
```

```
public void setRoomid(String roomid) {  
    this.roomid = roomid;  
}
```

```
public String getRoomtype() {  
    return roomtype;  
}
```

```
public void setRoomtype(String roomtype) {  
    this.roomtype = roomtype;  
}
```

```
public String getRoomisavailable() {  
    return roomisavailable;  
}
```

```
public void setRoomisavailable(String roomisavailable) {  
    this.roomisavailable = roomisavailable;  
}
```

```
public String getRoomprice() {  
    return roomprice;  
}
```

```
public void setRoomprice(String roomprice) {  
    this.roomprice = roomprice;  
}
```

```
}
```

```
public String getDining() {
```

```
    return dining;
```

```
}
```

```
public void setDining(String dining) {
```

```
    this.dining = dining;
```

```
}
```

```
public String getGymtiming() {
```

```
    return gymtiming;
```

```
}
```

```
public void setGymtiming(String gymtiming) {
```

```
    this.gymtiming = gymtiming;
```

```
}
```

```
    public DisplayModel(String roomid, String roomtype, String roomisavailable, String  
roomprice, String dining, String gymtiming)
```

```
{
```

```
    super();
```

```
    this.roomid=roomid;
```

```
    this.roomtype=roomtype;
```

```
    this.roomisavailable=roomisavailable;
```

```
    this.roomprice=roomprice;
```

```
    this.dining=dining;
```

```
    this.gymtiming=gymtiming;
```

```
}
```

```

}

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" style="-fx-background-
color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.AddCust">
    <children>
        <Label layoutX="191.0" layoutY="30.0" text="Add Customer To Database"
textFill="#0598ff">
            <font>
                <Font size="18.0" />
            </font></Label>
        <TextField fx:id="NewCustId" layoutX="226.0" layoutY="69.0"
promptText="Enter Customer ID" style="-fx-background-color: transparent; -fx-
border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewCustName" layoutX="226.0" layoutY="108.0"
promptText="Enter Customer Name" style="-fx-background-color: transparent; -fx-
border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewCustUser" layoutX="226.0" layoutY="144.0"
prefHeight="27.0" prefWidth="158.0" promptText="Enter Customer Username" style="-
fx-background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="NewCustPass" layoutX="226.0" layoutY="183.0"
promptText="Enter Customer Password" style="-fx-background-color: transparent; -
fx-border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewCustPhone" layoutX="226.0" layoutY="230.0"
promptText="Enter Phone Number" style="-fx-background-color: transparent; -fx-
border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewCustEmail" layoutX="226.0" layoutY="269.0"
promptText="Enter Email" style="-fx-background-color: transparent; -fx-border-
color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewCustCurrBalance" layoutX="224.0" layoutY="309.0"
promptText="Enter Current Balance" style="-fx-background-color: transparent; -fx-
border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <Button layoutX="409.0" layoutY="209.0" mnemonicParsing="false"
onAction="#initData" style="-fx-background-color: #0598ff;" text="Add Customer"
textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="496.0" layoutY="31.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

```

```

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" style="-fx-background-
color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.AddEmp">
    <children>
        <Label layoutX="184.0" layoutY="32.0" text="Add Employee To Database"
textFill="#0598ff">
            <font>
                <Font size="19.0" />
            </font></Label>
        <TextField fx:id="NewEmpId" layoutX="226.0" layoutY="80.0" promptText="Enter
Employee ID" style="-fx-background-color: transparent; -fx-border-color: #0598ff;
-fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewEmpName" layoutX="226.0" layoutY="119.0"
promptText="Enter Employee Name" style="-fx-background-color: transparent; -fx-
border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewEmpUser" layoutX="226.0" layoutY="165.0"
prefHeight="26.0" prefWidth="155.0" promptText="Enter Employee Username" style="-
fx-background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="NewEmpPass" layoutX="226.0" layoutY="211.0"
promptText="Enter Employee Password" style="-fx-background-color: transparent; -
fx-border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewEmpPhone" layoutX="226.0" layoutY="261.0"
promptText="Enter Phone Number" style="-fx-background-color: transparent; -fx-
border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewEmpEmail" layoutX="226.0" layoutY="303.0"
promptText="Enter Email" style="-fx-background-color: transparent; -fx-border-
color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <Button layoutX="433.0" layoutY="188.0" mnemonicParsing="false"
onAction="#initData" style="-fx-background-color: #0598ff;" text="Add Employee"
textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="480.0" layoutY="33.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" style="-fx-background-
color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.AddRoom">
    <children>
        <TextField fx:id="NewRoomId" layoutX="226.0" layoutY="72.0"
promptText="Enter Room ID" style="-fx-background-color: transparent; -fx-border-
color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewRoomType" layoutX="226.0" layoutY="123.0"
promptText="Enter Room Type" style="-fx-background-color: transparent; -fx-border-
color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
    </children>
</AnchorPane>

```

```

        <TextField fx:id="NewRoomIsAvailable" layoutX="226.0" layoutY="168.0"
promptText="Enter Room Is Available" style="-fx-background-color: transparent; -
fx-border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewRoomPrice" layoutX="226.0" layoutY="221.0"
promptText="Enter Room Price" style="-fx-background-color: transparent; -fx-
border-color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="NewDining" layoutX="226.0" layoutY="273.0"
promptText="Enter Dining" style="-fx-background-color: transparent; -fx-border-
color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <Label layoutX="211.0" layoutY="24.0" text="Add Room To Database"
textFill="#0598ff">
            <font>
                <Font size="17.0" />
            </font></Label>
        <TextField fx:id="NewGymTiming" layoutX="226.0" layoutY="319.0"
promptText="Enter GymTiming" style="-fx-background-color: transparent; -fx-border-
color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <Button layoutX="454.0" layoutY="222.0" mnemonicParsing="false"
onAction="#initData" style="-fx-background-color: #0598ff;" text="Add Room"
textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="473.0" layoutY="38.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

```

```

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" style="-fx-background-
color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.Admin">
    <children>
        <Label layoutX="227.0" layoutY="25.0" prefHeight="35.0" prefWidth="146.0"
text="Administator" textFill="#0598ff">
            <font>
                <Font size="24.0" />
            </font></Label>
        <Button layoutX="159.0" layoutY="111.0" mnemonicParsing="false"
onAction="#GoAdminViewRoom" prefHeight="26.0" prefWidth="134.0" style="-fx-
background-color: #0598ff;" text="View Rooms" textFill="WHITE" />
        <Button layoutX="160.0" layoutY="151.0" mnemonicParsing="false"
onAction="#GoAdminViewEmp" prefHeight="26.0" prefWidth="134.0" style="-fx-
background-color: #0598ff;" text="View Employees" textFill="WHITE" />
        <Button layoutX="160.0" layoutY="187.0" mnemonicParsing="false"
onAction="#GoAdminViewCust" prefHeight="26.0" prefWidth="134.0" style="-fx-
background-color: #0598ff;" text="View Customers" textFill="WHITE" />
        <Button layoutX="160.0" layoutY="232.0" mnemonicParsing="false"
onAction="#GoAdminViewBooked" prefHeight="26.0" prefWidth="134.0" style="-fx-
background-color: #0598ff;" text="View Booked Rooms" textFill="WHITE" />
        <Button layoutX="160.0" layoutY="275.0" mnemonicParsing="false"
onAction="#GoAdminAddRoom" prefHeight="26.0" prefWidth="134.0" style="-fx-
background-color: #0598ff;" text="Add Room" textFill="WHITE" />
    </children>
</AnchorPane>

```

```

        <Button layoutX="160.0" layoutY="317.0" mnemonicParsing="false"
onAction="#GoAdminAddEmp" prefHeight="26.0" prefWidth="134.0" style="-fx-
background-color: #0598ff;" text="Add Employee" textFill="WHITE" />
        <Button layoutX="159.0" layoutY="356.0" mnemonicParsing="false"
onAction="#GoAdminAddCust" prefHeight="26.0" prefWidth="134.0" style="-fx-
background-color: #0598ff;" text="Add Customer" textFill="WHITE" />
        <Button layoutX="340.0" layoutY="111.0" mnemonicParsing="false"
onAction="#GoAdminDelRoom" prefHeight="26.0" prefWidth="122.0" style="-fx-
background-color: #0598ff;" text="Delete Room" textFill="WHITE" />
        <Button layoutX="340.0" layoutY="151.0" mnemonicParsing="false"
onAction="#GoAdminDelEmp" prefHeight="26.0" prefWidth="122.0" style="-fx-
background-color: #0598ff;" text="Delete Employee" textFill="WHITE" />
        <Button layoutX="340.0" layoutY="187.0" mnemonicParsing="false"
onAction="#GoAdminDelCust" prefHeight="26.0" prefWidth="122.0" style="-fx-
background-color: #0598ff;" text="Delete Customer" textFill="WHITE" />
        <Button layoutX="338.0" layoutY="232.0" mnemonicParsing="false"
onAction="#GoAdminUpdateRoom" prefHeight="26.0" prefWidth="121.0" style="-fx-
background-color: #0598ff;" text="Update Room" textFill="WHITE" />
        <Button layoutX="338.0" layoutY="316.0" mnemonicParsing="false"
onAction="#GoAdminUpdateEmp" prefHeight="26.0" prefWidth="122.0" style="-fx-
background-color: #0598ff;" text="Update Employee" textFill="WHITE" />
        <Button layoutX="338.0" layoutY="275.0" mnemonicParsing="false"
onAction="#GoAdminUpdateCust" prefHeight="26.0" prefWidth="122.0" style="-fx-
background-color: #0598ff;" text="Update Customer" textFill="WHITE" />
        <Label layoutX="191.0" layoutY="73.0" text="What would you like to do?"
textFill="#0598ff">
            <font>
                <Font size="18.0" />
            </font>
        </Label>
        <Button fx:id="closebutton" layoutX="530.0" layoutY="30.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

```

```

<AnchorPane fx:id="anchortable" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
style="-fx-background-color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.Customer">
    <children>
        <Label layoutX="251.0" layoutY="24.0" prefHeight="35.0" prefWidth="116.0"
text="Customer" textFill="#0598ff">
            <font>
                <Font size="24.0" />
            </font></Label>
    </children>

```

```

        <Button layoutX="250.0" layoutY="137.0" mnemonicParsing="false"
onAction="#GoCustomerViewRoom" style="-fx-background-color: #0598ff;" text="View
ALL Rooms" textFill="WHITE" />
        <TextField fx:id="textArea" layoutX="42.0" layoutY="187.0" prefHeight="26.0"
prefWidth="382.0" promptText="Enter the Room ID you are looking for" />
        <Button layoutX="445.0" layoutY="187.0" mnemonicParsing="false"
onAction="#searchButtonClicked" prefHeight="26.0" prefWidth="98.0" style="-fx-
background-color: #0598ff;" text="Search" textFill="WHITE" />
        <TableView fx:id="tableView" layoutX="37.0" layoutY="223.0"
prefHeight="86.0" prefWidth="510.0">
            <columns>
                <TableColumn fx:id="col_roomid" prefWidth="75.0" text="RoomId" />
                <TableColumn fx:id="col_roomtype" prefWidth="75.0" text="RoomType" />
                <TableColumn fx:id="col_roomisavailable" prefWidth="75.0"
text="RoomIsAvailable" />
                <TableColumn fx:id="col_roomprice" prefWidth="75.0" text="RoomPrice"
/>
                <TableColumn fx:id="col_dining" prefWidth="75.0" text="Dining" />
                <TableColumn fx:id="col_gymtiming" prefWidth="75.0" text="GymTiming"
/>
            </columns>
            <columnResizePolicy>
                <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
            </columnResizePolicy>
        </TableView>
        <Button layoutX="242.0" layoutY="318.0" mnemonicParsing="false"
onAction="#bookButtonClicked" style="-fx-background-color: #0598ff;" text="Book
This Room" textFill="WHITE" />
        <Label layoutX="218.0" layoutY="82.0" text="What would you like to do?"
textFill="#0598ff">
            <font>
                <Font size="15.0" />
            </font>
        </Label>
        <Button fx:id="closebutton" layoutX="518.0" layoutY="29.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

```

```

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" style="-fx-background-
color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.DelCust">
    <children>
        <Label layoutX="121.0" layoutY="42.0" prefHeight="36.0" prefWidth="365.0"
text="Delete Customer From Database" textFill="#0598ff">
            <font>
                <Font size="25.0" />
            </font></Label>
    </children>

```



```

        <TextField fx:id="DelCustId" layoutX="132.0" layoutY="174.0"
prefHeight="26.0" prefWidth="350.0" promptText="Enter the Customer ID of the
Customer you want to delete" style="-fx-background-color: transparent; -fx-border-
color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <Button layoutX="248.0" layoutY="244.0" mnemonicParsing="false"
onAction="#deleteButtonClicked" style="-fx-background-color: #0598ff;"
text="Delete Customer" textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="282.0" layoutY="286.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

```

```

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" style="-fx-background-
color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.DeLEmp">
    <children>
        <Label layoutX="120.0" layoutY="37.0" text="Delete Employee From Database"
textFill="#0598ff">
            <font>
                <Font size="25.0" />
            </font></Label>
        <TextField fx:id="DeLEmpId" layoutX="122.0" layoutY="174.0"
prefHeight="26.0" prefWidth="360.0" promptText="Enter Employee ID of Employee you
want to delete" style="-fx-background-color: transparent; -fx-border-color:
#0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <Button layoutX="247.0" layoutY="248.0" mnemonicParsing="false"
onAction="#deleteButtonClicked" style="-fx-background-color: #0598ff;"
text="Delete Employee" textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="276.0" layoutY="288.0"
mnemonicParsing="false" onAction="#closeButtonClicked" style="-fx-background-
color: #0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

```

```

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" style="-fx-background-
color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.DeLRoom">
    <children>

```

```

        <Label layoutX="141.0" layoutY="41.0" prefHeight="36.0" prefWidth="325.0"
text="Delete Room From Database" textFill="#0598ff">
        <font>
        <Font size="25.0" />
        </font></Label>
        <TextField fx:id="DelRoomId" layoutX="140.0" layoutY="187.0"
prefHeight="26.0" prefWidth="318.0" promptText="Enter Room ID for Room that you
want to delete" style="-fx-background-color: transparent; -fx-border-color:
#0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <Button layoutX="257.0" layoutY="249.0" mnemonicParsing="false"
onAction="#deleteButtonClicked" style="-fx-background-color: #0598ff;"
text="Delete Room" textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="278.0" layoutY="295.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.layout.AnchorPane?>

```

```

<AnchorPane fx:id="anchortable" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.Display">
    <children>
        <TableView fx:id="tableView" layoutX="169.0" layoutY="92.0"
prefHeight="400.0" prefWidth="600.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="0.0">
            <columns>
                <TableColumn fx:id="col_roomid" prefWidth="75.0" text="RoomId" />
                <TableColumn fx:id="col_roomtype" prefWidth="75.0" text="RoomType" />
                <TableColumn fx:id="col_roomisavailable" prefWidth="75.0"
text="RoomIsAvailable" />
                <TableColumn fx:id="col_roomprice" prefWidth="75.0" text="RoomPrice"
/>
                <TableColumn fx:id="col_dining" prefWidth="75.0" text="Dining" />
                <TableColumn fx:id="col_gymtiming" prefWidth="75.0" text="GymTiming"
/>
            </columns>
            <columnResizePolicy>
                <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
            </columnResizePolicy>
        </TableView>
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.layout.AnchorPane?>

```

```

<AnchorPane fx:id="anchortable" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.DisplayBookedRec">
    <children>
        <TableView fx:id="tableView" layoutX="180.0" layoutY="73.0"
prefHeight="400.0" prefWidth="600.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="0.0">
            <columns>
                <TableColumn fx:id="col_roomid" prefWidth="75.0" text="RoomId" />
                <TableColumn fx:id="col_custid" prefWidth="75.0" text="CustId" />
            </columns>
            <columnResizePolicy>
                <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
            </columnResizePolicy>
        </TableView>
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.layout.AnchorPane?>

```

```

<AnchorPane fx:id="anchortable" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.DisplayCustRec">
    <children>
        <TableView fx:id="tableView" layoutX="200.0" layoutY="100.0"
prefHeight="400.0" prefWidth="600.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="0.0">
            <columns>
                <TableColumn fx:id="col_custid" prefWidth="75.0" text="CustId" />
                <TableColumn fx:id="col_custname" prefWidth="75.0" text="CustName" />
                <TableColumn fx:id="col_custuser" prefWidth="75.0" text="CustUser" />
                <TableColumn fx:id="col_custpass" prefWidth="75.0" text="CustPass" />
                <TableColumn fx:id="col_custphone" prefWidth="75.0" text="CustPhone"
/>
                <TableColumn fx:id="col_custemail" prefWidth="75.0" text="CustEmail"
/>
                <TableColumn fx:id="col_custcurrbalance" prefWidth="75.0"
text="CustCurrBalance" />
            </columns>
            <columnResizePolicy>
                <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
            </columnResizePolicy>
        </TableView>
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane fx:id="anchortable" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.DisplayEmpRec">
    <children>
        <TableView fx:id="tableView" layoutX="162.0" layoutY="100.0"
prefHeight="400.0" prefWidth="600.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="0.0">
            <columns>
                <TableColumn fx:id="col_empid" prefWidth="75.0" text="EmpId" />
                <TableColumn fx:id="col_empname" prefWidth="75.0" text="EmpName" />
                <TableColumn fx:id="col_empuser" prefWidth="75.0" text="EmpUser" />
                <TableColumn fx:id="col_emppass" prefWidth="75.0" text="EmpPass" />
                <TableColumn fx:id="col_empphone" prefWidth="75.0" text="EmpPhone" />
                <TableColumn fx:id="col_emppemail" prefWidth="75.0" text="EmpEmail" />
            </columns>
            <columnResizePolicy>
                <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
            </columnResizePolicy>
        </TableView>
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.Employee">
    <children>
        <Label layoutX="240.0" layoutY="47.0" text="Employee" textFill="#0598ff">
            <font>
                <Font size="28.0" />
            </font></Label>
        <Button layoutX="242.0" layoutY="187.0" mnemonicParsing="false"
onAction="#GoEmployeeViewRoom" style="-fx-background-color: #0598ff;" text="View
Room Details" textFill="WHITE" />
        <Button layoutX="232.0" layoutY="234.0" mnemonicParsing="false"
onAction="#GoEmployeeViewCust" style="-fx-background-color: #0598ff;" text="View
Customer Details" textFill="WHITE" />
        <Label layoutX="190.0" layoutY="113.0" text="What would you like to do?"
textFill="#0598ff">
            <font>
                <Font size="18.0" />
            </font>
        </Label>
    </children>
</AnchorPane>

```

```

        <Button fx:id="closebutton" layoutX="497.0" layoutY="34.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ChoiceBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

```

```

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" style="-fx-background-
color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.LoginAccount">
    <children>
        <TextField fx:id="usernamefield" layoutX="229.0" layoutY="187.0"
promptText="Enter Username" style="-fx-background-color: transparent; -fx-border-
color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <Button layoutX="207.0" layoutY="277.0" mnemonicParsing="false"
onAction="#LoginInitiate" prefHeight="26.0" prefWidth="73.0" style="-fx-
background-color: #0598ff;" text="Login" textFill="WHITE" />
        <ChoiceBox fx:id="UserChoice" accessibleRole="MENU_BUTTON" layoutX="228.0"
layoutY="128.0" prefWidth="150.0" style="-fx-background-color: #0598ff;" />
        <Button layoutX="337.0" layoutY="277.0" mnemonicParsing="false"
onAction="#clearFields" prefHeight="26.0" prefWidth="74.0" style="-fx-background-
color: #0598ff;" text="Clear" textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="273.0" layoutY="321.0"
mnemonicParsing="false" onAction="#closeButtonAction" prefHeight="26.0"
prefWidth="74.0" style="-fx-background-color: #0598ff;" text="Exit"
textFill="WHITE" />
        <PasswordField fx:id="passwordfield" layoutX="229.0" layoutY="222.0"
promptText="Enter Password" style="-fx-background-color: transparent; -fx-border-
color: #0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <Label layoutX="207.0" layoutY="37.0" prefHeight="30.0" prefWidth="195.0"
text="Welcome to IIT Hotel" textFill="#0598ff">
            <font>
                <Font size="20.0" />
            </font>
        </Label>
    </children>
</AnchorPane>

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

```

```

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.UpdateCust">
    <children>
        <Label layoutX="160.0" layoutY="37.0" text="Update Customer In Database"
textFill="#0598ff">
            <font>
                <Font size="21.0" />
            </font></Label>
        <TextField fx:id="UpdateCustId" layoutX="90.0" layoutY="88.0"
prefHeight="26.0" prefWidth="293.0" promptText="Enter Customer ID whose details
you want to update" style="-fx-background-color: transparent; -fx-border-color:
#0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="UpdateCustName" layoutX="90.0" layoutY="124.0"
prefHeight="27.0" prefWidth="294.0" promptText="Enter Customer Name" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateCustUser" layoutX="90.0" layoutY="161.0"
prefHeight="27.0" prefWidth="294.0" promptText="Enter Customer Username" style="-
fx-background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateCustPass" layoutX="90.0" layoutY="201.0"
prefHeight="27.0" prefWidth="294.0" promptText="Enter Customer Password" style="-
fx-background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateCustPhone" layoutX="90.0" layoutY="244.0"
prefHeight="27.0" prefWidth="294.0" promptText="Enter Phone Number" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateCustEmail" layoutX="90.0" layoutY="283.0"
prefHeight="27.0" prefWidth="294.0" promptText="Enter Email" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateCustCurrBalance" layoutX="90.0" layoutY="319.0"
prefHeight="27.0" prefWidth="294.0" promptText="Enter Current Balance" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <Button layoutX="425.0" layoutY="175.0" mnemonicParsing="false"
onAction="#updateButtonClicked" style="-fx-background-color: #0598ff;"
text="Update Customer" textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="454.0" layoutY="216.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"

```



```

xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.UpdateEmp">
    <children>
        <Label layoutX="166.0" layoutY="31.0" text="Update Employee In Database"
textFill="#0598ff">
            <font>
                <Font size="20.0" />
            </font></Label>
        <TextField fx:id="UpdateEmpId" layoutX="54.0" layoutY="83.0"
prefHeight="26.0" prefWidth="316.0" promptText="Enter Employee ID of Employee that
you want to update" style="-fx-background-color: transparent; -fx-border-color:
#0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="UpdateEmpName" layoutX="54.0" layoutY="129.0"
prefHeight="26.0" prefWidth="316.0" promptText="Enter Employee Name" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateEmpUser" layoutX="54.0" layoutY="173.0"
prefHeight="26.0" prefWidth="316.0" promptText="Enter Employee Username" style="-
fx-background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateEmpPass" layoutX="54.0" layoutY="222.0"
prefHeight="26.0" prefWidth="316.0" promptText="Enter Employee Password" style="-
fx-background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateEmpPhone" layoutX="54.0" layoutY="270.0"
prefHeight="26.0" prefWidth="316.0" promptText="Enter Phone Number" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateEmpEmail" layoutX="54.0" layoutY="312.0"
prefHeight="26.0" prefWidth="316.0" promptText="Enter Email" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <Button layoutX="439.0" layoutY="187.0" mnemonicParsing="false"
onAction="#updateButtonClicked" style="-fx-background-color: #0598ff;"
text="Update Employee" textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="477.0" layoutY="235.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" style="-fx-background-
color: white;" xmlns="http://javafx.com/javafx/11.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.UpdateRoom">
    <children>
        <Label layoutX="184.0" layoutY="37.0" text="Update Room In Database"
textFill="#0598ff">
            <font>
                <Font size="20.0" />
            </font>

```

```

        </font></Label>
        <TextField fx:id="UpdateRoomId" layoutX="76.0" layoutY="93.0"
prefHeight="26.0" prefWidth="330.0" promptText="Enter Room ID of Room that you
want to delete" style="-fx-background-color: transparent; -fx-border-color:
#0598ff; -fx-border-width: 0px 0px 2px 0px;" />
        <TextField fx:id="UpdateRoomType" layoutX="76.0" layoutY="133.0"
prefHeight="26.0" prefWidth="330.0" promptText="Enter Room Type" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateRoomIsAvailable" layoutX="76.0" layoutY="174.0"
prefHeight="26.0" prefWidth="330.0" promptText="Enter Room Is Available" style="-
fx-background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateRoomPrice" layoutX="76.0" layoutY="213.0"
prefHeight="26.0" prefWidth="330.0" promptText="Enter Room Price" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateDining" layoutX="76.0" layoutY="255.0"
prefHeight="26.0" prefWidth="330.0" promptText="Enter Dining" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <TextField fx:id="UpdateGymTiming" layoutX="76.0" layoutY="304.0"
prefHeight="26.0" prefWidth="330.0" promptText="Enter Gym Timing" style="-fx-
background-color: transparent; -fx-border-color: #0598ff; -fx-border-width: 0px
0px 2px 0px;" />
        <Button layoutX="445.0" layoutY="200.0" mnemonicParsing="false"
onAction="#updateButtonClicked" style="-fx-background-color: #0598ff;"
text="Update Room" textFill="WHITE" />
        <Button fx:id="closebutton" layoutX="472.0" layoutY="255.0"
mnemonicParsing="false" onAction="#closeButtonAction" style="-fx-background-color:
#0598ff;" text="Exit" textFill="WHITE" />
    </children>
</AnchorPane>

```