# JAVA CODE OF GUEST HOUSE MANAGEMENT SYSTEM:

## Account:

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Optional;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.geometry.Pos;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.ButtonBar;

import javafx.scene.control.ButtonType;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.scene.layout.VBox;

import javafx.stage.Modality;

import javafx.stage.Stage;

import verifications.Checks;

public class AccountController implements Initializable

{

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private Label name;

@FXML private Label gender;

@FXML private Label dob;

@FXML private Label cnic;

@FXML private Label age;

@FXML private Label username;

@FXML private Label joiningDate;

@FXML private Label mob1;

@FXML private Label mob2;

@FXML private Label mob3;

@FXML private Label address1\_1;

@FXML private Label address1\_2;

@FXML private Label address2\_1;

@FXML private Label address2\_2;

@FXML private Label address3\_1;

@FXML private Label address3\_2;

String usernameOfLoggedInPerson;

String customerID = "";

public void setParameter(String s)

{

try

{

System.out.println("\nUser Name from Account : "+s);

username.setText(s);

usernameOfLoggedInPerson = s;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

//ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, CNIC, joining\_date from customer where email = '" + s + "';");

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, CNIC, joining\_date, address, city\_name, country\_name, phone\_number, customer.customerID from customer, address, customer\_phone, customer\_address, country, city where Customer.customerID = customer\_phone.customerID and customer\_address.customerID = customer.customerID and customer\_address.addressID = address.addressID and address.City\_cityID = city.cityID and city.countryID = country.countryID and email = '" + s + "';");

boolean temp = rs.next();

name.setText(rs.getString(1) + " " + rs.getString(2));

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

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

System.out.println("\nDate Of Birth : "+dob.getText());

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

joiningDate.setText(rs.getString(6));

customerID = rs.getString(11);

//for Age we Do,

String dataForAge[] = dob.getText().split("-");

int ageInInt = Checks.ageReturner(Integer.parseInt(dataForAge[0]), Integer.parseInt(dataForAge[1]), Integer.parseInt(dataForAge[2]));

age.setText(Integer.toString(ageInInt));

rs = stmt.executeQuery("select distinct address, city\_name, country\_name from customer, address, customer\_address, country, city where customer\_address.customerID = customer.customerID and customer\_address.addressID = address.addressID and address.City\_cityID = city.cityID and city.countryID = country.countryID and customer.email = '"+ s +"';");

temp = rs.next();

address1\_1.setText(rs.getString(1) + ", ");

address1\_2.setText(rs.getString(2) + ", " + rs.getString(3));

int counter = 1;

while(rs.next())

{

if(counter == 1)

{

address2\_1.setText(rs.getString(1) + ", ");

address2\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

else

{

address3\_1.setText(rs.getString(1) + ", ");

address3\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

counter++;

}

rs = stmt.executeQuery("select distinct phone\_number from customer, customer\_phone where Customer.customerID = customer\_phone.customerID and customer.email = '"+s+"';");

temp = rs.next();

mob1.setText(rs.getString(1));

counter = 1;

while(rs.next())

{

if(counter == 1)

{

mob2.setText(rs.getString(1));

}

else

{

mob3.setText(rs.getString(1));

}

counter++;

}

}

catch(Exception e)

{

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

}

}

public void changeInfoButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ChangingInfo.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ChangingInfoController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

System.out.println("Username in back button of Account : " + username.getText());

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerProfileController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

System.out.println("Username in back button of Account : " + username.getText());

}

public void resetPasswordButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ForgetPassword2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ForgetPassword2Controller controller = loader.getController();

controller.setParameter("Account", usernameOfLoggedInPerson, cnic.getText(), mob1.getText());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void deleteAccountButtonPushed(ActionEvent event) throws IOException, SQLException

{

Alert a1 = new Alert(Alert.AlertType.CONFIRMATION);

a1.setTitle("Deletion Confirmation");

a1.setContentText("Are you sure you want to DELETE Your Account? ");

a1.setHeaderText(null); //you can also set header

ButtonType yes = new ButtonType("YES", ButtonBar.ButtonData.YES);

ButtonType cancel = new ButtonType("Cancel", ButtonBar.ButtonData.CANCEL\_CLOSE);

a1.getButtonTypes().setAll(yes, cancel);

Optional<ButtonType> result = a1.showAndWait();

if(result.get() == yes) //type can also be "OK" as well

{

System.out.println("OKAY");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from customer where customerID = '" + customerID + "';");

rs.next();

String sql = "INSERT INTO `mydb`.`customer\_deleted` (`customerID`, `first\_name`, `last\_name`, `gender`, `DOB`, `CNIC`, `email`, `joining\_date`, `leaving\_date`) VALUES ('"+rs.getString(1)+"', '"+rs.getString(2)+"', '"+rs.getString(3)+"', '"+rs.getString(4)+"', '"+rs.getString(5)+"', '"+rs.getString(6)+"', '"+rs.getString(7)+"', '"+rs.getString(9)+"', '"+Checks.getCurrentdate()+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL InjectionS

int rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select complaintID from complaint where customerID = '" + customerID + "';");

while(rs.next())

{

sql = "DELETE FROM `mydb`.`response` WHERE (`comlpaintID` = '"+rs.getString(1)+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL InjectionS

rowsAffected = preparedStatement.executeUpdate();

}

sql = "DELETE FROM `mydb`.`complaint` WHERE (`customerID` = '"+customerID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL InjectionS

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select billsID from bills where customerID = '" + customerID + "';");

while(rs.next())

{

sql = "DELETE FROM `mydb`.`service\_record` WHERE (`billsID` = '"+rs.getString(1)+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL InjectionS

rowsAffected = preparedStatement.executeUpdate();

}

sql = "DELETE FROM `mydb`.`bills` WHERE (`customerID` = '"+customerID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL InjectionS

rowsAffected = preparedStatement.executeUpdate();

sql = "DELETE FROM `mydb`.`customer` WHERE (`customerID` = '"+customerID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL InjectionS

rowsAffected = preparedStatement.executeUpdate();

Parent tableViewParent = FXMLLoader.load(getClass().getResource("CustomerLogin.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

Alert a2 = new Alert(Alert.AlertType.INFORMATION);

a2.setTitle("Successfully Deleted ");

a2.setContentText("Your Account has been Deleted successfully, We Will Miss You !");

a2.setHeaderText(null); //you can also set header

a2.showAndWait();

}

else

{

System.out.println("Cancel");

}

}

public void logoutButtonPushed(ActionEvent event) throws IOException //this is INFORMATION

{

Alert a1 = new Alert(Alert.AlertType.CONFIRMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a1.setTitle("Logout Confirmation");

a1.setContentText("Are you sure you want to LOGOUT? ");

a1.setHeaderText(null); //you can also set header

ButtonType yes = new ButtonType("YES", ButtonBar.ButtonData.YES);

ButtonType cancel = new ButtonType("Cancel", ButtonBar.ButtonData.CANCEL\_CLOSE);

a1.getButtonTypes().setAll(yes, cancel);

Optional<ButtonType> result = a1.showAndWait();

if(result.get() == yes) //type can also be "OK" as well

{

System.out.println("OKAY");

Parent tableViewParent = FXMLLoader.load(getClass().getResource("CustomerLogin.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

{

System.out.println("Cancel");

}

}

@Override

public void initialize(URL url, ResourceBundle rb)

{

logo = new ImageView("logo1.png");

logo2 = new ImageView("account.png");

// TODO

}

}

## Admin Account Comtroller

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Optional;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.ButtonType;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AdminAccountController implements Initializable {

@FXML private ImageView logo;

String empID = "";

String cnic = "";

String mob = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select CNIC, phone\_number from employee, employee\_phone where employee.empID = '"+empID+"';");

rs.next();

cnic = rs.getString(1);

mob = rs.getString(2);

}

public void logOutButtonPushed(ActionEvent event) throws IOException, SQLException

{

Alert a1 = new Alert(Alert.AlertType.CONFIRMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a1.setTitle("LOGOUT");

a1.setContentText("ARE YOU SURE YOU WANT TO LOGOUT? ");

a1.setHeaderText(null); //you can also set header

Optional<ButtonType> result = a1.showAndWait();

if(result.get() == ButtonType.OK) //type can also be "OK" as well

{

System.out.println("OKAY");

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminLogin.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

{

System.out.println("Cancel");

}

}

public void changePasswordButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ForgetPassword2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ForgetPassword2Controller controller = loader.getController();

controller.setParameter("AdminAccount", empID, cnic, mob);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AdminProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Admin Employee Controller

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AdminEmployeeController implements Initializable {

@FXML private ImageView logo;

String empID = "";

// String cnic = "";

// String mob = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

// Connection conn;

// conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

// Statement stmt = conn.createStatement();

//

// ResultSet rs = stmt.executeQuery("select CNIC, phone\_number from employee, employee\_phone where employee.empID = '"+empID+"';");

// rs.next();

// cnic = rs.getString(1);

// mob = rs.getString(2);

}

public void addNewEmployee(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RegisterEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RegisterEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void deleteEmployee(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("DeleteEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

DeleteEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void changeSalary(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ChangeSalary.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ChangeSalaryController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AdminProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Admin Login Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.PrintWriter;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import java.util.Scanner;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.CheckBox;

import javafx.scene.control.Label;

import javafx.scene.control.PasswordField;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AdminLoginController implements Initializable {

/\*\*

\* Initializes the controller class.

\*/

@FXML private Label label;

@FXML private ImageView msgIcon;

@FXML private ImageView lockIcon;

@FXML private ImageView close;

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private TextField username;

@FXML private PasswordField password;

String empID = "";

@FXML

public void close\_app(MouseEvent event)

{

System.exit(0);

}

@FXML private CheckBox rememberMe;

//function for CHECK BOX

public void checkBox(String uName, String pass) throws FileNotFoundException

{

PrintWriter pw = new PrintWriter(new FileOutputStream("admin.txt"));

if(rememberMe.isSelected())

{

pw.println(uName + " " + pass);

System.out.println("WRITTEN \n");

}

else

{

pw.println();

System.out.println("\nNOT Written\n");

}

pw.close();

}

public void getCreditials() throws FileNotFoundException

{

Scanner reader = new Scanner(new FileInputStream("admin.txt"));

if(reader.hasNextLine())

{

String credentials[] = reader.nextLine().split(" ");

username.setText(credentials[0]);

password.setText(credentials[1]);

rememberMe.setSelected(true);

}

reader.close();

}

public void loginButtonPushed(ActionEvent event) throws IOException, SQLException

{

String tempU = username.getText();

String tempP = password.getText();

if(!tempU.equals("") && !tempP.equals(""))

{

if(tempU.subSequence(0, 5).equals("empID") || tempU.subSequence(0, 5).equals("empid"))

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select admin.empID from employee, admin where admin.empID = employee.empID and admin.empID = '" + tempU.substring(5, tempU.length()) + "' and admin.password = '"+tempP+"';");

if(rs.next())

{

checkBox(tempU, tempP);

empID = rs.getString(1);

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AdminProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("Please Enter Correct Credentials!");

}

else

label.setText("Please Enter Correct Credentials!");

}

else

label.setText("Please Enter Username and Password First !");

//REST OF THE WORk

}

public void forgetPasswordButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ForgetPassword.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ForgetPasswordController controller = loader.getController();

controller.setParameter("AdminLogin.fxml");

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

Parent tableViewParent = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@Override

public void initialize(URL url, ResourceBundle rb) {

try

{

getCreditials();

}

catch(Exception e)

{

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

}

// TODO

msgIcon = new ImageView("id.png");

lockIcon = new ImageView("lock.png");

close = new ImageView("close.png");

logo = new ImageView("logo1.png");

logo2 = new ImageView("adminLogo.png");

}

}

## Admin Profile Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AdminProfileController implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private ImageView welcome;

@FXML private ImageView customer;

@FXML private ImageView employee;

@FXML private ImageView earning;

@FXML private ImageView account;

@FXML private Label label;

String empID = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("Select first\_name, last\_name from employee where empID = '"+empID+"';");

rs.next();

label.setText(rs.getString(1) + " " + rs.getString(2));

}

@FXML

public void customerImagePressed(MouseEvent event) throws IOException, SQLException // this function is call when home image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Customer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerController controller = loader.getController();

controller.setParameter(empID, "AdminProfile.fxml");

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void customerButtonPressed(ActionEvent event) throws IOException, SQLException // this function is call when acount Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Customer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerController controller = loader.getController();

controller.setParameter(empID, "AdminProfile.fxml");

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void employeeImagePressed(MouseEvent event) throws IOException, SQLException // this function is call when home image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AdminEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void employeeButtonPressed(ActionEvent event) throws IOException, SQLException // this function is call when acount Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AdminEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void earningImagePressed(MouseEvent event) throws IOException, SQLException // this function is call when home image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Earning.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

EarningController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void earningButtonPressed(ActionEvent event) throws IOException, SQLException // this function is call when acount Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Earning.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

EarningController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void accountImagePressed(MouseEvent event) throws IOException, SQLException // this function is call when home image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminAccount.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AdminAccountController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void accountButtonPressed(ActionEvent event) throws IOException, SQLException // this function is call when acount Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminAccount.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AdminAccountController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("admin2.png");

welcome = new ImageView("welcome3.png");

earning = new ImageView("Earning.png");

account = new ImageView("account.png");

customer = new ImageView("customerR.png");

employee = new ImageView("employeeR.png");

}

}

## All houses Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AllHousesController implements Initializable

{

@FXML private ImageView logo;

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox branch;

@FXML private ComboBox category;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void showResultButtonPushed(ActionEvent event) throws IOException, SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(category.getValue().toString().equals("ALL"))

{

ResultSet rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location where branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type not like 'roomsOnly%' and house.houseID not in (select houseID from house\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(1);

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(houseNameS, rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), "AVAILABLE"));

}

rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location where branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type not like 'roomsOnly%' and house.houseID in (select houseID from house\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(1);

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(houseNameS, rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), "BOOKED"));

}

}

else

{

ResultSet rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name from house, branch, city, location, house\_booking where branch.branch\_name = '"+ branch.getValue().toString() +"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and huose\_Type = '"+category.getValue().toString()+"' and house.houseID not in (select house\_booking.houseID from house\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(1);

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(houseNameS, rs.getString(2), rs.getString(3), rs.getString(4), category.getValue().toString(), "AVAILABLE"));

}

rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name from house, branch, city, location, house\_booking where branch.branch\_name = '"+ branch.getValue().toString() +"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and huose\_Type = '"+category.getValue().toString()+"' and house.houseID in (select house\_booking.houseID from house\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(1);

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(houseNameS, rs.getString(2), rs.getString(3), rs.getString(4), category.getValue().toString(), "BOOKED"));

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("HouseMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb){

// TODO

branch.getItems().addAll("ISB Branch 1", "ISB Branch 2", "PEW Branch 1", "PEW Branch 2", "LHR Branch", "KHI Branch");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

String branchID = "";

String branchName = "";

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branchID from employee where empID = '"+ empID +"';");

while(rs.next())

{

branchID = rs.getString(1);

}

rs = stmt.executeQuery("select branch\_name from branch where branchID = '" + branchID + "';");

while(rs.next())

{

branchName = rs.getString(1);

}

}

catch(SQLException e)

{

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

}

branch.setValue(branchName);

category.setValue("ALL");

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

logo = new ImageView("logo1.png");

}

}

## All Room Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AllRoomController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox branch;

@FXML private ComboBox category;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void showResultButtonPushed(ActionEvent event) throws IOException, SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(category.getValue().toString().equals("ALL"))

{

ResultSet rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type like 'roomsOnly%' and room.roomID not in (select room\_booking.roomID from room\_booking);");

while(rs.next())

{

String roomNumberS = rs.getString(1);

System.out.println("House Name from DATA BASE : " + roomNumberS);

list.add(new HouseModalTable(roomNumberS, "House No. " + rs.getString(2), rs.getString(3), rs.getString(4) + ", " + rs.getString(5), category.getValue().toString(), "AVAILABLE"));

}

rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type like 'roomsOnly%' and room.roomID in (select room\_booking.roomID from room\_booking);");

while(rs.next())

{

String roomNumberS = rs.getString(1);

System.out.println("House Name from DATA BASE : " + roomNumberS);

list.add(new HouseModalTable(roomNumberS, "House No. " + rs.getString(2), rs.getString(3), rs.getString(4) + ", " + rs.getString(5), category.getValue().toString(), "BOOKED"));

}

}

else

{

ResultSet rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type = 'roomsOnly"+category.getValue().toString()+"' and room.roomID not in (select room\_booking.roomID from room\_booking);");

while(rs.next())

{

String roomNumberS = rs.getString(1);

System.out.println("House Name from DATA BASE : " + roomNumberS);

list.add(new HouseModalTable(roomNumberS, "House No. " + rs.getString(2), rs.getString(3), rs.getString(4) + ", " + rs.getString(5), category.getValue().toString(), "AVAILABLE"));

}

rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type = 'roomsOnly"+category.getValue().toString()+"' and room.roomID in (select room\_booking.roomID from room\_booking);");

while(rs.next())

{

String roomNumberS = rs.getString(1);

System.out.println("House Name from DATA BASE : " + roomNumberS);

list.add(new HouseModalTable(roomNumberS, "House No. " + rs.getString(2), rs.getString(3), rs.getString(4) + ", " + rs.getString(5), category.getValue().toString(), "BOOKED"));

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RoomMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb){

// TODO

branch.getItems().addAll("ISB Branch 1", "ISB Branch 2", "PEW Branch 1", "PEW Branch 2", "LHR Branch", "KHI Branch");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

String branchID = "";

String branchName = "";

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branchID from employee where empID = '"+ empID +"';");

while(rs.next())

{

branchID = rs.getString(1);

}

rs = stmt.executeQuery("select branch\_name from branch where branchID = '" + branchID + "';");

while(rs.next())

{

branchName = rs.getString(1);

}

System.out.println(branchName);

}

catch(SQLException e)

{

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

}

branch.setValue(branchName);

category.setValue("ALL");

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

logo = new ImageView("logo1.png");

}

}

## Available Houses 2:

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AvailableHouses2Controller implements Initializable {

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> srNoC;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox status;

@FXML private ComboBox category;

String usernameOfLoggedInPerson;

String branchNameFromPrevScene = "";

public void setParameter(String person, HouseModalTable hmt)

{

usernameOfLoggedInPerson = person;

branchNameFromPrevScene = hmt.getBranchName();

System.out.println("Branch Name from Second : " + branchNameFromPrevScene);

bothCombBoxUpdated();

}

public void bothCombBoxUpdated()

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

//PreparedStatement stmt = conn.prepareStatement();

ResultSet rs;

ResultSet rs2;

if(status.getValue().equals("AVAILABLE"))

{

if(category.getValue().equals("LUXURY"))

{

rs = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Luxury') and house.houseID not in (select houseID from house\_booking);");

int counter = 1;

while(rs.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs.getString(3), rs.getString(4), rs.getString(2), "AVAILABLE"));

counter++;

}

}

else if(category.getValue().equals("ECONOMY"))

{

rs = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Economy') and house.houseID not in (select houseID from house\_booking);");

int counter = 1;

while(rs.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs.getString(3), rs.getString(4), rs.getString(2), "AVAILABLE"));

counter++;

}

}

else

{

rs = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Luxury' or huose\_Type = 'Economy') and house.houseID not in (select houseID from house\_booking);");

int counter = 1;

while(rs.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs.getString(3), rs.getString(4), rs.getString(2), "AVAILABLE"));

counter++;

}

}

}

else if(status.getValue().equals("BOOKED"))

{

if(category.getValue().equals("LUXURY"))

{

rs = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Luxury') and house.houseID in (select houseID from house\_booking);");

int counter = 1;

while(rs.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs.getString(3), rs.getString(4), rs.getString(2), "BOOKED"));

counter++;

}

}

else if(category.getValue().equals("ECONOMY"))

{

rs = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Economy') and house.houseID in (select houseID from house\_booking);");

int counter = 1;

while(rs.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs.getString(3), rs.getString(4), rs.getString(2), "BOOKED"));

counter++;

}

}

else

{

rs = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Luxury' or huose\_Type = 'Economy') and house.houseID in (select houseID from house\_booking);");

int counter = 1;

while(rs.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs.getString(3), rs.getString(4), rs.getString(2), "BOOKED"));

counter++;

}

}

}

else

{

if(category.getValue().equals("LUXURY"))

{

rs = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Luxury') and house.houseID not in (select houseID from house\_booking);");

int counter = 1;

while(rs.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs.getString(3), rs.getString(4), rs.getString(2), "AVAILABLE"));

counter++;

}

rs2 = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Luxury') and house.houseID in (select houseID from house\_booking);");

while(rs2.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs2.getString(3), rs2.getString(4), rs2.getString(2), "BOOKED"));

counter++;

}

}

else if(category.getValue().equals("ECONOMY"))

{

rs = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Economy') and house.houseID not in (select houseID from house\_booking);");

int counter = 1;

while(rs.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs.getString(3), rs.getString(4), rs.getString(2), "AVAILABLE"));

counter++;

}

rs2 = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Economy') and house.houseID in (select houseID from house\_booking);");

while(rs2.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs2.getString(3), rs2.getString(4), rs2.getString(2), "BOOKED"));

counter++;

}

}

else

{

rs = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Luxury' or huose\_Type = 'Economy') and house.houseID not in (select houseID from house\_booking);");

int counter = 1;

while(rs.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs.getString(3), rs.getString(4), rs.getString(2), "AVAILABLE"));

counter++;

}

rs2 = stmt.executeQuery("select branch\_name, huose\_Type, location, city\_name from house, branch, location, city where house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'Luxury' or huose\_Type = 'Economy') and house.houseID in (select houseID from house\_booking);");

while(rs2.next())

{

String houseName = "House No. " + counter;

list.add(new HouseModalTable(Integer.toString(counter), houseName, rs2.getString(3), rs2.getString(4), rs2.getString(2), "BOOKED"));

counter++;

}

}

}

}

catch(SQLException e)

{

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

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableHousesController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

status.getItems().addAll("ALL", "BOOKED", "AVAILABLE");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

status.setValue("AVAILABLE");

category.setValue("ALL");

srNoC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

}

}

## Available Houses Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AvailableHousesController implements Initializable {

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> srNoC;

@FXML private TableColumn<HouseModalTable, String> branchNameC;

@FXML private Button detailedView;

@FXML private ComboBox city;

@FXML private Label label;

String usernameOfLoggedInPerson = "";

String branchNameForNextScreen = "";

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

}

public void uponComboSelection()

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(city.getValue().equals("Islamabad"))

{

ResultSet rs = stmt.executeQuery("select branch\_name from branch, location, city where branch.locationID = location.locationID and location.cityID = city.cityID and city\_name = 'Islamabad';");

int counter = 1;

while(rs.next())

{

branchNameForNextScreen = rs.getString(1);

System.out.println(branchNameForNextScreen);

list.add(new HouseModalTable(Integer.toString(counter), branchNameForNextScreen));

counter++;

}

}

else if(city.getValue().equals("Karachi"))

{

ResultSet rs = stmt.executeQuery("select branch\_name from branch, location, city where branch.locationID = location.locationID and location.cityID = city.cityID and city\_name = 'Karachi';");

int counter = 1;

while(rs.next())

{

branchNameForNextScreen = rs.getString(1);

System.out.println(branchNameForNextScreen);

list.add(new HouseModalTable(Integer.toString(counter), branchNameForNextScreen));

counter++;

}

}

else if(city.getValue().equals("Lahore"))

{

ResultSet rs = stmt.executeQuery("select branch\_name from branch, location, city where branch.locationID = location.locationID and location.cityID = city.cityID and city\_name = 'Lahore';");

int counter = 1;

while(rs.next())

{

branchNameForNextScreen = rs.getString(1);

System.out.println(branchNameForNextScreen);

list.add(new HouseModalTable(Integer.toString(counter), branchNameForNextScreen));

counter++;

}

}

else if(city.getValue().equals("Peshawar"))

{

ResultSet rs = stmt.executeQuery("select branch\_name from branch, location, city where branch.locationID = location.locationID and location.cityID = city.cityID and city\_name = 'Peshawar';");

int counter = 1;

while(rs.next())

{

branchNameForNextScreen = rs.getString(1);

System.out.println(branchNameForNextScreen);

list.add(new HouseModalTable(Integer.toString(counter), branchNameForNextScreen));

counter++;

}

}

}

catch(Exception e)

{

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

}

tableView.setItems(list);

}

//this method will move the make the detailed view button available

public void userClickOnTable()

{

this.detailedView.setDisable(false);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("House.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void seeDetailedButtonPushed(ActionEvent event) throws IOException

{

if(!branchNameForNextScreen.equals(""))

{

System.out.println(branchNameForNextScreen);

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableHouses2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableHouses2Controller controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson, tableView.getSelectionModel().getSelectedItem());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("Please Select A Row First");

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

detailedView.setDisable(true);

city.getItems().addAll("Islamabad", "Lahore", "Karachi", "Peshawar");

city.setValue("Islamabad");

srNoC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

uponComboSelection();

}

}

## Available Houses for main Houses Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AvailableHousesFromMainHousesController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox branch;

@FXML private ComboBox category;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void showResultButtonPushed(ActionEvent event) throws IOException, SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(category.getValue().toString().equals("ALL"))

{

ResultSet rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location where branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type not like 'roomsOnly%' and house.houseID not in (select houseID from house\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(1);

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(houseNameS, rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), "AVAILABLE"));

}

}

else

{

ResultSet rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name from house, branch, city, location, house\_booking where branch.branch\_name = '"+ branch.getValue().toString() +"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and huose\_Type = '"+category.getValue().toString()+"' and house.houseID not in (select house\_booking.houseID from house\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(1);

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(houseNameS, rs.getString(2), rs.getString(3), rs.getString(4), category.getValue().toString(), "AVAILABLE"));

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("HouseMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb){

// TODO

branch.getItems().addAll("ISB Branch 1", "ISB Branch 2", "PEW Branch 1", "PEW Branch 2", "LHR Branch", "KHI Branch");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

String branchID = "";

String branchName = "";

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branchID from employee where empID = '"+ empID +"';");

while(rs.next())

{

branchID = rs.getString(1);

}

rs = stmt.executeQuery("select branch\_name from branch where branchID = '" + branchID + "';");

while(rs.next())

{

branchName = rs.getString(1);

}

}

catch(SQLException e)

{

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

}

branch.setValue(branchName);

category.setValue("ALL");

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

logo = new ImageView("logo1.png");

}

}

## Available Room 2 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AvailableRoom2Controller implements Initializable {

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> srNoC;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox status;

@FXML private ComboBox category;

String usernameOfLoggedInPerson;

String branchNameFromPrevScene = "";

public void setParameter(String person, HouseModalTable hmt)

{

usernameOfLoggedInPerson = person;

branchNameFromPrevScene = hmt.getBranchName();

System.out.println("Branch Name from Second : " + branchNameFromPrevScene);

bothCombBoxUpdated();

}

public void bothCombBoxUpdated()

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

//PreparedStatement stmt = conn.prepareStatement();

ResultSet rs;

ResultSet rs2;

if(status.getValue().equals("AVAILABLE"))

{

if(category.getValue().equals("LUXURY"))

{

rs = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyLuxury') and room.roomID not in (select roomID from room\_booking);");

//int counter = 1;

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "AVAILABLE"));

}

}

else if(category.getValue().equals("ECONOMY"))

{

rs = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyEconomy') and room.roomID not in (select roomID from room\_booking);");

//int counter = 1;

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "AVAILABLE"));

}

}

else

{

rs = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyLuxury' or huose\_Type = 'roomsOnlyEconomy') and room.roomID not in (select roomID from room\_booking);");

//int counter = 1;

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "AVAILABLE"));

}

}

}

else if(status.getValue().equals("BOOKED"))

{

if(category.getValue().equals("LUXURY"))

{

rs = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyLuxury') and room.roomID in (select roomID from room\_booking);");

//int counter = 1;

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "BOOKED"));

}

}

else if(category.getValue().equals("ECONOMY"))

{

rs = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyEconomy') and room.roomID in (select roomID from room\_booking);");

//int counter = 1;

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "BOOKED"));

}

}

else

{

rs = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyLuxury' or huose\_Type = 'roomsOnlyEconomy') and room.roomID in (select roomID from room\_booking);");

//int counter = 1;

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "BOOKED"));

}

}

}

else

{

if(category.getValue().equals("LUXURY"))

{

rs = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyLuxury') and room.roomID not in (select roomID from room\_booking);");

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "AVAILABLE"));

}

rs2 = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyLuxury') and room.roomID in (select roomID from room\_booking);");

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "BOOKED"));

}

}

else if(category.getValue().equals("ECONOMY"))

{

rs = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyEconomy') and room.roomID not in (select roomID from room\_booking);");

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "AVAILABLE"));

}

rs2 = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyEconomy') and room.roomID in (select roomID from room\_booking);");

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "BOOKED"));

}

}

else

{

rs = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyLuxury' or huose\_Type = 'roomsOnlyEconomy') and room.roomID not in (select roomID from room\_booking);");

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "AVAILABLE"));

//counter++;

}

rs2 = stmt.executeQuery("select huose\_Type, location, city\_name, house.houseID, room.room\_number from house, branch, location, city, room where room.houseID = house.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and branch\_name = '"+ branchNameFromPrevScene +"' and (huose\_Type = 'roomsOnlyLuxury' or huose\_Type = 'roomsOnlyEconomy') and room.roomID in (select roomID from room\_booking);");

while(rs.next())

{

String houseName = "House No. " + rs.getString(4);

String temp = rs.getString(1);

String roomType = temp.substring(9, temp.length());

list.add(new HouseModalTable(rs.getString(5), houseName, rs.getString(2), rs.getString(3), roomType, "BOOKED"));

//counter++;

}

}

}

}

catch(SQLException e)

{

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

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableRoom.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableRoomController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

status.getItems().addAll("ALL", "BOOKED", "AVAILABLE");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

status.setValue("AVAILABLE");

category.setValue("ALL");

srNoC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

}

}

## Available Room Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AvailableRoomController implements Initializable {

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> srNoC;

@FXML private TableColumn<HouseModalTable, String> branchNameC;

@FXML private Button detailedView;

@FXML private ComboBox city;

@FXML private Label label;

String usernameOfLoggedInPerson = "";

String branchNameForNextScreen = "";

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

}

public void uponComboSelection()

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(city.getValue().equals("Islamabad"))

{

ResultSet rs = stmt.executeQuery("select branch\_name from branch, location, city where branch.locationID = location.locationID and location.cityID = city.cityID and city\_name = 'Islamabad';");

int counter = 1;

while(rs.next())

{

branchNameForNextScreen = rs.getString(1);

System.out.println(branchNameForNextScreen);

list.add(new HouseModalTable(Integer.toString(counter), branchNameForNextScreen));

counter++;

}

}

else if(city.getValue().equals("Karachi"))

{

ResultSet rs = stmt.executeQuery("select branch\_name from branch, location, city where branch.locationID = location.locationID and location.cityID = city.cityID and city\_name = 'Karachi';");

int counter = 1;

while(rs.next())

{

branchNameForNextScreen = rs.getString(1);

System.out.println(branchNameForNextScreen);

list.add(new HouseModalTable(Integer.toString(counter), branchNameForNextScreen));

counter++;

}

}

else if(city.getValue().equals("Lahore"))

{

ResultSet rs = stmt.executeQuery("select branch\_name from branch, location, city where branch.locationID = location.locationID and location.cityID = city.cityID and city\_name = 'Lahore';");

int counter = 1;

while(rs.next())

{

branchNameForNextScreen = rs.getString(1);

System.out.println(branchNameForNextScreen);

list.add(new HouseModalTable(Integer.toString(counter), branchNameForNextScreen));

counter++;

}

}

else if(city.getValue().equals("Peshawar"))

{

ResultSet rs = stmt.executeQuery("select branch\_name from branch, location, city where branch.locationID = location.locationID and location.cityID = city.cityID and city\_name = 'Peshawar';");

int counter = 1;

while(rs.next())

{

branchNameForNextScreen = rs.getString(1);

System.out.println(branchNameForNextScreen);

list.add(new HouseModalTable(Integer.toString(counter), branchNameForNextScreen));

counter++;

}

}

}

catch(Exception e)

{

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

}

tableView.setItems(list);

}

//this method will move the make the detailed view button available

public void userClickOnTable()

{

this.detailedView.setDisable(false);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Room.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void seeDetailedButtonPushed(ActionEvent event) throws IOException

{

if(!branchNameForNextScreen.equals(""))

{

System.out.println(branchNameForNextScreen);

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableRoom2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableRoom2Controller controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson, tableView.getSelectionModel().getSelectedItem());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("Please Select A Row First");

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

detailedView.setDisable(true);

city.getItems().addAll("Islamabad", "Lahore", "Karachi", "Peshawar");

city.setValue("Islamabad");

srNoC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

uponComboSelection();

}

}

## Available Rooms Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class AvailableRoomsController implements Initializable

{

@FXML private ImageView logo;

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox branch;

@FXML private ComboBox category;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void showResultButtonPushed(ActionEvent event) throws IOException, SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(category.getValue().toString().equals("ALL"))

{

ResultSet rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type like 'roomsOnly%' and room.roomID not in (select room\_booking.roomID from room\_booking);");

while(rs.next())

{

String roomNumberS = rs.getString(1);

System.out.println("House Name from DATA BASE : " + roomNumberS);

list.add(new HouseModalTable(roomNumberS, "House No. " + rs.getString(2), rs.getString(3), rs.getString(4) + ", " + rs.getString(5), category.getValue().toString(), "AVAILABLE"));

}

}

else

{

ResultSet rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type = 'roomsOnly"+category.getValue().toString()+"' and room.roomID not in (select room\_booking.roomID from room\_booking);");

while(rs.next())

{

String roomNumberS = rs.getString(1);

System.out.println("House Name from DATA BASE : " + roomNumberS);

list.add(new HouseModalTable(roomNumberS, "House No. " + rs.getString(2), rs.getString(3), rs.getString(4) + ", " + rs.getString(5), category.getValue().toString(), "AVAILABLE"));

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RoomMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb)

{

// TODO

branch.getItems().addAll("ISB Branch 1", "ISB Branch 2", "PEW Branch 1", "PEW Branch 2", "LHR Branch", "KHI Branch");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

String branchID = "";

String branchName = "";

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branchID from employee where empID = '"+ empID +"';");

while(rs.next())

{

branchID = rs.getString(1);

}

rs = stmt.executeQuery("select branch\_name from branch where branchID = '" + branchID + "';");

while(rs.next())

{

branchName = rs.getString(1);

}

System.out.println(branchName);

}

catch(SQLException e)

{

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

}

branch.setValue(branchName);

category.setValue("ALL");

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

logo = new ImageView("logo1.png");

}

}

## Bills Controller:

package guesthousemanagement;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.PrintWriter;

import java.io.StringWriter;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.TextArea;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

public class BillsController implements Initializable

{

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private ImageView currentBill;

@FXML private ImageView billRecord;

String usernameOfLoggedInPerson;

String counts = "";

String rate = "";

String serviceName = "";

int total = 0;

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select service\_name, rate, counts from bills, services, (select customerID from customer where email = '" + usernameOfLoggedInPerson + "')temps where temps.customerID = bills.customerID and services.serviceID = bills.serviceID;");

while(rs.next())

{

serviceName = serviceName + rs.getString(1) + "#";

int temp1 = Integer.parseInt(rs.getString(2));

int temp2 = Integer.parseInt(rs.getString(3));

rate = rate + temp1 + "#";

counts = counts + temp2 + "#";

total = total + (temp1 \* temp2);

}

System.out.println(serviceName + "\n" + rate + "\n" + counts);

}

catch(Exception e)

{

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

}

}

public void currentBillButtonPushed()

{

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("CURRENT BILLS");

a1.setContentText("Your Current Bills : " + total);

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

}

public void billRecordButtonPushed()

{

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("BILLING RECORD");

a1.setContentText("Total Bill : " + total);

a1.setHeaderText("For Complete Record, Select Show Details!"); //you can also set header

String temp = "\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\tSERVICE NAME \t\t SERVICE RATE \t\t NUMBERS OF REQ.\n\t-----------------------------------------------------------------------------\n";

String serviceNameA[] = serviceName.split("#");

String rateA[] = rate.split("#");

String countsA[] = counts.split("#");

for(int i = 0; i < serviceNameA.length; i++)

{

if(serviceNameA[i].length() >= 10)

temp = temp + "\t" + serviceNameA[i] + "\t\t\t\t" + rateA[i] + "\t\t\t\t\t" + countsA[i] + "\n";

else if(serviceNameA[i].length() < 10 && serviceNameA[i].length() > 6)

temp = temp + "\t" + serviceNameA[i] + "\t\t\t\t\t" + rateA[i] + "\t\t\t\t\t" + countsA[i] + "\n";

else

temp = temp + "\t" + serviceNameA[i] + "\t\t\t\t\t\t" + rateA[i] + "\t\t\t\t\t" + countsA[i] + "\n";

}

temp = temp + "\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

// laundryman

// cook

// repairer

// sweeper

// housekeeping

TextArea ta = new TextArea(temp);

ta.setEditable(false);

a1.getDialogPane().setExpandableContent(ta);

a1.showAndWait();

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerProfileController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb)

{

logo = new ImageView("logo1.png");

logo2 = new ImageView("bill.png");

currentBill = new ImageView("currentBill.png");

billRecord = new ImageView("billRecord.png");

// TODO

}

}

## Book Houses Controller 2:

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Optional;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.ButtonType;

import javafx.scene.control.ComboBox;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class BookHouses2Controller implements Initializable {

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private Button bookButton;

@FXML private Label label;

@FXML private DatePicker date;

String usernameOfLoggedInPerson = "";

String branchNameFromPrevScene = "";

String categoryName = "";

String customerID = "";

int houseNo = 0;

int check = 0;

public void setParameter(String person, String branchNameFromPrevScene, String categoryName)

{

this.usernameOfLoggedInPerson = person;

this.branchNameFromPrevScene = branchNameFromPrevScene;

this.categoryName = categoryName;

setTableData();

}

public void userClickOnTable()

{

check++;

if(check > 1)

this.bookButton.setDisable(false);

}

public void userClickOnDaTe()

{

check++;

if(check > 1)

this.bookButton.setDisable(false);

}

public void setTableData()

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name from house, branch, city, location, house\_booking where branch.branch\_name = '"+ branchNameFromPrevScene +"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and huose\_Type = '" + categoryName + "' and house.houseID not in (select house\_booking.houseID from house\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(1);

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(houseNameS, rs.getString(2), rs.getString(3), rs.getString(4), categoryName, "AVAILABLE"));

}

}

catch(Exception e)

{

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

}

tableView.setItems(list);

}

public void addDataInRecord()

{

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int recordUpdate = stmt.executeUpdate("INSERT INTO `mydb`.`customer\_record` (`customerID`, `booking`, `booking\_date`, `leaving\_date`, `bills`) VALUES ('" + customerID + "', 'h" + houseNo + "', '" + Checks.getCurrentdate() + "', '" + date.getValue().toString() + "', '0');");

}

catch(Exception e)

{

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

}

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BookHousesController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void bookButtonPushed(ActionEvent event) throws IOException

{

if(Checks.checkLeavingDate(date.getValue().toString()))

{

HouseModalTable hmt = tableView.getSelectionModel().getSelectedItem();

Alert a1 = new Alert(Alert.AlertType.CONFIRMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a1.setTitle("CONFIRMATION");

a1.setContentText("ARE YOU SURE YOU WANT TO BOOK THIS HOUSE ? ");

//a1.setHeaderText("Basic ALERT 2 "); //you can also set header

Optional<ButtonType> result = a1.showAndWait();

if(result.get() == ButtonType.OK) //type can also be "OK" as well

{

System.out.println("OKAY");

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select customerID from customer where email = '"+ usernameOfLoggedInPerson +"';");

while(rs.next())

{

customerID = rs.getString(1);

}

String temp = hmt.getSrNo();

houseNo = Integer.parseInt(temp.substring(10, temp.length()));

int recordUpdate = stmt.executeUpdate("INSERT INTO `mydb`.`house\_booking` (`houseID`, `customerID`) VALUES ('" + houseNo + "', ' " + customerID + " ');");

addDataInRecord();

Alert a2 = new Alert(Alert.AlertType.INFORMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a2.setTitle("INFORMATION");

a2.setContentText("Your House has Successfully Booked ! ");

a2.setHeaderText(null); //you can also set header

a2.showAndWait();

}

catch(Exception e)

{

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

}

}

else

{

System.out.println("Cancel");

}

}

else

label.setText("Incorrect Leaving Date !!!");

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

this.bookButton.setDisable(true);

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

setTableData();

}

}

## Book Houses Controller 2:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.lang.reflect.InvocationTargetException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ChoiceBox;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class BookHousesController implements Initializable {

@FXML private ComboBox branchCombo;

@FXML private ComboBox categoryCombo;

@FXML private Label label;

@FXML private Button nextButton;

String usernameOfLoggedInPerson;

String branchName = "";

String cityName = "";

String categoryName = "";

//String cityAS[];

String categoryAS[];

String branchAS[];

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

}

public void userClickOnTable()

{

this.nextButton.setDisable(false);

}

public void undisableButton()

{

if(!branchCombo.getValue().toString().equals("Select Branch") && !categoryCombo.getValue().toString().equals("Select Category"))

nextButton.setDisable(false);

}

public void nextButtonPushed(ActionEvent event) throws IOException

{

if(!branchCombo.getValue().toString().equals("Select Branch") && !categoryCombo.getValue().toString().equals("Select Category"))

{

branchName = branchCombo.getValue().toString();

categoryName = categoryCombo.getValue().toString();

System.out.println(branchName);

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookHouses2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BookHouses2Controller controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson, branchName, categoryName);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("Please Select A Row First");

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("House.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

categoryCombo.getItems().addAll("LUXURY", "ECONOMY");

categoryCombo.setValue("Select Category");

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branch\_name from branch;");

while(rs.next())

branchCombo.getItems().addAll(rs.getString(1));

branchCombo.setValue("Select Branch");

}

catch(Exception e)

{

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

}

this.nextButton.setDisable(true);

}

}

## Book Room 2 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Optional;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.ButtonType;

import javafx.scene.control.ComboBox;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class BookRoom2Controller implements Initializable {

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private Button bookButton;

@FXML private DatePicker date;

@FXML private Label label;

String usernameOfLoggedInPerson = "";

String branchNameFromPrevScene = "";

String categoryName = "";

String customerID = "";

int roomIDForCustomerRecord = 0;

int check = 0;

public void setParameter(String person, String branchNameFromPrevScene, String categoryN)

{

this.usernameOfLoggedInPerson = person;

this.branchNameFromPrevScene = branchNameFromPrevScene;

this.categoryName = categoryN;

System.out.println("\nPerson : " + person + "\nbranchName : " + branchNameFromPrevScene + "\nCategory : " + categoryName);

setTableData();

}

public void userClickOnTable()

{

check++;

if(check > 1)

this.bookButton.setDisable(false);

}

public void userClickOnDaTe()

{

check++;

if(check > 1)

this.bookButton.setDisable(false);

}

public void setTableData()

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select room.room\_number, house.houseID, location, city\_name from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branchNameFromPrevScene+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and huose\_Type = '"+categoryName+"' and room.roomID not in (select room\_booking.roomID from room\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(2);

String categoryS = categoryName.substring(9, categoryName.length());

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(rs.getString(1), houseNameS, rs.getString(3), rs.getString(4), categoryS, "AVAILABLE"));

}

}

catch(Exception e)

{

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

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookRoom.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BookRoomController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void addDataInRecord()

{

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

System.out.println("I am in Add Data Record Now");

int recordUpdate = stmt.executeUpdate("INSERT INTO `mydb`.`customer\_record` (`customerID`, `booking`, `booking\_date`, `leaving\_date`, `bills`) VALUES ('" + customerID + "', 'r" + roomIDForCustomerRecord + "', '" + Checks.getCurrentdate() + "', '" + date.getValue().toString() + "', '0');");

}

catch(Exception e)

{

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

}

}

public void bookButtonPushed(ActionEvent event) throws IOException

{

if(Checks.checkLeavingDate(date.getValue().toString()))

{

HouseModalTable hmt = tableView.getSelectionModel().getSelectedItem();

Alert a1 = new Alert(Alert.AlertType.CONFIRMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a1.setTitle("CONFIRMATION");

a1.setContentText("ARE YOU SURE YOU WANT TO BOOK THIS HOUSE ? ");

//a1.setHeaderText("Basic ALERT 2 "); //you can also set header

Optional<ButtonType> result = a1.showAndWait();

if(result.get() == ButtonType.OK) //type can also be "OK" as well

{

System.out.println("OKAY");

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select customerID from customer where email = '"+ usernameOfLoggedInPerson +"';");

customerID = "";

while(rs.next())

{

customerID = rs.getString(1);

}

String roomNo = hmt.getSrNo();

String temp = hmt.getBranchName(); // in this case this is House Name

String houseID = temp.substring(10, temp.length());

System.out.println("I am in Book Button Pushed 1");

System.out.println("\nPerson : " + customerID + "\nbranchName : " + branchNameFromPrevScene + "\nCategory : " + categoryName);

rs = stmt.executeQuery("select room.roomID from room, house where house.houseID = room.houseID and house.houseID = '"+ houseID +"' and huose\_Type = '"+ categoryName +"' and room.room\_number = '"+ roomNo +"';");

String roomID = "";

while(rs.next())

roomID = rs.getString(1);

roomIDForCustomerRecord = Integer.parseInt(roomID);

System.out.println("I am in Book Button Push 2");

int recordUpdate = stmt.executeUpdate("INSERT INTO `mydb`.`room\_booking` (`roomID`, `customerID`) VALUES ('" + roomIDForCustomerRecord + "', ' " + customerID + " ');");

addDataInRecord();

Alert a2 = new Alert(Alert.AlertType.INFORMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a2.setTitle("INFORMATION");

a2.setContentText("Your House has Successfully Booked ! ");

a2.setHeaderText(null); //you can also set header

a2.showAndWait();

}

catch(Exception e)

{

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

}

}

else

{

System.out.println("Cancel");

}

}

else

label.setText("Incorrect Leaving Date !!!");

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

this.bookButton.setDisable(true);

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

setTableData();

}

}

## Book Room Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class BookRoomController implements Initializable {

@FXML private ComboBox branchCombo;

@FXML private ComboBox categoryCombo;

@FXML private Label label;

@FXML private Button nextButton;

String usernameOfLoggedInPerson;

String branchName = "";

String cityName = "";

String categoryName = "";

//String cityAS[];

String categoryAS[];

String branchAS[];

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

}

public void userClickOnTable()

{

this.nextButton.setDisable(false);

}

public void undisableButton()

{

if(!branchCombo.getValue().toString().equals("Select Branch") && !categoryCombo.getValue().toString().equals("Select Category"))

nextButton.setDisable(false);

}

public void nextButtonPushed(ActionEvent event) throws IOException

{

if(!branchCombo.getValue().toString().equals("Select Branch") && !categoryCombo.getValue().toString().equals("Select Category"))

{

branchName = branchCombo.getValue().toString();

if(categoryCombo.getValue().toString().equals("LUXURY"))

categoryName = "roomsOnlyLuxury";

else

categoryName = "roomsOnlyEconomy";

System.out.println(branchName);

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookRoom2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BookRoom2Controller controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson, branchName, categoryName);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("Please Select A Row First");

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Room.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

categoryCombo.getItems().addAll("LUXURY", "ECONOMY");

categoryCombo.setValue("Select Category");

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branch\_name from branch;");

while(rs.next())

branchCombo.getItems().addAll(rs.getString(1));

branchCombo.setValue("Select Branch");

}

catch(Exception e)

{

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

}

this.nextButton.setDisable(true);

}

}

## Booked Houses Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class BookedHousesController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox branch;

@FXML private ComboBox category;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void showResultButtonPushed(ActionEvent event) throws IOException, SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(category.getValue().toString().equals("ALL"))

{

ResultSet rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location where branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type not like 'roomsOnly%' and house.houseID in (select houseID from house\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(1);

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(houseNameS, rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), "BOOKED"));

}

}

else

{

ResultSet rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name from house, branch, city, location, house\_booking where branch.branch\_name = '"+ branch.getValue().toString() +"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and huose\_Type = '"+category.getValue().toString()+"' and house.houseID in (select house\_booking.houseID from house\_booking);");

while(rs.next())

{

String houseNameS = "House No. " + rs.getString(1);

System.out.println("House Name from DATA BASE : " + houseNameS);

list.add(new HouseModalTable(houseNameS, rs.getString(2), rs.getString(3), rs.getString(4), category.getValue().toString(), "BOOKED"));

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("HouseMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb){

// TODO

branch.getItems().addAll("ISB Branch 1", "ISB Branch 2", "PEW Branch 1", "PEW Branch 2", "LHR Branch", "KHI Branch");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

String branchID = "";

String branchName = "";

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branchID from employee where empID = '"+ empID +"';");

while(rs.next())

{

branchID = rs.getString(1);

}

rs = stmt.executeQuery("select branch\_name from branch where branchID = '" + branchID + "';");

while(rs.next())

{

branchName = rs.getString(1);

}

}

catch(SQLException e)

{

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

}

branch.setValue(branchName);

category.setValue("ALL");

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

logo = new ImageView("logo1.png");

}

}

## Booked Rooms:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class BookedRoomsController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox branch;

@FXML private ComboBox category;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void showResultButtonPushed(ActionEvent event) throws IOException, SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(category.getValue().toString().equals("ALL"))

{

ResultSet rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type like 'roomsOnly%' and room.roomID in (select room\_booking.roomID from room\_booking);");

while(rs.next())

{

String roomNumberS = rs.getString(1);

System.out.println("House Name from DATA BASE : " + roomNumberS);

list.add(new HouseModalTable(roomNumberS, "House No. " + rs.getString(2), rs.getString(3), rs.getString(4) + ", " + rs.getString(5), category.getValue().toString(), "BOOKED"));

}

}

else

{

ResultSet rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type = 'roomsOnly"+category.getValue().toString()+"' and room.roomID in (select room\_booking.roomID from room\_booking);");

while(rs.next())

{

String roomNumberS = rs.getString(1);

System.out.println("House Name from DATA BASE : " + roomNumberS);

list.add(new HouseModalTable(roomNumberS, "House No. " + rs.getString(2), rs.getString(3), rs.getString(4) + ", " + rs.getString(5), category.getValue().toString(), "BOOKED"));

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RoomMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb)

{

// TODO

branch.getItems().addAll("ISB Branch 1", "ISB Branch 2", "PEW Branch 1", "PEW Branch 2", "LHR Branch", "KHI Branch");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

String branchID = "";

String branchName = "";

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branchID from employee where empID = '"+ empID +"';");

while(rs.next())

{

branchID = rs.getString(1);

}

rs = stmt.executeQuery("select branch\_name from branch where branchID = '" + branchID + "';");

while(rs.next())

{

branchName = rs.getString(1);

}

System.out.println(branchName);

}

catch(SQLException e)

{

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

}

branch.setValue(branchName);

category.setValue("ALL");

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

logo = new ImageView("logo1.png");

}

}

## Change Salary Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ChangeSalaryController implements Initializable {

@FXML private ImageView logo;

@FXML private TextField cnic;

@FXML private Label label;

@FXML private TextField salary;

@FXML private Label label2;

@FXML private Label label3;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void seePreviousSalaryButtonPushed() throws SQLException

{

if(!cnic.getText().equals(""))

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String employeeID = "";

ResultSet rs = stmt.executeQuery("select salary from employee where CNIC = '"+cnic.getText()+"';");

if(rs.next())

{

label2.setText(rs.getString(1) + " Rupees ");

}

else

{

label.setText("Please Enter Valid CNIC !");

}

}

else

label.setText("Please Enter Eny CNIC First !");

}

public void changeSalary(ActionEvent event) throws IOException, SQLException

{

if(!cnic.getText().equals(""))

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String employeeID = "";

ResultSet rs = stmt.executeQuery("select empID from employee where CNIC = '"+cnic.getText()+"';");

if(rs.next())

{

employeeID = rs.getString(1);

String sql = "UPDATE `mydb`.`employee` SET `salary` = '"+salary.getText()+"' WHERE (`empID` = '"+employeeID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

}

else

label.setText("No Such Person Exits in Employe Database or Invalid CNIC !");

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("Successfully Changed ");

a1.setContentText("Employee Salary has been Changed successfully!");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

backButtonPushed(event);

}

else

label.setText("Please Enter Eny CNIC First !");

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

AdminEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Changing Info Controller:

package guesthousemanagement;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.PrintWriter;

import java.io.StringWriter;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.RadioButton;

import javafx.scene.control.TextArea;

import javafx.scene.control.TextField;

import javafx.scene.control.ToggleGroup;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

public class ChangingInfoController implements Initializable

{

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private TextField firstName;

@FXML private TextField lastName;

@FXML private DatePicker dob;

@FXML private TextField cnic;

//For Radio Button

@FXML private RadioButton male;

@FXML private RadioButton female;

@FXML private ToggleGroup favLang; //this is for grouping the upper 2 radio button, so that only one button can be selected at a time

@FXML private TextField mob1;

@FXML private TextField mob2;

@FXML private TextField mob3;

@FXML private TextField address1\_city;

@FXML private TextField address1\_country;

@FXML private TextField address1\_address;

@FXML private TextField address2\_city;

@FXML private TextField address2\_country;

@FXML private TextField address2\_address;

@FXML private TextField address3\_city;

@FXML private TextField address3\_country;

@FXML private TextField address3\_address;

String prevAddress1ID = "";

String prevAddress2ID = "";

String prevAddress3ID = "";

String prevMob1 = "";

String prevMob2 = "";

String prevMob3 = "";

@FXML private Label label;

String usernameOfLoggedInPerson;

String gender = "";

public void setParameter(String s)

{

try

{

usernameOfLoggedInPerson = s;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String employeeID = "";

ResultSet rs = stmt.executeQuery("select \* from customer where email = '"+s+"';");

if(rs.next())

{

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

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

cnic.setText(rs.getString(6));

dob.setValue(Checks.stringToLocalDateConverter(rs.getString(5)));

gender = rs.getString(4);

if(gender.equals("Male"))

favLang.selectToggle(male);

else

favLang.selectToggle(female);

}

int counter = 1;

rs = stmt.executeQuery("select distinct address, city.city\_name, country.country\_name, address.addressID from address, city, country, customer, customer\_address where customer.customerID = customer\_address.customerID and address.addressID = customer\_address.addressID and address.City\_cityID = city.cityID and city.countryID = country.countryID and customer.email = '"+s+"';;");

while(rs.next())

{

if(counter == 1)

{

address1\_address.setText(rs.getString(1));

address1\_city.setText(rs.getString(2));

address1\_country.setText(rs.getString(3));

prevAddress1ID = rs.getString(4);

}

else if(counter == 2)

{

address2\_address.setVisible(true);

address2\_city.setVisible(true);

address2\_country.setVisible(true);

address2\_address.setText(rs.getString(1));

address2\_city.setText(rs.getString(2));

address2\_country.setText(rs.getString(3));

prevAddress2ID = rs.getString(4);

}

else

{

address3\_address.setVisible(true);

address3\_city.setVisible(true);

address3\_country.setVisible(true);

address3\_address.setText(rs.getString(1));

address3\_city.setText(rs.getString(2));

address3\_country.setText(rs.getString(3));

prevAddress3ID = rs.getString(4);

}

counter++;

}

counter = 1;

rs = stmt.executeQuery("select phone\_number from customer natural join customer\_phone where email = '"+s+"';");

while(rs.next())

{

if(counter == 1)

{

mob1.setText(rs.getString(1));

prevMob1 = mob1.getText();

}

else if(counter == 2)

{

mob2.setVisible(true);

mob2.setText(rs.getString(1));

prevMob2 = mob2.getText();

}

else

{

mob3.setVisible(true);

mob3.setText(rs.getString(1));

prevMob3 = mob3.getText();

}

counter++;

}

}

catch(Exception e)

{

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

}

}

public boolean cnicCheck(String s) throws SQLException

{

boolean check = false;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String cnic = "";

ResultSet rs = stmt.executeQuery("select CNIC from customer where email = '"+usernameOfLoggedInPerson+"';");

if(rs.next())

cnic = rs.getString(1);

if(cnic.equals(s))

check = true;

return check;

}

public void updateData(ActionEvent event) throws IOException, SQLException

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(Checks.alphaCheckIgnoreSpace(firstName.getText()) && Checks.alphaCheckIgnoreSpace(lastName.getText()))

{

if(Checks.cnicCheck(cnic.getText()))

{

if(Checks.cnicCheckData(cnic.getText()) || cnicCheck(cnic.getText()))

{

String date = dob.getValue().toString();

String dateData[] = date.split("-");

System.out.println("Year : "+ dateData[0]);

System.out.println("MOnth : "+ dateData[1]);

System.out.println("Day : "+ dateData[2]);

if(Checks.ageCheck(Integer.parseInt(dateData[0]), Integer.parseInt(dateData[1]), Integer.parseInt(dateData[2])))

{

//System.out.println(favLang.getSelectedToggle().toString());

if(favLang.getSelectedToggle().equals(male))

gender = "Male";

else

gender = "Female";

System.out.println(gender);

String sql = "UPDATE `mydb`.`customer` SET `first\_name` = '"+firstName.getText()+"', `last\_name` = '"+lastName.getText()+"', `gender` = '"+gender+"', `DOB` = '"+date+"', `CNIC` = '"+cnic.getText()+"' WHERE (`email` = '"+usernameOfLoggedInPerson+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

if(address1\_address.getText().equals("") || address1\_city.getText().equals("") || address1\_country.getText().equals(""))

label.setText("You Cannot Leave any Address Field Empty!");

else

addressUpdate();

if(mob1.getText().equals(""))

label.setText("You Cannot Leave Mobile Field Empty!");

else

mobileUpdate();

saveAlert(event);

//MObile update cpmes here

}

else

label.setText("Your Age shoudn't be Under 18 !");

}

else

label.setText("Your Entered CNIC is already registered!");

}

else

label.setText("Please Enter CNIC in Format like '12345-1234567-1' !");

}

else

label.setText("First Name should only consist of Alphabets !");

}

public void addressUpdate() throws SQLException

{

System.out.println("=====================================================================================");

System.out.println("=========================== ADDRESSS 1 =============================================");

System.out.println("=====================================================================================");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String customerID = "";

ResultSet rs = stmt.executeQuery("Select customerID from customer where email = '"+usernameOfLoggedInPerson+"';");

if(rs.next())

customerID = rs.getString(1);

String countryID = "";

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address1\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

else

{

String sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+address1\_country.getText()+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address1\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

}

System.out.println("CountryID : " + countryID);

String cityID = "";

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address1\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

String sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+address1\_city.getText()+"', '"+countryID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address1\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

}

System.out.println("City ID : " + cityID);

String addressID = "";

rs = stmt.executeQuery("Select addressID from address where address = '"+address1\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

String sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+address1\_address.getText()+"', '"+cityID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select addressID from address where address = '"+address1\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

}

System.out.println("CustomerID : " +customerID);

System.out.println("AddressID : " +addressID);

System.out.println("Previous Address ID : " + prevAddress1ID);

String sql = "UPDATE `mydb`.`customer\_address` SET `addressID` = '"+addressID+"' WHERE (`addressID` = '"+prevAddress1ID+"') and (`customerID` = '"+customerID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

System.out.println("=====================================================================================");

System.out.println("=========================== ADDRESSS 2 =============================================");

System.out.println("=====================================================================================");

if(address2\_address.isVisible())

{

if(address1\_address.getText().equals("") || address1\_city.getText().equals("") || address1\_country.getText().equals(""))

label.setText("You Cannot Leave Previous Address Field Empty !");

else

{

countryID = "";

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address2\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+address2\_country.getText()+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address2\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

}

System.out.println("Country ID : " + countryID);

cityID = "";

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address2\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+address2\_city.getText()+"', '"+countryID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address2\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

}

System.out.println("City ID : " + cityID);

addressID = "";

rs = stmt.executeQuery("Select addressID from address where address = '"+address2\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+address2\_address.getText()+"', '"+cityID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select addressID from address where address = '"+address2\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

}

System.out.println("CustomerID : " +customerID);

System.out.println("Address ID : " + addressID);

System.out.println("Previous AddressID : " + prevAddress2ID);

if(prevAddress2ID.equals(""))

{

sql = "INSERT INTO `mydb`.`customer\_address` (`addressID`, `customerID`) VALUES ('"+addressID+"', '"+customerID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

}

else

{

sql = "UPDATE `mydb`.`customer\_address` SET `addressID` = '"+addressID+"' WHERE (`addressID` = '"+prevAddress2ID+"') and (`customerID` = '"+customerID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

}

}

}

System.out.println("=====================================================================================");

System.out.println("============================== ADDRESSS 3 ==========================================");

System.out.println("=====================================================================================");

if(address3\_address.isVisible() || !address3\_address.getText().equals(""))

{

System.out.println("Inside of address 3");

if(address2\_address.getText().equals("") || address2\_city.getText().equals("") || address2\_country.getText().equals(""))

label.setText("You Cannot Leave Previous Address Field Empty !");

else

{

System.out.println("Purely Inside of Address 3");

countryID = "";

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address3\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+address3\_country.getText()+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address3\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

}

System.out.println("Country ID : " + countryID);

cityID = "";

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address3\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+address3\_city.getText()+"', '"+countryID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address3\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

}

System.out.println("City ID : " + cityID);

addressID = "";

rs = stmt.executeQuery("Select addressID from address where address = '"+address3\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+address3\_address.getText()+"', '"+cityID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

rs = stmt.executeQuery("Select addressID from address where address = '"+address3\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

}

System.out.println("Address ID : " + addressID);

System.out.println("CustomerID : " +customerID);

System.out.println("Previous Address ID : " + prevAddress3ID);

if(prevAddress3ID.equals(""))

{

sql = "INSERT INTO `mydb`.`customer\_address` (`addressID`, `customerID`) VALUES ('"+addressID+"', '"+customerID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

}

else

{

sql = "UPDATE `mydb`.`customer\_address` SET `addressID` = '"+addressID+"' WHERE (`addressID` = '"+prevAddress3ID+"') and (`customerID` = '"+customerID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

}

}

}

}

public void mobileUpdate() throws SQLException

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String customerID = "";

ResultSet rs = stmt.executeQuery("Select customerID from customer where email = '"+usernameOfLoggedInPerson+"';");

if(rs.next())

customerID = rs.getString(1);

String sql = "UPDATE `mydb`.`customer\_phone` SET `phone\_number` = '"+mob1.getText()+"' WHERE (`phone\_number` = '"+prevMob1+"') and (`customerID` = '"+customerID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

if(mob2.isVisible())

{

if(prevMob2.equals(""))

{

sql = "INSERT INTO `mydb`.`customer\_phone` (`customerID`, `phone\_number`) VALUES ('"+customerID+"', '"+mob2.getText()+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

}

else

{

sql = "UPDATE `mydb`.`customer\_phone` SET `phone\_number` = '"+mob2.getText()+"' WHERE (`phone\_number` = '"+prevMob2+"') and (`customerID` = '"+customerID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

}

}

if(mob3.isVisible())

{

if(prevMob3.equals(""))

{

sql = "INSERT INTO `mydb`.`customer\_phone` (`customerID`, `phone\_number`) VALUES ('"+customerID+"', '"+mob3.getText()+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

}

else

{

sql = "UPDATE `mydb`.`customer\_phone` SET `phone\_number` = '"+mob3.getText()+"' WHERE (`phone\_number` = '"+prevMob3+"') and (`customerID` = '"+customerID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

}

}

}

public void addMobileNumber()

{

if(!mob2.isVisible() && !mob1.getText().equals(""))

mob2.setVisible(true);

else if(!mob3.isVisible() && !mob2.getText().equals(""))

mob3.setVisible(true);

else

{

if(mob2.isVisible())

label.setText("Already Visible");

else if(mob1.getText().equals(""))

label.setText("You Cannot Leave Mobile Number Field Empty");

if(mob3.isVisible())

label.setText("Already Visible");

else if(mob2.getText().equals(""))

label.setText("You Cannot Left Mobile Number Field Empty");

if(mob2.isVisible() && mob3.isVisible())

label.setText("You Can Only Add Three numbers");

}

}

public void addAddress()

{

if(address3\_address.isVisible())

label.setText("You Cannot Add more than 3 Addresses!");

if(address2\_address.isVisible())

{

if(address2\_address.getText().equals("") || address2\_city.getText().equals("") || address2\_country.getText().equals(""))

{

label.setText("You Cannot Leave Any previous Address Filed Empty ! ");

}

else

{

address3\_address.setVisible(true);

address3\_city.setVisible(true);

address3\_country.setVisible(true);

}

}

else

{

if(address1\_address.getText().equals("") || address1\_city.getText().equals("") || address1\_country.getText().equals(""))

{

label.setText("You Cannot Leave Any previous Address Filed Empty ! ");

}

else

{

address2\_address.setVisible(true);

address2\_city.setVisible(true);

address2\_country.setVisible(true);

}

}

}

public void saveAlert(ActionEvent event) throws IOException //this is INFORMATION

{

Alert a1 = new Alert(Alert.AlertType.INFORMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a1.setTitle("Save Changes");

a1.setContentText("Your Changes have been Saved ! ");

//a1.setHeaderText("Basic ALERT 2 "); //you can also set header

a1.showAndWait();

backButtonPushed(event);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Account.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

AccountController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

//System.out.println("Username in back button of Account : " + username.getText());

}

@Override

public void initialize(URL url, ResourceBundle rb)

{

logo = new ImageView("logo1.png");

logo2 = new ImageView("account.png");

favLang = new ToggleGroup();

male.setToggleGroup(favLang);

female.setToggleGroup(favLang);

address2\_address.setVisible(false);

address2\_city.setVisible(false);

address2\_country.setVisible(false);

address3\_address.setVisible(false);

address3\_city.setVisible(false);

address3\_country.setVisible(false);

mob2.setVisible(false);

mob3.setVisible(false);

// TODO

}

}

## Complaint Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ComplaintController implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private ImageView create;

@FXML private ImageView record;

@FXML private ImageView response;

String usernameOfLoggedInPerson;

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

}

public void createComplaintButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("createComplaint1.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CreateComplaint1Controller controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void createComplaintImagePushed(MouseEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("createComplaint1.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CreateComplaint1Controller controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void complaintRecordButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ComplaintRecord.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintRecordController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void complaintRecordImagePushed(MouseEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ComplaintRecord.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintRecordController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void complaintResponseButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ComplaintResponse.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintResponseController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void complaintResponseImagePushed(MouseEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ComplaintResponse.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintResponseController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerProfileController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb)

{

logo = new ImageView("logo1.png");

logo2 = new ImageView("complaintLogo.png");

create = new ImageView("createComplaint.png");

record = new ImageView("complaintRecord.png");

response = new ImageView("complaintResponse.png");

// TODO

}

}

## Complaint Modal Table:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import javafx.beans.property.SimpleStringProperty;

/\*\*

\*

\* @author Hamza Barak

\*/

public class ComplaintModalTable

{

private SimpleStringProperty srNo;

private SimpleStringProperty complaintCategory;

private SimpleStringProperty complaint;

private SimpleStringProperty submissionDate;

private SimpleStringProperty status;

public ComplaintModalTable(String srNo, String complaintCategory, String complaint, String submissionDate)

{

this.srNo = new SimpleStringProperty(srNo);

this.complaintCategory = new SimpleStringProperty(complaintCategory);

this.complaint = new SimpleStringProperty(complaint);

this.submissionDate = new SimpleStringProperty(submissionDate);

}

public ComplaintModalTable(String srNo, String complaintCategory, String complaint, String submissionDate, String status)

{

this.srNo = new SimpleStringProperty(srNo);

this.complaintCategory = new SimpleStringProperty(complaintCategory);

this.complaint = new SimpleStringProperty(complaint);

this.submissionDate = new SimpleStringProperty(submissionDate);

this.status = new SimpleStringProperty(status);

}

public String getSrNo() {

return srNo.get();

}

public void setSrNo(String srNo) {

this.srNo = new SimpleStringProperty(srNo);

}

public String getComplaintCategory() {

return complaintCategory.get();

}

public void setComplaintCategory(String complaintCategory) {

this.complaintCategory = new SimpleStringProperty(complaintCategory);

}

public String getComplaint() {

return complaint.get();

}

public void setComplaint(String complaint) {

this.complaint = new SimpleStringProperty(complaint);

}

public String getSubmissionDate() {

return submissionDate.get();

}

public void setSubmissionDate(String submissionDate) {

this.submissionDate = new SimpleStringProperty(submissionDate);

}

public String getStatus() {

return status.get();

}

public void setStatus(String status) {

this.status = new SimpleStringProperty(status);

}

}

## Complaint Record Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.Optional;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.ButtonType;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ComplaintRecordController implements Initializable {

@FXML private TableView<ComplaintModalTable> tableView;

@FXML private TableColumn<ComplaintModalTable, String> srNo;

@FXML private TableColumn<ComplaintModalTable, String> complaintCategory; //this will consider as the House Name

@FXML private TableColumn<ComplaintModalTable, String> complaint;

@FXML private TableColumn<ComplaintModalTable, String> submissionDate;

@FXML private TableColumn<ComplaintModalTable, String> status;

String usernameOfLoggedInPerson = "";

String categoryName = "";

String customerID = "";

String complaintID = "";

public void setParameter(String person)

{

this.usernameOfLoggedInPerson = person;

setTableData();

}

public void setTableData()

{

ObservableList<ComplaintModalTable> list = FXCollections.observableArrayList();

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

System.out.println("I am Before query 1");

ResultSet rs = stmt.executeQuery("select customerID from customer where email = '"+ usernameOfLoggedInPerson +"';");

while(rs.next())

customerID = rs.getString(1);

System.out.println("custoerID : " + customerID);

String categoryS = "";

String complaintS = "";

String dateS = "";

int counter = 1;

System.out.println("I am Before query 2");

rs = stmt.executeQuery("select complaint.complaintID, complaint\_category.category, complaint.complaint, complaint.complaint\_date from customer, complaint, complaint\_category where customer.customerID = complaint.customerID and complaint\_category.complaintCategoryID = complaint.complaintCategoryID and customer.customerID = '"+customerID+"';");

while(rs.next())

{

complaintID = complaintID + rs.getString(1) + "#";

categoryS = categoryS + rs.getString(2) + "#";

complaintS = complaintS + rs.getString(3) + "#";

dateS = dateS + rs.getString(4) + "#";

}

String complaintIDA[] = complaintID.split("#");

String categoryA[] = categoryS.split("#");

String complaintA[] = complaintS.split("#");

String dateA[] = dateS.split("#");

for(int i = 0; i < complaintIDA.length; i++)

{

System.out.println("I am Before query 3");

rs = stmt.executeQuery("select complaint.complaintID from complaint where complaintID = '" + complaintIDA[i] + "' and complaintID in (select complaintID from response);");

if(rs.next())

list.add(new ComplaintModalTable(Integer.toString(counter), categoryA[i], complaintA[i], dateA[i], "Resolved"));

else

list.add(new ComplaintModalTable(Integer.toString(counter), categoryA[i], complaintA[i], dateA[i], "Pending"));

counter++;

}

}

catch(Exception e)

{

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

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Complaint.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

srNo.setCellValueFactory(new PropertyValueFactory<ComplaintModalTable, String>("srNo"));

complaintCategory.setCellValueFactory(new PropertyValueFactory<ComplaintModalTable, String>("complaintCategory"));

complaint.setCellValueFactory(new PropertyValueFactory<ComplaintModalTable, String>("complaint"));

submissionDate.setCellValueFactory(new PropertyValueFactory<ComplaintModalTable, String>("submissionDate"));

status.setCellValueFactory(new PropertyValueFactory<ComplaintModalTable, String>("status"));

setTableData();

}

}

## Complaint Response 2 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.TextArea;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ComplaintResponse2Controller implements Initializable {

@FXML private TextArea textArea;

String usernameOfLoggedInPerson = "";

String complaintID = "";

String response = "";

public void setParameter(String person, ComplaintModalTable cmt)

{

usernameOfLoggedInPerson = person;

complaintID = cmt.getStatus();

System.out.println("Branch Name from Second : " + complaintID);

setData();

}

public void setData()

{

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select response.response from response where complaintID = '" + complaintID + "'");

while(rs.next())

response = rs.getString(1);

for(int i = 0; i < response.length(); i++)

{

if(i%30 == 0)

response = response.substring(0, i) + "-\n" + response.substring(i, response.length());

}

textArea.setText(response);

}

catch(Exception e)

{

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

}

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ComplaintResponse.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintResponseController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

setData();

textArea.setEditable(false);

}

}

## Complaint Response Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

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.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ComplaintResponseController implements Initializable

{

@FXML private TableView<ComplaintModalTable> tableView;

@FXML private TableColumn<ComplaintModalTable, String> srNo;

@FXML private TableColumn<ComplaintModalTable, String> complaintCategory; //this will consider as the House Name

@FXML private TableColumn<ComplaintModalTable, String> complaint;

@FXML private TableColumn<ComplaintModalTable, String> submissionDate;

@FXML private Button responseButton;

String usernameOfLoggedInPerson = "";

String categoryName = "";

String customerID = "";

String complaintID = "";

public void setParameter(String person)

{

this.usernameOfLoggedInPerson = person;

setTableData();

}

public void userClickOnTable()

{

this.responseButton.setDisable(false);

}

public void setTableData()

{

ObservableList<ComplaintModalTable> list = FXCollections.observableArrayList();

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

System.out.println("I am Before query 1");

ResultSet rs = stmt.executeQuery("select customerID from customer where email = '"+ usernameOfLoggedInPerson +"';");

while(rs.next())

customerID = rs.getString(1);

System.out.println("custoerID : " + customerID);

int counter = 1;

System.out.println("I am Before query 2");

//here i am storing complaintID in status variable of Complaint Modal Tabel

rs = stmt.executeQuery("select complaint\_category.category, complaint.complaint, complaint.complaint\_date, complaint.complaintID from customer, complaint, complaint\_category where customer.customerID = complaint.customerID and complaint\_category.complaintCategoryID = complaint.complaintCategoryID and customer.customerID = '"+customerID+"' and complaint.complaintID in (SELECT complaintID from response);");

while(rs.next())

{

//====================== NOTE =========================================

//here i am storing complaintID in 'status' variable of Complaint Modal Tabel

list.add(new ComplaintModalTable(Integer.toString(counter), rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4)));

counter++;

}

}

catch(Exception e)

{

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

}

tableView.setItems(list);

}

public void seeResponseButtonPressed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ComplaintResponse2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintResponse2Controller controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson, tableView.getSelectionModel().getSelectedItem());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Complaint.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

responseButton.setDisable(true);

srNo.setCellValueFactory(new PropertyValueFactory<ComplaintModalTable, String>("srNo"));

complaintCategory.setCellValueFactory(new PropertyValueFactory<ComplaintModalTable, String>("complaintCategory"));

complaint.setCellValueFactory(new PropertyValueFactory<ComplaintModalTable, String>("complaint"));

submissionDate.setCellValueFactory(new PropertyValueFactory<ComplaintModalTable, String>("submissionDate"));

setTableData();

}

}

## Create Complaint 1 controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TextArea;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class CreateComplaint1Controller implements Initializable {

@FXML private ComboBox categoryCombo;

@FXML private TextArea textArea;

@FXML private Label label;

@FXML private Button submitButton;

String usernameOfLoggedInPerson;

String categoryName = "";

int check = 0;

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

}

public void userClickOnCombo()

{

check++;

if(check == 2)

this.submitButton.setDisable(false);

}

public void userClickOnTextArea()

{

check++;

if(check == 2)

this.submitButton.setDisable(false);

}

public void submitComplaintButtonPushed(ActionEvent event) throws IOException

{

if(!categoryCombo.getValue().toString().equals("Select Category"))

{

if(!textArea.getText().equals(""))

{

categoryName = categoryCombo.getValue().toString();

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select customerID from customer where email = '" + usernameOfLoggedInPerson + "';");

String customerID = "";

while(rs.next())

{

customerID = rs.getString(1);

}

rs = stmt.executeQuery("select complaintCategoryID from complaint\_category where category = '" + categoryCombo.getValue().toString() + "';");

String categoryID = "";

while(rs.next())

{

categoryID = rs.getString(1);

}

String sql = "INSERT INTO `mydb`.`complaint` (`complaintCategoryID`, `complaint`, `customerID`, `complaint\_date`) VALUES ('"+categoryID+"', '"+textArea.getText()+"', '"+customerID+"', '"+Checks.getCurrentdate()+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql);

int rowsAffected = preparedStatement.executeUpdate();

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("Complaint");

a1.setContentText("Your Complaint Has Been Entered !");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Complaint.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

catch(Exception e)

{

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

}

}

else

label.setText("Fill The Text Area");

}

else

label.setText("Please Select Category First");

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Complaint.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

categoryCombo.getItems().addAll("food", "laundry", "cleaning", "mismanagement", "house condition", "other");

categoryCombo.setValue("Select Complaint Category");

this.submitButton.setDisable(true);

}

}

## Customer:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import javafx.beans.property.SimpleStringProperty;

/\*\*

\*

\* @author Hamza Barak

\*/

public class Customer

{

private SimpleStringProperty srNo;

private SimpleStringProperty customerName;

private SimpleStringProperty cnic;

private SimpleStringProperty service;

private SimpleStringProperty date;

private SimpleStringProperty time;

public Customer(String srNo, String customerName)

{

this.srNo = new SimpleStringProperty(srNo);

this.customerName = new SimpleStringProperty(customerName);

}

public Customer(String srNo, String customerName, String cnic)

{

this.srNo = new SimpleStringProperty(srNo);

this.customerName = new SimpleStringProperty(customerName);

this.cnic = new SimpleStringProperty(cnic);

}

public Customer(String srNo, String service, String date, String time)

{

this.srNo = new SimpleStringProperty(srNo);

this.service = new SimpleStringProperty(service);

this.date = new SimpleStringProperty(date);

this.time = new SimpleStringProperty(time);

}

public String getSrNo() {

return srNo.get();

}

public void setSrNo(String srNo) {

this.srNo = new SimpleStringProperty(srNo);

}

public String getService() {

return service.get();

}

public void setService(String service) {

this.service = new SimpleStringProperty(service);

}

public String getDate() {

return date.get();

}

public void setDate(String date) {

this.date = new SimpleStringProperty(date);

}

public String getTime() {

return time.get();

}

public void setTime(String time) {

this.time = new SimpleStringProperty(time);

}

public String getCustomerName() {

return customerName.get();

}

public void setCustomerName(String customerName) {

this.customerName = new SimpleStringProperty(customerName);

}

public String getCnic() {

return cnic.get();

}

public void setCnic(String cnic) {

this.cnic = new SimpleStringProperty(cnic);

}

}

## Customer All Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class CustomerAllController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<Customer> tableView;

@FXML private TableColumn<Customer, String> srNoC;

@FXML private TableColumn<Customer, String> customerNameC;

@FXML private TableColumn<Customer, String> cnicC;

@FXML private ComboBox combo;

@FXML private TextField searchBar;

@FXML private Button search;

@FXML private Button DetailView;

@FXML private Label label;

@FXML private Label label2;

int total\_customer = 0;

String empID = "";

String prevPageID = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from customer;");

while(rs.next())

total\_customer++;

label.setText("TOTAL CUSTOMERS : " + total\_customer);

setTableData();

}

public void setParameter(String empID, String s) throws SQLException

{

this.empID = empID;

this.prevPageID = s;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from customer;");

while(rs.next())

total\_customer++;

label.setText("TOTAL CUSTOMERS : " + total\_customer);

setTableData();

}

public void userClickOnTable()

{

this.DetailView.setDisable(false);

}

public void searchButtonPushed() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(combo.getValue().toString().equals("NAME"))

{

int counter = 1;

String name[] = searchBar.getText().split(" ");

String firstName = "";

String lastName = "";

if(name.length > 1)

{

firstName = name[0];

lastName = name[1];

}

else if(name.length == 1)

{

firstName = name[0];

}

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where first\_name = '"+ firstName +"' and last\_name = '"+ lastName +"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("FIRST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where first\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered First Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("LAST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where last\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Enterd Last Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CNIC"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where CNIC = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered CNIC is Not Present in our Database");

}

else if(combo.getValue().toString().equals("E-MAIL"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where email = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered E-MAIL is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CUSTOMER ID"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where customerID = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Customer ID is Not Present in our Database");

}

else

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer;");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

}

tableView.setItems(list);

}

public void setTableData() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer;");

while(rs.next())

{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}

tableView.setItems(list);

}

public void detailedViewButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("DetailedCustomerInfo.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

DetailedCustomerInfoController controller = loader.getController();

controller.setParameter(empID, tableView.getSelectionModel().getSelectedItem(), prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Customer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

DetailView.setDisable(true);

combo.getItems().addAll("NAME", "FIRST NAME", "LAST NAME","CNIC", "E-MAIL", "CUSTOMER ID", "Show All Data");

combo.setValue("Enter Your Desired Input for Search ");

srNoC.setCellValueFactory(new PropertyValueFactory<Customer, String>("srNo"));

customerNameC.setCellValueFactory(new PropertyValueFactory<Customer, String>("customerName"));

cnicC.setCellValueFactory(new PropertyValueFactory<Customer, String>("cnic"));

}

}

## Customer Booking Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class CustomerBookingController implements Initializable {

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> srNoC;

@FXML private TableColumn<HouseModalTable, String> booking; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> address;

@FXML private TableColumn<HouseModalTable, String> bookingDate;

@FXML private TableColumn<HouseModalTable, String> leavingDate;

@FXML private ComboBox combo;

String empID = "";

String customerCNIC = "";

String prevPageID = "";

public void setParameter(String empID, String customerCNIC, String s) throws SQLException

{

this.empID = empID;

this.customerCNIC = customerCNIC;

this.prevPageID = s;

setTableDataUponCombo();

}

public void setTableDataUponCombo() throws SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("Select customerID from customer where CNIC = '" + customerCNIC + "';");

boolean temp = rs.next();

String customerID = rs.getString(1);

if(combo.getValue().toString().equals("LUXURY"))

{

boolean check = false;

String houseID[];

String tempHouse = "";

String address[];

String tempAddress = "";

rs = stmt.executeQuery("select house.houseID, branch.branch\_name, location.location from house, house\_booking, branch, location where house.houseID = house\_booking.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and house\_booking.customerID = '"+customerID+"' and house.huose\_Type = 'Luxury';");

while(rs.next())

{

check = true;

tempHouse = tempHouse + rs.getString(1) + "#";

tempAddress = tempAddress + rs.getString(2) + ", " + rs.getString(3) + "#";

}

houseID = tempHouse.split("#");

address = tempAddress.split("#");

String booking[];

String leaving[];

String tempBooking = "";

String tempLeaving = "";

for(int i = 0; i < houseID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'h" + houseID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

int counter = 1;

if(check)

{

for(int i = 0; i < houseID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "House." + houseID[i], address[i], booking[i], leaving[i]));

counter++;

}

}

//=============================================================================================================================================================

check = false;

String tempForEmpty[] = new String[0];

houseID = tempForEmpty;

address = tempForEmpty;

booking = tempForEmpty;

leaving = tempForEmpty;

String room[]; //Room Number Not the Room ID

String roomID[];

tempHouse = "";

tempAddress = "";

tempBooking = "";

tempLeaving = "";

String tempRoomNo = "";

String tempRoomID = "";

rs = stmt.executeQuery("select room.room\_number, room.roomID, house.houseID, branch.branch\_name, location.location from room, house, room\_booking, branch, location where room.houseID = house.houseID and room.roomID = room\_booking.roomID and house.branchID = branch.branchID and branch.locationID = location.locationID and room\_booking.customerID = '"+customerID+"' and house.huose\_Type = 'roomsOnlyLuxury';");

while(rs.next())

{

check = true;

tempRoomNo = tempRoomNo + rs.getString(1) + "#";

tempRoomID = tempRoomID + rs.getString(2) + "#";

tempHouse = tempHouse + rs.getString(3) + "#";

tempAddress = tempAddress + rs.getString(4) + ", " + rs.getString(5) + "#";

}

room = tempRoomNo.split("#");

roomID = tempRoomID.split("#");

houseID = tempHouse.split("#");

address = tempAddress.split("#");

for(int i = 0; i < roomID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'r" + roomID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

if(check)

{

for(int i = 0; i < roomID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "Room No.(" + roomID[i] + ") HouseID (" + houseID[i] + ")", address[i], booking[i], leaving[i]));

counter++;

}

}

}

else if(combo.getValue().toString().equals("ECONOMY"))

{

boolean check = false;

String houseID[];

String tempHouse = "";

String address[];

String tempAddress = "";

rs = stmt.executeQuery("select house.houseID, branch.branch\_name, location.location from house, house\_booking, branch, location where house.houseID = house\_booking.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and house\_booking.customerID = '"+customerID+"' and house.huose\_Type = 'Economy';");

while(rs.next())

{

check = true;

tempHouse = tempHouse + rs.getString(1) + "#";

tempAddress = tempAddress + rs.getString(2) + ", " + rs.getString(3) + "#";

}

houseID = tempHouse.split("#");

address = tempAddress.split("#");

String booking[];

String leaving[];

String tempBooking = "";

String tempLeaving = "";

for(int i = 0; i < houseID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'h" + houseID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

int counter = 1;

if(check)

{

for(int i = 0; i < houseID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "House." + houseID[i], address[i], booking[i], leaving[i]));

counter++;

}

}

//=============================================================================================================================================================

check = false;

String tempForEmpty[] = new String[0];

houseID = tempForEmpty;

address = tempForEmpty;

booking = tempForEmpty;

leaving = tempForEmpty;

String room[]; //Room Number Not the Room ID

String roomID[];

tempHouse = "";

tempAddress = "";

tempBooking = "";

tempLeaving = "";

String tempRoomNo = "";

String tempRoomID = "";

rs = stmt.executeQuery("select room.room\_number, room.roomID, house.houseID, branch.branch\_name, location.location from room, house, room\_booking, branch, location where room.houseID = house.houseID and room.roomID = room\_booking.roomID and house.branchID = branch.branchID and branch.locationID = location.locationID and room\_booking.customerID = '"+customerID+"' and house.huose\_Type = 'roomsOnlyEconomy';");

while(rs.next())

{

check = true;

tempRoomNo = tempRoomNo + rs.getString(1) + "#";

tempRoomID = tempRoomID + rs.getString(2) + "#";

tempHouse = tempHouse + rs.getString(3) + "#";

tempAddress = tempAddress + rs.getString(4) + ", " + rs.getString(5) + "#";

}

room = tempRoomNo.split("#");

roomID = tempRoomID.split("#");

houseID = tempHouse.split("#");

address = tempAddress.split("#");

for(int i = 0; i < roomID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'r" + roomID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

if(check)

{

for(int i = 0; i < roomID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "Room No.(" + roomID[i] + ") HouseID (" + houseID[i] + ")", address[i], booking[i], leaving[i]));

counter++;

}

}

}

else

{

boolean check = false;

String houseID[];

String tempHouse = "";

String address[];

String tempAddress = "";

rs = stmt.executeQuery("select house.houseID, branch.branch\_name, location.location from house, house\_booking, branch, location where house.houseID = house\_booking.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and house\_booking.customerID = '"+customerID+"';");

while(rs.next())

{

check = true;

tempHouse = tempHouse + rs.getString(1) + "#";

tempAddress = tempAddress + rs.getString(2) + ", " + rs.getString(3) + "#";

}

houseID = tempHouse.split("#");

address = tempAddress.split("#");

String booking[];

String leaving[];

String tempBooking = "";

String tempLeaving = "";

for(int i = 0; i < houseID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'h" + houseID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

int counter = 1;

if(check)

{

for(int i = 0; i < houseID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "House." + houseID[i], address[i], booking[i], leaving[i]));

counter++;

}

}

//=============================================================================================================================================================

check = false;

String tempForEmpty[] = new String[0];

houseID = tempForEmpty;

address = tempForEmpty;

booking = tempForEmpty;

leaving = tempForEmpty;

String room[]; //Room Number Not the Room ID

String roomID[];

tempHouse = "";

tempAddress = "";

tempBooking = "";

tempLeaving = "";

String tempRoomNo = "";

String tempRoomID = "";

rs = stmt.executeQuery("select room.room\_number, room.roomID, house.houseID, branch.branch\_name, location.location from room, house, room\_booking, branch, location where room.houseID = house.houseID and room.roomID = room\_booking.roomID and house.branchID = branch.branchID and branch.locationID = location.locationID and room\_booking.customerID = '"+customerID+"';");

while(rs.next())

{

check = true;

tempRoomNo = tempRoomNo + rs.getString(1) + "#";

tempRoomID = tempRoomID + rs.getString(2) + "#";

tempHouse = tempHouse + rs.getString(3) + "#";

tempAddress = tempAddress + rs.getString(4) + ", " + rs.getString(5) + "#";

}

room = tempRoomNo.split("#");

roomID = tempRoomID.split("#");

houseID = tempHouse.split("#");

address = tempAddress.split("#");

for(int i = 0; i < roomID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'r" + roomID[i] + "';");

System.out.println("room ID from All (else) : " + roomID[i]);

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

if(check)

{

for(int i = 0; i < roomID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "Room No.(" + roomID[i] + ") HouseID (" + houseID[i] + ")", address[i], booking[i], leaving[i]));

counter++;

}

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("DetailedCustomerInfo.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

DetailedCustomerInfoController controller = loader.getController();

controller.setParameter(empID, customerCNIC, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

combo.getItems().addAll("ALL", "LUXURY", "ECONOMY");

combo.setValue("ALL");

srNoC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

booking.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("booking"));

address.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("address"));

bookingDate.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("bookingDate"));

leavingDate.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("leavingDate"));

}

}

## Customer Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class CustomerController implements Initializable {

@FXML private ImageView logo;

String empID = "";

String prevPageID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void setParameter(String empID, String s)

{

this.empID = empID;

this.prevPageID = s;

}

public void customerAllButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerAll.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerAllController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void leaveCustomerButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("LeavedCustomer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

LeavedCustomerController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void GivenServicesButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ServicesGivenToCustomer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ServicesGivenToCustomerController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void customerBillsButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomersBill.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomersBillController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

if(prevPageID.equals("AdminProfile.fxml"))

loader.setLocation(getClass().getResource("AdminProfile.fxml"));

else

loader.setLocation(getClass().getResource("ReceptionistProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

if(prevPageID.equals("AdminProfile.fxml"))

{

AdminProfileController controller = loader.getController();

controller.setParameter(empID);

}

else

{

ReceptionistProfileController controller = loader.getController();

controller.setParameter(empID);

}

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Customer login:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.PrintWriter;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import java.sql.\*;

import java.sql.DriverManager;

import java.util.Scanner;

import javafx.scene.control.CheckBox;

import javafx.scene.control.PasswordField;

import javafx.scene.control.TextField;

public class CustomerLogin implements Initializable {

/\*\*

\* Initializes the controller class.

\*/

@FXML private ImageView msgIcon;

@FXML private ImageView lockIcon;

@FXML private ImageView close;

@FXML private ImageView logo; // Main logo

@FXML private ImageView logo2; // for customer logo

@FXML private Label label;

@FXML private TextField username;

@FXML private PasswordField password;

@FXML

public void close\_app(MouseEvent event)

{

System.exit(0);

}

@FXML private CheckBox rememberMe;

//function for CHECK BOX

public void checkBox(String uName, String pass) throws FileNotFoundException

{

PrintWriter pw = new PrintWriter(new FileOutputStream("customer.txt"));

if(rememberMe.isSelected())

{

pw.println(uName + " " + pass);

System.out.println("WRITTEN \n");

}

else

{

pw.println();

System.out.println("\nNOT Written\n");

}

pw.close();

}

public void getCreditials() throws FileNotFoundException

{

Scanner reader = new Scanner(new FileInputStream("customer.txt"));

if(reader.hasNextLine())

{

String credentials[] = reader.nextLine().split(" ");

username.setText(credentials[0]);

password.setText(credentials[1]);

rememberMe.setSelected(true);

}

reader.close();

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

Parent tableViewParent = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving to page/Screen for reseting password

public void forgetPasswordButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ForgetPassword.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ForgetPasswordController controller = loader.getController();

controller.setParameter("CustomerLogin.fxml");

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving to registration page/Screen

public void registerButtonPushed(ActionEvent event) throws IOException

{

Parent tableViewParent = FXMLLoader.load(getClass().getResource("RegisterCustomer.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving to Customer Profile

public void loginButtonPushed(ActionEvent event) throws IOException

{

if(!(username.getText().equals("") || password.getText().equals("")) && loginAuthentication())

{

String tempUsername = username.getText();

String tempPassword = password.getText();

checkBox(tempUsername, tempPassword);

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerProfileController controller = loader.getController();

controller.setParameter(username.getText());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

{

if(username.getText().equals("") || password.getText().equals(""))

label.setText("You Cannot Leave Any of the Field Empty!");

if(!loginAuthentication())

label.setText("You Are NOT Registered in Our DATABASE !");

}

}

public boolean loginAuthentication() //this method return true when login successfull else return false

{

boolean check = false;

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT Customer.email, Customer.password from Customer where Customer.email = '" + username.getText() + "' and " + " Customer.password = '" + password.getText() + "';");

boolean temp = rs.next();

//System.out.println("EMal : " + rs.getString(1) + "\t\t" + "Password : " + rs.getString(2));

if(temp && rs.getString(1).equals(username.getText()) && rs.getString(2).equals(password.getText()))

{

check = true;

System.out.println("EMal : " + rs.getString(1) + "\t\t" + "Password : " + rs.getString(2));

}

}

catch(Exception e)

{

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

}

return check;

}

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

msgIcon = new ImageView("id.png");

lockIcon = new ImageView("lock.png");

close = new ImageView("close.png");

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerLogo.png");

try

{

getCreditials();

}

catch(Exception e)

{

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

}

}

}

## Customer Profile Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import java.sql.\*;

import java.sql.DriverManager;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.input.MouseEvent;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class CustomerProfileController implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private ImageView welcome;

@FXML private ImageView house;

@FXML private ImageView room;

@FXML private ImageView service;

@FXML private ImageView bill;

@FXML private ImageView complaint;

@FXML private ImageView account;

@FXML private Label nameLabel;

String username = "";

String firstName = "";

String lastName = "";

public void setParameter(String s)

{

username = s;

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select first\_name, last\_name from customer where email = '" + username + "';");

rs.next();

firstName = rs.getString(1);

lastName = rs.getString(2);

System.out.println(firstName + " " + lastName);

nameLabel.setText(firstName + " " + lastName);

}

catch(Exception e)

{

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

}

}

//============================================== IMAGE Methods =====================================================

@FXML

public void homeImagePressed(MouseEvent event) throws IOException // this function is call when home image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("House.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void roomImagePressed(MouseEvent event) throws IOException // this function is call when room image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Room.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void serviceImagePressed(MouseEvent event) throws IOException // this function is call when service image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Service.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ServiceController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void billImagePressed(MouseEvent event) throws IOException // this function is call when bill image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Bills.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BillsController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void complaintImagePressed(MouseEvent event) throws IOException // this function is call when complaint image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Complaint.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

System.out.println("Username in Profile : " + username);

ComplaintController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void accountImagePressed(MouseEvent event) throws IOException // this function is call when account image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Account.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

System.out.println("Username in Profile : " + username);

AccountController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//------------------------------------------ END Image Methods -----------------------------------------------------------

//============================================ BUTTON METHODS ============================================================

public void homeButtonPressed(ActionEvent event) throws IOException // this function is call when home Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("House.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void roomButtonPressed(ActionEvent event) throws IOException // this function is call when room Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Room.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void serviceButtonPressed(ActionEvent event) throws IOException // this function is call when service Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Service.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ServiceController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void billButtonPressed(ActionEvent event) throws IOException // this function is call when bill Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Bills.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BillsController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void complaintButtonPressed(ActionEvent event) throws IOException // this function is call when complaint Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Complaint.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ComplaintController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void accountButtonPressed(ActionEvent event) throws IOException // this function is call when acount Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Account.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AccountController controller = loader.getController();

controller.setParameter(username);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//--------------------------------------- END BUTTON METHODS ----------------------------------------------------------

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerNew.png");

welcome = new ImageView("welcome3.png");

house = new ImageView("house.png");

room = new ImageView("room.png");

service = new ImageView("service.png");

bill = new ImageView("bill.png");

complaint = new ImageView("complaint.png");

account = new ImageView("account.png");

//setFirstName();

}

}

## Customers Bill Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class CustomersBillController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<Customer> tableView;

@FXML private TableColumn<Customer, String> srNoC;

@FXML private TableColumn<Customer, String> customerNameC;

@FXML private TableColumn<Customer, String> cnicC;

@FXML private ComboBox combo;

@FXML private TextField searchBar;

@FXML private Button search;

@FXML private Button DetailView;

@FXML private Label label;

@FXML private Label label2;

int total\_customer = 0;

String empID = "";

String prevPageID = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from customer;");

while(rs.next())

total\_customer++;

label.setText("TOTAL CUSTOMERS : " + total\_customer);

setTableData();

}

public void setParameter(String empID, String s) throws SQLException

{

this.empID = empID;

this.prevPageID = s;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from customer;");

while(rs.next())

total\_customer++;

label.setText("TOTAL CUSTOMERS : " + total\_customer);

setTableData();

}

public void userClickOnTable()

{

this.DetailView.setDisable(false);

}

public void searchButtonPushed() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(combo.getValue().toString().equals("NAME"))

{

int counter = 1;

String name[] = searchBar.getText().split(" ");

String firstName = "";

String lastName = "";

if(name.length > 1)

{

firstName = name[0];

lastName = name[1];

}

else if(name.length == 1)

{

firstName = name[0];

}

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where first\_name = '"+ firstName +"' and last\_name = '"+ lastName +"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("FIRST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where first\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered First Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("LAST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where last\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Enterd Last Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CNIC"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where CNIC = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered CNIC is Not Present in our Database");

}

else if(combo.getValue().toString().equals("E-MAIL"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where email = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered E-MAIL is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CUSTOMER ID"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where customerID = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Customer ID is Not Present in our Database");

}

else

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer;");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

}

tableView.setItems(list);

}

public void setTableData() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer;");

while(rs.next())

{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}

tableView.setItems(list);

}

public void detailedViewButtonPushed() throws SQLException

{

String cnic = tableView.getSelectionModel().getSelectedItem().getCnic();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select customerID from customer where CNIC = '"+cnic+"';");

rs.next();

String customerID = rs.getString(1);

rs = stmt.executeQuery("select customerID, sum(bills) as Total from customer\_record where customerID = '"+customerID+"' group by customerID;");

if(rs.next())

{

Alert a1 = new Alert(Alert.AlertType.INFORMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a1.setTitle("Bill Of Customer");

a1.setContentText("Total Bill of Selected Customer : " + rs.getString(2));

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

}

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Customer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb)

{

// TODO

logo = new ImageView("logo1.png");

DetailView.setDisable(true);

combo.getItems().addAll("NAME", "FIRST NAME", "LAST NAME","CNIC", "E-MAIL", "CUSTOMER ID", "Show All Data");

combo.setValue("Enter Your Desired Input for Search ");

srNoC.setCellValueFactory(new PropertyValueFactory<Customer, String>("srNo"));

customerNameC.setCellValueFactory(new PropertyValueFactory<Customer, String>("customerName"));

cnicC.setCellValueFactory(new PropertyValueFactory<Customer, String>("cnic"));

}

}

## Delete Employee:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class DeleteEmployeeController implements Initializable {

@FXML private ImageView logo;

@FXML private TextField cnic;

@FXML private Label label;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void deleteEmployee() throws SQLException

{

if(!cnic.getText().equals(""))

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String employeeID = "";

ResultSet rs = stmt.executeQuery("select empID from employee where CNIC = '"+cnic.getText()+"';");

if(rs.next())

{

employeeID = rs.getString(1);

}

else

label.setText("No Such Person Exits in Employe Database !");

if(!employeeID.equals(""))

{

rs = stmt.executeQuery("select \* from employee where empID = '"+employeeID+"';");

while(rs.next())

{

String sql = "INSERT INTO `mydb`.`empoyeeleft` (`empID`, `first\_name`, `last\_name`, `gender`, `DOB`, `CNIC`, `joiningDate`, `leavingDate`, `salary`, `branch`) VALUES ('"+rs.getString(1)+"', '"+rs.getString(2)+"', '"+rs.getString(3)+"', '"+rs.getString(4)+"', '"+rs.getString(5)+"', '"+rs.getString(6)+"', '"+rs.getString(7)+"', '"+Checks.getCurrentdate()+"', '"+rs.getString(8)+"', '"+rs.getString(9)+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

}

String sql = "DELETE FROM `mydb`.`employee` WHERE (`empID` = '"+employeeID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("Successfully Deleted ");

a1.setContentText("Employee has been Deleted successfully!");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

}

}

else

label.setText("Please Enter Eny CNIC First !");

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

AdminEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Detailed Customer Info:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class DetailedCustomerInfoController implements Initializable {

@FXML private ImageView logo;

@FXML private Label name;

@FXML private Label gender;

@FXML private Label dob;

@FXML private Label cnic;

@FXML private Label age;

@FXML private Label username;

@FXML private Label joiningDate;

@FXML private Label mob1;

@FXML private Label mob2;

@FXML private Label mob3;

@FXML private Label address1\_1;

@FXML private Label address1\_2;

@FXML private Label address2\_1;

@FXML private Label address2\_2;

@FXML private Label address3\_1;

@FXML private Label address3\_2;

String empID = "";

String customerCNIC = "";

String customerID = "";

String prevPageID = "";

public void setParameter(String empID, String c)

{

this.empID = empID;

this.customerCNIC = c;

setDetailDataOfUser();

}

public void setParameter(String empID, Customer c)

{

this.empID = empID;

this.customerCNIC = c.getCnic();

setDetailDataOfUser();

}

public void setParameter(String empID, Customer c, String s)

{

this.empID = empID;

this.customerCNIC = c.getCnic();

this.prevPageID = s;

setDetailDataOfUser();

}

public void setParameter(String empID, String c, String s)

{

this.empID = empID;

this.customerCNIC = c;

this.prevPageID = s;

setDetailDataOfUser();

}

public void setDetailDataOfUser()

{

try

{

System.out.println("Customer CNIC from Account : "+customerCNIC);

cnic.setText(customerCNIC);

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

//ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, CNIC, joining\_date from customer where email = '" + s + "';");

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, email, joining\_date, address, city\_name, country\_name, phone\_number, customer.customerID from customer, address, customer\_phone, customer\_address, country, city where Customer.customerID = customer\_phone.customerID and customer\_address.customerID = customer.customerID and customer\_address.addressID = address.addressID and address.City\_cityID = city.cityID and city.countryID = country.countryID and customer.CNIC = '" + customerCNIC + "';");

boolean temp = rs.next();

name.setText(rs.getString(1) + " " + rs.getString(2));

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

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

System.out.println("\nDate Of Birth : "+dob.getText());

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

joiningDate.setText(rs.getString(6));

customerID = rs.getString(11);

//for Age we Do,

String dataForAge[] = dob.getText().split("-");

int ageInInt = Checks.ageReturner(Integer.parseInt(dataForAge[0]), Integer.parseInt(dataForAge[1]), Integer.parseInt(dataForAge[2]));

age.setText(Integer.toString(ageInInt));

rs = stmt.executeQuery("select distinct address, city\_name, country\_name from customer, address, customer\_address, country, city where customer\_address.customerID = customer.customerID and customer\_address.addressID = address.addressID and address.City\_cityID = city.cityID and city.countryID = country.countryID and customer.CNIC = '"+ customerCNIC +"';");

temp = rs.next();

address1\_1.setText(rs.getString(1) + ", ");

address1\_2.setText(rs.getString(2) + ", " + rs.getString(3));

int counter = 1;

while(rs.next())

{

if(counter == 1)

{

address2\_1.setText(rs.getString(1) + ", ");

address2\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

else

{

address3\_1.setText(rs.getString(1) + ", ");

address3\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

counter++;

}

rs = stmt.executeQuery("select distinct phone\_number from customer, customer\_phone where Customer.customerID = customer\_phone.customerID and customer.CNIC = '"+customerCNIC+"';");

temp = rs.next();

mob1.setText(rs.getString(1));

counter = 1;

while(rs.next())

{

if(counter == 1)

{

mob2.setText(rs.getString(1));

}

else

{

mob3.setText(rs.getString(1));

}

counter++;

}

}

catch(Exception e)

{

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

}

}

public void seeBookingButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerBooking.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerBookingController controller = loader.getController();

controller.setParameter(empID, customerCNIC, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerAll.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerAllController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Earning Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.TextArea;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class EarningController implements Initializable {

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void wholeEarningButtonPushed() throws SQLException

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String bills = "0";

System.out.println("BILLS : " +bills);

ResultSet rs = stmt.executeQuery("select sum(bills) from customer\_record;");

rs.next();

bills = rs.getString(1);

int earning = 0;

String roomData[];

String houseData[];

String houseTemp = "";

String roomTemp = "";

rs = stmt.executeQuery("select booking from customer\_record;");

while(rs.next())

{

String selection = rs.getString(1);

System.out.println(selection);

if(selection.startsWith("r"))

roomTemp = roomTemp + selection + "#";

else

houseTemp = houseTemp + selection + "#";

}

roomData = roomTemp.split("#");

houseData = houseTemp.split("#");

for(int i = 0; i < roomData.length; i++)

{

rs = stmt.executeQuery("select house.huose\_type from house, room where room.houseID = house.houseID and room.roomID = '"+roomData[i].substring(1, roomData[i].length())+"';");

while(rs.next())

{

if(rs.getString(1).equals("roomsOnlyEconomy"))

earning = earning + 10000;

else

earning = earning + 15000;

}

}

for(int i = 0; i < houseData.length; i++)

{

rs = stmt.executeQuery("select house.huose\_type from house where houseID = '"+houseData[i].substring(1, houseData[i].length())+"';");

while(rs.next())

{

if(rs.getString(1).equals("Economy"))

earning = earning + 20000;

else

earning = earning + 25000;

}

}

earning = earning + Integer.parseInt(bills);

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("TOTAL EARNING");

a1.setContentText("Total Earnings : " + earning + " Rupees");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

}

public void earningRecordButtonPushed() throws SQLException

{

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("EARNING RECORD");

a1.setContentText("See the whole Record of Earning!");

a1.setHeaderText("See the whole Record of Earning!");

String temp = "\t-----------------------------------------------------------------------------------------------------------------\n\t CUSTOMER ID \t\t BOOKING \t\t BOOKING DATE \t\t LEAVING DATE \t\t BILLS\n\t------------------------------------------------------------------------------------------------------------------\n";

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM mydb.customer\_record;");

while(rs.next())

{

temp = temp + "\t\t\t " + rs.getString(1) + "\t\t\t\t" + rs.getString(2) + "\t\t\t " + rs.getString(3) + "\t\t\t " + rs.getString(4) + "\t\t\t " + rs.getString(5) + "\n";

}

temp = temp + "\t-----------------------------------------------------------------------------------------------------------------";

//TextArea ta = new TextArea(s.toString());

TextArea ta = new TextArea(temp);

a1.getDialogPane().setExpandableContent(ta);

a1.showAndWait();

}

public void earningOfThisMonthButtonPushed() throws SQLException

{

String date = Checks.getCurrentdate();

String dateData[] = date.split("-");

String currentDate = "";

if(dateData[1].length() == 1)

currentDate = dateData[0] + "-" + "0" + dateData[1];

else

currentDate = dateData[0] + "-" + dateData[1];

System.out.println("Current Date : " + currentDate);

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int bills = 0;

System.out.println("Bills : " + bills);

ResultSet rs = stmt.executeQuery("select sum(bills) from customer\_record where booking\_date like '"+currentDate+"%';");

if(rs.next())

{

String temp = rs.getString(1);

System.out.println("TEMP : " + temp);

if(temp != null)

bills = Integer.parseInt(temp);

}

int earning = 0;

String roomData[];

String houseData[];

String houseTemp = "";

String roomTemp = "";

rs = stmt.executeQuery("select booking, booking\_date from customer\_record where booking\_date like '"+currentDate+"%';");

while(rs.next())

{

String bookingDate = rs.getString(2);

String selection = rs.getString(1);

// if(bookingDate.substring(0, 7).equals(Checks.getCurrentdate().substring(0, 7)))

// {

if(selection.startsWith("r"))

roomTemp = roomTemp + selection + "#";

else

houseTemp = houseTemp + selection + "#";

// }

}

roomData = roomTemp.split("#");

houseData = houseTemp.split("#");

for(int i = 0; i < roomData.length; i++)

{

if(!roomData[i].equals(""))

{

System.out.println("ROOM DATA FROM ARRAY INDEX " + i + " : " + roomData[i]);

rs = stmt.executeQuery("select house.huose\_type from house, room where room.houseID = house.houseID and room.roomID = '"+roomData[i].substring(1, roomData[i].length())+"';");

while(rs.next())

{

if(rs.getString(1).equals("roomsOnlyEconomy"))

earning = earning + 10000;

else

earning = earning + 15000;

}

}

}

for(int i = 0; i < houseData.length; i++)

{

if(!houseData[i].equals(""))

{

rs = stmt.executeQuery("select house.huose\_type from house where houseID = '"+houseData[i].substring(1, houseData[i].length())+"';");

while(rs.next())

{

if(rs.getString(1).equals("Economy"))

earning = earning + 20000;

else

earning = earning + 25000;

}

}

}

earning = earning + bills;

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("TOTAL EARNING");

a1.setContentText("Total Earnings of this Month : " + earning + " Rupees");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AdminProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

}

}

## Employee:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import javafx.beans.property.SimpleStringProperty;

/\*\*

\*

\* @author Hamza Barak

\*/

public class Employee {

private SimpleStringProperty srNo;

private SimpleStringProperty employeeName;

private SimpleStringProperty cnic;

public Employee(String srNo, String employeeName, String cnic)

{

this.srNo = new SimpleStringProperty(srNo);

this.employeeName = new SimpleStringProperty(employeeName);

this.cnic = new SimpleStringProperty(cnic);

}

public String getSrNo() {

return srNo.get();

}

public void setSrNo(String srNo) {

this.srNo = new SimpleStringProperty(srNo);

}

public String getEmployeeName() {

return employeeName.get();

}

public void setEmployeeName(String employeeName) {

this.employeeName = new SimpleStringProperty(employeeName);

}

public String getCnic() {

return cnic.get();

}

public void setCnic(String cnic) {

this.cnic = new SimpleStringProperty(cnic);

}

}

## Employee 2 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class Employee2Controller implements Initializable {

@FXML private ImageView logo;

@FXML private Label name;

@FXML private Label gender;

@FXML private Label dob;

@FXML private Label cnic;

@FXML private Label age;

@FXML private Label salary;

@FXML private Label joiningDate;

@FXML private Label mob1;

@FXML private Label mob2;

@FXML private Label mob3;

@FXML private Label address1\_1;

@FXML private Label address1\_2;

@FXML private Label address2\_1;

@FXML private Label address2\_2;

@FXML private Label address3\_1;

@FXML private Label address3\_2;

String empID = "";

String employeeCNIC = "";

String employeeID = "";

public void setParameter(String empID, String c)

{

this.empID = empID;

this.employeeCNIC = c;

setDetailDataOfUser();

}

public void setParameter(String empID, Employee c)

{

this.empID = empID;

this.employeeCNIC = c.getCnic();

setDetailDataOfUser();

}

public void setDetailDataOfUser()

{

try

{

System.out.println("Customer CNIC from Account : "+employeeCNIC);

cnic.setText(employeeCNIC);

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

//ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, CNIC, joining\_date from customer where email = '" + s + "';");

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, salary, joinDate, employee.empID from employee where employee.CNIC = '" + employeeCNIC + "';");

boolean temp = rs.next();

name.setText(rs.getString(1) + " " + rs.getString(2));

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

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

System.out.println("\nDate Of Birth : "+dob.getText());

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

joiningDate.setText(rs.getString(6));

String employeeID = rs.getString(7);

rs = stmt.executeQuery("select distinct address, city\_name, country\_name from employee, address, address\_has\_employee, country, city where address\_has\_employee.empID = employee.empID and address\_has\_employee.addressID = address.addressID and address.City\_cityID = city.cityID and city.countryID = country.countryID and employee.CNIC = '"+ employeeCNIC +"';");

temp = rs.next();

address1\_1.setText(rs.getString(1) + ", ");

address1\_2.setText(rs.getString(2) + ", " + rs.getString(3));

int counter = 1;

while(rs.next())

{

if(counter == 1)

{

address2\_1.setText(rs.getString(1) + ", ");

address2\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

else

{

address3\_1.setText(rs.getString(1) + ", ");

address3\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

counter++;

}

rs = stmt.executeQuery("select distinct phone\_number from employee, employee\_phone where employee.empID = employee\_phone.empID and employee.CNIC = '"+employeeCNIC+"';");

temp = rs.next();

mob1.setText(rs.getString(1));

counter = 1;

while(rs.next())

{

if(counter == 1)

{

mob2.setText(rs.getString(1));

}

else

{

mob3.setText(rs.getString(1));

}

counter++;

}

}

catch(Exception e)

{

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

}

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Employee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

EmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Employee Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class EmployeeController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<Employee> tableView;

@FXML private TableColumn<Employee, String> srNoC;

@FXML private TableColumn<Employee, String> customerNameC;

@FXML private TableColumn<Employee, String> cnicC;

@FXML private ComboBox combo;

@FXML private TextField searchBar;

@FXML private Button search;

@FXML private Button DetailView;

@FXML private Label label;

@FXML private Label label2;

int total\_customer = 0;

String empID = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from employee;");

while(rs.next())

total\_customer++;

label.setText("TOTAL Employees : " + total\_customer);

setTableData();

}

public void userClickOnTable()

{

this.DetailView.setDisable(false);

}

public void searchButtonPushed() throws SQLException

{

ObservableList<Employee> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(combo.getValue().toString().equals("NAME"))

{

int counter = 1;

String name[] = searchBar.getText().split(" ");

String firstName = "";

String lastName = "";

if(name.length > 1)

{

firstName = name[0];

lastName = name[1];

}

else if(name.length == 1)

{

firstName = name[0];

}

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where first\_name = '"+ firstName +"' and last\_name = '"+ lastName +"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("FIRST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where first\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered First Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("LAST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where last\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Enterd Last Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CNIC"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where CNIC = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered CNIC is Not Present in our Database");

}

else if(combo.getValue().toString().equals("EMPLOYEE ID"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where employeeID = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered EMPLOYEE ID is Not Present in our Database");

}

else

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee;");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

}

tableView.setItems(list);

}

public void setTableData() throws SQLException

{

ObservableList<Employee> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee;");

while(rs.next())

{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}

tableView.setItems(list);

}

public void detailedViewButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Employee2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Employee2Controller controller = loader.getController();

controller.setParameter(empID, tableView.getSelectionModel().getSelectedItem());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("EmpployeeMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

EmpployeeMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

DetailView.setDisable(true);

combo.getItems().addAll("NAME", "FIRST NAME", "LAST NAME","CNIC", "EMPLOYEE ID", "Show All Data");

combo.setValue("Enter Your Desired Input for Search ");

srNoC.setCellValueFactory(new PropertyValueFactory<Employee, String>("srNo"));

customerNameC.setCellValueFactory(new PropertyValueFactory<Employee, String>("employeeName"));

cnicC.setCellValueFactory(new PropertyValueFactory<Employee, String>("cnic"));

}

}

## Employee Main Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class EmpployeeMainController implements Initializable {

@FXML private ImageView logo;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void employeeAllButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Employee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

EmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void leaveEmployeeButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("LeaveEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

LeaveEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void modifyEmployeeButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ModifyEmployee1.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ModifyEmployee1Controller controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ReceptionistProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

FXML Document Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import javafx.stage.Stage;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.\*;

import javafx.stage.Stage;

/\*\*

\*

\* @author Hamza Barak

\*/

public class FXMLDocumentController implements Initializable {

@FXML private ImageView logo;

//Change Screen to Customer Login

@FXML

public void changeScreenToCustomerLogin(ActionEvent event) throws IOException

{

Parent tableViewParent = FXMLLoader.load(getClass().getResource("CustomerLogin.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void changeScreenToAdminLogin(ActionEvent event) throws IOException

{

Parent tableViewParent = FXMLLoader.load(getClass().getResource("AdminLogin.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void changeScreenToReceptionistLogin(ActionEvent event) throws IOException

{

Parent tableViewParent = FXMLLoader.load(getClass().getResource("ReceptionistLogin.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@Override

public void initialize(URL url, ResourceBundle rb)

{

logo = new ImageView("logo1.png");

}

}

## Forget Password 2:

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Label;

import javafx.scene.control.PasswordField;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ForgetPassword2Controller implements Initializable

{

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private PasswordField pass1;

@FXML private PasswordField pass2;

@FXML private Label label;

public String backPageParameter; //this attribute will store store the parameteric value for moving back to previous page and also to verify from which (customer/employee/admin) the forget password request is made

String cnic = "";

String mob = "";

String username = "";

String backPageParameterFromAccounts = "";

String thingToBeReturned = "";

public void setParameter(String s, String backPageParameterFromAccounts, String cnic, String mob)

{

this.backPageParameter = s;

this.backPageParameterFromAccounts = backPageParameterFromAccounts;

this.cnic = cnic;

this.mob = mob;

}

public void setParameter(String s, String cnic, String mob)

{

this.backPageParameter = s;

this.cnic = cnic;

this.mob = mob;

}

public void changePasswordButtonPushed(ActionEvent event) throws IOException, SQLException

{

if(!(pass1.getText().equals("") || pass2.getText().equals("")) && pass1.getText().equals(pass2.getText()))

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(backPageParameter.equals("Account") || backPageParameterFromAccounts.equals("CustomerLogin.fxml"))

{

String sql = "UPDATE `mydb`.`customer` SET `password` = '"+pass1.getText()+"' WHERE (`CNIC` = '"+cnic+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql);

int rowsAffected = preparedStatement.executeUpdate();

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("Password Changed ");

a1.setContentText("Your Password has been changed successfully, Now you can login through your new Password !");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerLogin.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else if(backPageParameter.equals("ReceptionistAccount") || backPageParameterFromAccounts.equals("ReceptionistLogin.fxml"))

{

ResultSet rs = stmt.executeQuery("select empID from employee where CNIC = '"+ cnic +"';");

rs.next();

String empID = rs.getString(1);

String sql = "UPDATE `mydb`.`receptionist` SET `password` = '"+pass1.getText()+"' WHERE (`empID` = '"+empID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql);

int rowsAffected = preparedStatement.executeUpdate();

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("Password Changed ");

a1.setContentText("Your Password has been changed successfully, Now you can login through your new Password !");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistLogin.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else if(backPageParameter.equals("AdminAccount") || backPageParameterFromAccounts.equals("AdminLogin.fxml"))

{

ResultSet rs = stmt.executeQuery("select empID from employee where CNIC = '"+ cnic +"';");

rs.next();

String empID = rs.getString(1);

String sql = "UPDATE `mydb`.`admin` SET `password` = '"+pass1.getText()+"' WHERE (`empID` = '"+empID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql);

int rowsAffected = preparedStatement.executeUpdate();

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("Password Changed ");

a1.setContentText("Your Password has been changed successfully, Now you can login through your new Password !");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminLogin.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

}

else

{

if(!pass1.getText().equals(pass2.getText()))

{

label.setText("Password Did'nt Match Re Enter!");

pass1.setText("");

pass2.setText("");

}

else

label.setText("You Cannot Leave Any of the Field Empty!");

}

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource(backPageParameter+".fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

if(backPageParameter.equals("ForgetPassword"))

{

ForgetPasswordController controller = loader.getController();

controller.setParameter(backPageParameterFromAccounts);

}

if(backPageParameter.equals("Account"))

{

AccountController controller2 = loader.getController();

controller2.setParameter(backPageParameterFromAccounts);

}

if(backPageParameter.equals("ReceptionistAccount"))

{

ReceptionistAccountController controller2 = loader.getController();

controller2.setParameter(backPageParameterFromAccounts);

}

if(backPageParameter.equals("AdminAccount"))

{

AdminAccountController controller2 = loader.getController();

controller2.setParameter(backPageParameterFromAccounts);

}

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerLogo.png");

}

}

## Forget Password Controller:

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

public class ForgetPasswordController implements Initializable

{

@FXML private ImageView logo;

@FXML private ImageView logo2;

public String backPageParameter; //this attribute will store store the parameteric value for moving back to previous page and also to verify from which (customer/employee/admin) the forget password request is made

@FXML private TextField cnic;

@FXML private TextField mobNo;

@FXML private Label label;

public void setParameter(String s)

{

backPageParameter = s;

}

//this function is for next button in forget password controll

public void nextButtonPushed(ActionEvent event) throws IOException, SQLException

{

if(!(cnic.getText().equals("") || mobNo.getText().equals("")) && Checks.cnicCheck(cnic.getText()) && Checks.mobCheck(mobNo.getText()))

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(backPageParameter.equals("CustomerLogin.fxml"))

{

if(!Checks.cnicCheckData(cnic.getText()))

{

if(!Checks.mobCheckData(mobNo.getText()))

{

ResultSet rs = stmt.executeQuery("Select customer.customerID from customer, customer\_phone where customer.customerID = customer\_phone.customerID and customer.CNIC = '"+ cnic.getText() +"' and customer\_phone.phone\_number = '" + mobNo.getText() +"';");

if(rs.next())

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ForgetPassword2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ForgetPassword2Controller controller = loader.getController();

controller.setParameter("ForgetPassword", backPageParameter, cnic.getText(), mobNo.getText());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("Your Entere CNIC and Mobile Number Donot Belong to Same Person");

}

else

label.setText("Your Entered Mobile Number is NOT Registered !");

}

else

label.setText("Your Entered CNIC is NOT Registered !");

}

else if(backPageParameter.equals("ReceptionistLogin.fxml"))

{

ResultSet rs = stmt.executeQuery("Select receptionist.empID from receptionist, employee\_phone, employee where employee.empID = receptionist.empID and receptionist.empID = employee\_phone.empID and employee.CNIC = '"+ cnic.getText() +"' and employee\_phone.phone\_number = '" + mobNo.getText() +"';");

if(rs.next())

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ForgetPassword2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ForgetPassword2Controller controller = loader.getController();

controller.setParameter("ForgetPassword", backPageParameter, cnic.getText(), mobNo.getText());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("Your Enter CNIC and Mobile Number Donot Belong to Receptionist");

}

else if(backPageParameter.equals("AdminLogin.fxml"))

{

ResultSet rs = stmt.executeQuery("Select admin.empID from admin, employee\_phone, employee where employee.empID = admin.empID and admin.empID = employee\_phone.empID and employee.CNIC = '"+ cnic.getText() +"' and employee\_phone.phone\_number = '" + mobNo.getText() +"';");

if(rs.next())

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ForgetPassword2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ForgetPassword2Controller controller = loader.getController();

controller.setParameter("ForgetPassword", backPageParameter, cnic.getText(), mobNo.getText());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("Your Enter CNIC and Mobile Number Donot Belong to Receptionist");

}

}

else

{

if(!Checks.mobCheck(mobNo.getText()))

{

label.setText("Please! Enter Mob Number in Format Like '03035405615' !"); //there is possiility that either the customer from any of the country so there number pattern could be different

mobNo.setText("");

}

if(!Checks.cnicCheck(cnic.getText()))

{

label.setText("Please! Enter CNIC in Format Like '12345-1234567-1' !");

cnic.setText("");

}

if(cnic.getText().equals("") || mobNo.getText().equals(""))

label.setText("You Cannot Leave Any of the Field Empty!");

}

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

Parent tableViewParent = FXMLLoader.load(getClass().getResource(backPageParameter));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

backPageParameter = "";

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerLogo.png");

}

}

## Guest House Management System:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.Image;

import javafx.stage.Stage;

/\*\*

\*

\* @author Hamza Barak

\*/

public class GuestHouseManagement extends Application {

@Override

public void start(Stage stage) throws Exception {

Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

Scene scene = new Scene(root);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

scene.getStylesheets().add(css);

Image image = new Image("logo1.png");

stage.getIcons().add(image);

stage.setTitle("HB Guest House Management System ");

stage.setResizable(false);

stage.setScene(scene);

stage.show();

}

public static void main(String[] args) {

launch(args);

}

}

## Guest House Bookees Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class HouseBookeesController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox branch;

@FXML private ComboBox category;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void showResultButtonPushed(ActionEvent event) throws IOException, SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(category.getValue().toString().equals("ALL"))

{

String tempHouse = "";

String tempBranch = "";

String tempLocation = "";

String tempCity = "";

String tempType = "";

String house[];

String branch\_name[];

String location[];

String city[];

String houseType[];

ResultSet rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location where branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type not like 'roomsOnly%';");

while(rs.next())

{

tempHouse = tempHouse + rs.getString(1) + "#";

tempBranch = tempBranch + rs.getString(2) + "#";

tempLocation = tempLocation + rs.getString(3) + "#";

tempCity = tempCity + rs.getString(4) + "#";

tempType = tempType + rs.getString(5) + "#";

}

house = tempHouse.split("#");

branch\_name = tempBranch.split("#");

location = tempLocation.split("#");

city = tempCity.split("#");

houseType = tempType.split("#");

for(int i = 0; i < house.length; i++)

{

rs = stmt.executeQuery("SELECT booking, count(\*) as Total\_Bookees from customer\_record where booking = 'h"+house[i]+"' group by booking;");

if(rs.next())

{

list.add(new HouseModalTable("House No." + house[i], branch\_name[i], location[i], city[i], houseType[i], rs.getString(2)));

}

else

{

list.add(new HouseModalTable("House No." + house[i], branch\_name[i], location[i], city[i], houseType[i], "0"));

}

}

}

else

{

String tempHouse = "";

String tempBranch = "";

String tempLocation = "";

String tempCity = "";

String tempType = "";

String house[];

String branch\_name[];

String location[];

String city[];

String houseType[];

ResultSet rs = stmt.executeQuery("select distinct house.houseID, branch\_name, location, city\_name, house.huose\_Type from house, branch, city, location where branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type = '"+category.getValue().toString()+"';");

while(rs.next())

{

tempHouse = tempHouse + rs.getString(1) + "#";

tempBranch = tempBranch + rs.getString(2) + "#";

tempLocation = tempLocation + rs.getString(3) + "#";

tempCity = tempCity + rs.getString(4) + "#";

tempType = tempType + rs.getString(5) + "#";

}

house = tempHouse.split("#");

branch\_name = tempBranch.split("#");

location = tempLocation.split("#");

city = tempCity.split("#");

houseType = tempType.split("#");

for(int i = 0; i < house.length; i++)

{

rs = stmt.executeQuery("SELECT booking, count(\*) as Total\_Bookees from customer\_record where booking = 'h"+house[i]+"' group by booking;");

if(rs.next())

{

list.add(new HouseModalTable("House No." + house[i], branch\_name[i], location[i], city[i], houseType[i], rs.getString(2)));

}

else

{

list.add(new HouseModalTable("House No." + house[i], branch\_name[i], location[i], city[i], houseType[i], "0"));

}

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("HouseMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb){

// TODO

branch.getItems().addAll("ISB Branch 1", "ISB Branch 2", "PEW Branch 1", "PEW Branch 2", "LHR Branch", "KHI Branch");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

String branchID = "";

String branchName = "";

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branchID from employee where empID = '"+ empID +"';");

while(rs.next())

{

branchID = rs.getString(1);

}

rs = stmt.executeQuery("select branch\_name from branch where branchID = '" + branchID + "';");

while(rs.next())

{

branchName = rs.getString(1);

}

}

catch(SQLException e)

{

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

}

branch.setValue(branchName);

category.setValue("ALL");

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

logo = new ImageView("logo1.png");

}

}

## House Controller:

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

public class HouseController implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private ImageView viewHouse;

@FXML private ImageView bookHouse;

String usernameOfLoggedInPerson;

String cityID;

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

}

public void availableHouseButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableHousesController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void availableHouseImagePushed(MouseEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableHousesController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void bookHouseButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerProfileController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void bookHouseImagePushed(MouseEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BookHousesController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerProfileController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML private Label label;

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("bookHouse.png");

viewHouse = new ImageView("ViewAvailableHouses.png");

bookHouse = new ImageView("house.png");

}

}

## House Main Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class HouseMainController implements Initializable {

@FXML private ImageView logo;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void allHousesButtonPressed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AllHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AllHousesController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void bookedHousesButtonPressed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookedHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BookedHousesController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void availableHousesButtonPressed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableHousesFromMainHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableHousesFromMainHousesController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void allHouseBookesButtonPressed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("HouseBookees.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseBookeesController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ReceptionistProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## House Modal Table:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import javafx.beans.property.SimpleStringProperty;

public class HouseModalTable

{

private SimpleStringProperty srNo;

private SimpleStringProperty branchName;

private SimpleStringProperty location;

private SimpleStringProperty city;

private SimpleStringProperty category;

private SimpleStringProperty status;

private SimpleStringProperty address;

private SimpleStringProperty booking;

private SimpleStringProperty leavingDate;

private SimpleStringProperty bookingDate;

public HouseModalTable(String srNo, String branchName, String location, String city, String category, String status) {

this.srNo = new SimpleStringProperty(srNo);

this.branchName = new SimpleStringProperty(branchName);

this.location = new SimpleStringProperty(location);

this.city = new SimpleStringProperty(city);

this.category = new SimpleStringProperty(category);

this.status = new SimpleStringProperty(status);

}

public HouseModalTable(String srNo, String booking, String address, String bookingDate, String leavingDate)

{

this.srNo = new SimpleStringProperty(srNo);

this.booking = new SimpleStringProperty(booking);

this.address = new SimpleStringProperty(address);

this.bookingDate = new SimpleStringProperty(bookingDate);

this.leavingDate = new SimpleStringProperty(leavingDate);

}

public HouseModalTable(String srNo, String branchName)

{

this.srNo = new SimpleStringProperty(srNo);

this.branchName = new SimpleStringProperty(branchName);

}

public String getLocation() {

return location.get();

}

public void setLocation(String location) {

this.location = new SimpleStringProperty(location);

}

public String getBooking() {

return booking.get();

}

public void setBooking(String booking) {

this.booking = new SimpleStringProperty(booking);

}

public String getAddress() {

return address.get();

}

public void setAddress(String address) {

this.address = new SimpleStringProperty(address);

}

public String getBookingDate() {

return bookingDate.get();

}

public void setBookingDate(String bookingDate) {

this.bookingDate = new SimpleStringProperty(bookingDate);

}

public String getLeavingDate() {

return leavingDate.get();

}

public void setLeavingDate(String leavingDate) {

this.leavingDate = new SimpleStringProperty(leavingDate);

}

public String getCity() {

return city.get();

}

public void setCity(String city) {

this.city = new SimpleStringProperty(city);

}

public String getCategory() {

return category.get();

}

public void setCategory(String category) {

this.category = new SimpleStringProperty(category);

}

public String getStatus() {

return status.get();

}

public void setStatus(String status) {

this.status = new SimpleStringProperty(status);

}

public String getSrNo() {

return srNo.get();

}

public void setSrNo(String srNo) {

this.srNo = new SimpleStringProperty(srNo);

}

public String getBranchName() {

return branchName.get();

}

public void setBranchName(String branchName) {

this.branchName = new SimpleStringProperty(branchName);

}

}

## Leave Detailed Customer Info:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class LeaveDetailedCustomerInfoController implements Initializable {

@FXML private ImageView logo;

@FXML private Label name;

@FXML private Label gender;

@FXML private Label dob;

@FXML private Label cnic;

@FXML private Label username;

@FXML private Label joiningDate;

@FXML private Label leavingDate;

@FXML private Label mob1;

@FXML private Label mob2;

@FXML private Label mob3;

@FXML private Label address1\_1;

@FXML private Label address1\_2;

@FXML private Label address2\_1;

@FXML private Label address2\_2;

@FXML private Label address3\_1;

@FXML private Label address3\_2;

String empID = "";

String customerCNIC = "";

String customerID = "";

String prevPageID = "";

public void setParameter(String empID, String c)throws SQLException

{

this.empID = empID;

this.customerCNIC = c;

setDetailDataOfUser();

}

public void setParameter(String empID, Customer c) throws SQLException

{

this.empID = empID;

this.customerCNIC = c.getCnic();

setDetailDataOfUser();

}

public void setParameter(String empID, Customer c, String s) throws SQLException

{

this.empID = empID;

this.customerCNIC = c.getCnic();

this.prevPageID = s;

setDetailDataOfUser();

}

public void setParameter(String empID, String c, String s) throws SQLException

{

this.empID = empID;

this.customerCNIC = c;

this.prevPageID = s;

setDetailDataOfUser();

}

public void setDetailDataOfUser() throws SQLException

{

System.out.println("Customer CNIC from Account : "+customerCNIC);

cnic.setText(customerCNIC);

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

//ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, CNIC, joining\_date from customer where email = '" + s + "';");

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, email, joining\_date, leaving\_date, customer\_deleted.customerID from customer\_deleted where customer\_deleted.CNIC = '" + customerCNIC + "';");

boolean temp = rs.next();

name.setText(rs.getString(1) + " " + rs.getString(2));

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

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

System.out.println("\nDate Of Birth : "+dob.getText());

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

joiningDate.setText(rs.getString(6));

leavingDate.setText(rs.getString(7));

customerID = rs.getString(8);

//for Age we Do,

rs = stmt.executeQuery("select distinct address, city\_name, country\_name from customer\_deleted, address, customer\_address, country, city where customer\_address.customerID = customer\_deleted.customerID and customer\_address.addressID = address.addressID and address.City\_cityID = city.cityID and city.countryID = country.countryID and customer\_deleted.CNIC = '"+ customerCNIC +"';");

temp = rs.next();

address1\_1.setText(rs.getString(1) + ", ");

address1\_2.setText(rs.getString(2) + ", " + rs.getString(3));

int counter = 1;

while(rs.next())

{

if(counter == 1)

{

address2\_1.setText(rs.getString(1) + ", ");

address2\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

else

{

address3\_1.setText(rs.getString(1) + ", ");

address3\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

counter++;

}

rs = stmt.executeQuery("select distinct phone\_number from customer\_deleted, customer\_phone where customer\_deleted.customerID = customer\_phone.customerID and customer\_deleted.CNIC = '"+customerCNIC+"';");

temp = rs.next();

mob1.setText(rs.getString(1));

counter = 1;

while(rs.next())

{

if(counter == 1)

{

mob2.setText(rs.getString(1));

}

else

{

mob3.setText(rs.getString(1));

}

counter++;

}

}

public void seeBookingButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("LeavedCustomerBooking.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

LeavedCustomerBookingController controller = loader.getController();

controller.setParameter(empID, customerCNIC, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("LeavedCustomer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

LeavedCustomerController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Leave Employee Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class LeaveEmployeeController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<Employee> tableView;

@FXML private TableColumn<Employee, String> srNoC;

@FXML private TableColumn<Employee, String> customerNameC;

@FXML private TableColumn<Employee, String> cnicC;

@FXML private ComboBox combo;

@FXML private TextField searchBar;

@FXML private Button search;

@FXML private Button DetailView;

@FXML private Label label;

@FXML private Label label2;

int total\_customer = 0;

String empID = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from empoyeeleft;");

while(rs.next())

total\_customer++;

label.setText("TOTAL Deleted Employees : " + total\_customer);

setTableData();

}

public void userClickOnTable()

{

this.DetailView.setDisable(false);

}

public void searchButtonPushed() throws SQLException

{

ObservableList<Employee> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(combo.getValue().toString().equals("NAME"))

{

int counter = 1;

String name[] = searchBar.getText().split(" ");

String firstName = "";

String lastName = "";

if(name.length > 1)

{

firstName = name[0];

lastName = name[1];

}

else if(name.length == 1)

{

firstName = name[0];

}

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from empoyeeleft where first\_name = '"+ firstName +"' and last\_name = '"+ lastName +"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("FIRST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from empoyeeleft where first\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered First Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("LAST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from empoyeeleft where last\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Enterd Last Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CNIC"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from empoyeeleft where CNIC = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered CNIC is Not Present in our Database");

}

else if(combo.getValue().toString().equals("EMPLOYEE ID"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from empoyeeleft where empID = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Employee ID is Not Present in our Database");

}

else

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from empoyeeleft;");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

}

tableView.setItems(list);

}

public void setTableData() throws SQLException

{

ObservableList<Employee> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from empoyeeleft;");

while(rs.next())

{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}

tableView.setItems(list);

}

public void detailedViewButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("LeavedEmployee2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

LeavedEmployee2Controller controller = loader.getController();

controller.setParameter(empID, tableView.getSelectionModel().getSelectedItem());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("EmpployeeMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

EmpployeeMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

DetailView.setDisable(true);

combo.getItems().addAll("NAME", "FIRST NAME", "LAST NAME","CNIC", "EMPLOYEE ID", "Show All Data");

combo.setValue("Enter Your Desired Input for Search ");

srNoC.setCellValueFactory(new PropertyValueFactory<Employee, String>("srNo"));

customerNameC.setCellValueFactory(new PropertyValueFactory<Employee, String>("employeeName"));

cnicC.setCellValueFactory(new PropertyValueFactory<Employee, String>("cnic"));

}

}

## Leave Customer Booking Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class LeavedCustomerBookingController implements Initializable {

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> srNoC;

@FXML private TableColumn<HouseModalTable, String> booking; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> address;

@FXML private TableColumn<HouseModalTable, String> bookingDate;

@FXML private TableColumn<HouseModalTable, String> leavingDate;

@FXML private ComboBox combo;

String empID = "";

String customerCNIC = "";

String prevPageID = "";

public void setParameter(String empID, String customerCNIC) throws SQLException

{

this.empID = empID;

this.customerCNIC = customerCNIC;

setTableDataUponCombo();

}

public void setParameter(String empID, String customerCNIC, String s) throws SQLException

{

this.empID = empID;

this.customerCNIC = customerCNIC;

this.prevPageID = s;

setTableDataUponCombo();

}

public void setTableDataUponCombo() throws SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("Select customerID from customer\_deleted where CNIC = '" + customerCNIC + "';");

boolean temp = rs.next();

String customerID = rs.getString(1);

if(combo.getValue().toString().equals("LUXURY"))

{

boolean check = false;

String houseID[];

String tempHouse = "";

String address[];

String tempAddress = "";

rs = stmt.executeQuery("select house.houseID, branch.branch\_name, location.location from house, house\_booking, branch, location where house.houseID = house\_booking.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and house\_booking.customerID = '"+customerID+"' and house.huose\_Type = 'Luxury';");

while(rs.next())

{

check = true;

tempHouse = tempHouse + rs.getString(1) + "#";

tempAddress = tempAddress + rs.getString(2) + ", " + rs.getString(3) + "#";

}

houseID = tempHouse.split("#");

address = tempAddress.split("#");

String booking[];

String leaving[];

String tempBooking = "";

String tempLeaving = "";

for(int i = 0; i < houseID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'h" + houseID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

int counter = 1;

if(check)

{

for(int i = 0; i < houseID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "House." + houseID[i], address[i], booking[i], leaving[i]));

counter++;

}

}

//=============================================================================================================================================================

check = false;

String tempForEmpty[] = new String[0];

houseID = tempForEmpty;

address = tempForEmpty;

booking = tempForEmpty;

leaving = tempForEmpty;

String room[]; //Room Number Not the Room ID

String roomID[];

tempHouse = "";

tempAddress = "";

tempBooking = "";

tempLeaving = "";

String tempRoomNo = "";

String tempRoomID = "";

rs = stmt.executeQuery("select room.room\_number, room.roomID, house.houseID, branch.branch\_name, location.location from room, house, room\_booking, branch, location where room.houseID = house.houseID and room.roomID = room\_booking.roomID and house.branchID = branch.branchID and branch.locationID = location.locationID and room\_booking.customerID = '"+customerID+"' and house.huose\_Type = 'roomsOnlyLuxury';");

while(rs.next())

{

check = true;

tempRoomNo = tempRoomNo + rs.getString(1) + "#";

tempRoomID = tempRoomID + rs.getString(2) + "#";

tempHouse = tempHouse + rs.getString(3) + "#";

tempAddress = tempAddress + rs.getString(4) + ", " + rs.getString(5) + "#";

}

room = tempRoomNo.split("#");

roomID = tempRoomID.split("#");

houseID = tempHouse.split("#");

address = tempAddress.split("#");

for(int i = 0; i < roomID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'r" + roomID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

if(check)

{

for(int i = 0; i < roomID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "Room No.(" + roomID[i] + ") HouseID (" + houseID[i] + ")", address[i], booking[i], leaving[i]));

counter++;

}

}

}

else if(combo.getValue().toString().equals("ECONOMY"))

{

boolean check = false;

String houseID[];

String tempHouse = "";

String address[];

String tempAddress = "";

rs = stmt.executeQuery("select house.houseID, branch.branch\_name, location.location from house, house\_booking, branch, location where house.houseID = house\_booking.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and house\_booking.customerID = '"+customerID+"' and house.huose\_Type = 'Economy';");

while(rs.next())

{

check = true;

tempHouse = tempHouse + rs.getString(1) + "#";

tempAddress = tempAddress + rs.getString(2) + ", " + rs.getString(3) + "#";

}

houseID = tempHouse.split("#");

address = tempAddress.split("#");

String booking[];

String leaving[];

String tempBooking = "";

String tempLeaving = "";

for(int i = 0; i < houseID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'h" + houseID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

int counter = 1;

if(check)

{

for(int i = 0; i < houseID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "House." + houseID[i], address[i], booking[i], leaving[i]));

counter++;

}

}

//=============================================================================================================================================================

check = false;

String tempForEmpty[] = new String[0];

houseID = tempForEmpty;

address = tempForEmpty;

booking = tempForEmpty;

leaving = tempForEmpty;

String room[]; //Room Number Not the Room ID

String roomID[];

tempHouse = "";

tempAddress = "";

tempBooking = "";

tempLeaving = "";

String tempRoomNo = "";

String tempRoomID = "";

rs = stmt.executeQuery("select room.room\_number, room.roomID, house.houseID, branch.branch\_name, location.location from room, house, room\_booking, branch, location where room.houseID = house.houseID and room.roomID = room\_booking.roomID and house.branchID = branch.branchID and branch.locationID = location.locationID and room\_booking.customerID = '"+customerID+"' and house.huose\_Type = 'roomsOnlyEconomy';");

while(rs.next())

{

check = true;

tempRoomNo = tempRoomNo + rs.getString(1) + "#";

tempRoomID = tempRoomID + rs.getString(2) + "#";

tempHouse = tempHouse + rs.getString(3) + "#";

tempAddress = tempAddress + rs.getString(4) + ", " + rs.getString(5) + "#";

}

room = tempRoomNo.split("#");

roomID = tempRoomID.split("#");

houseID = tempHouse.split("#");

address = tempAddress.split("#");

for(int i = 0; i < roomID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'r" + roomID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

if(check)

{

for(int i = 0; i < roomID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "Room No.(" + roomID[i] + ") HouseID (" + houseID[i] + ")", address[i], booking[i], leaving[i]));

counter++;

}

}

}

else

{

boolean check = false;

String houseID[];

String tempHouse = "";

String address[];

String tempAddress = "";

rs = stmt.executeQuery("select house.houseID, branch.branch\_name, location.location from house, house\_booking, branch, location where house.houseID = house\_booking.houseID and house.branchID = branch.branchID and branch.locationID = location.locationID and house\_booking.customerID = '"+customerID+"';");

while(rs.next())

{

check = true;

tempHouse = tempHouse + rs.getString(1) + "#";

tempAddress = tempAddress + rs.getString(2) + ", " + rs.getString(3) + "#";

}

houseID = tempHouse.split("#");

address = tempAddress.split("#");

String booking[];

String leaving[];

String tempBooking = "";

String tempLeaving = "";

for(int i = 0; i < houseID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'h" + houseID[i] + "';");

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

int counter = 1;

if(check)

{

for(int i = 0; i < houseID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "House." + houseID[i], address[i], booking[i], leaving[i]));

counter++;

}

}

//=============================================================================================================================================================

check = false;

String tempForEmpty[] = new String[0];

houseID = tempForEmpty;

address = tempForEmpty;

booking = tempForEmpty;

leaving = tempForEmpty;

String room[]; //Room Number Not the Room ID

String roomID[];

tempHouse = "";

tempAddress = "";

tempBooking = "";

tempLeaving = "";

String tempRoomNo = "";

String tempRoomID = "";

rs = stmt.executeQuery("select room.room\_number, room.roomID, house.houseID, branch.branch\_name, location.location from room, house, room\_booking, branch, location where room.houseID = house.houseID and room.roomID = room\_booking.roomID and house.branchID = branch.branchID and branch.locationID = location.locationID and room\_booking.customerID = '"+customerID+"';");

while(rs.next())

{

check = true;

tempRoomNo = tempRoomNo + rs.getString(1) + "#";

tempRoomID = tempRoomID + rs.getString(2) + "#";

tempHouse = tempHouse + rs.getString(3) + "#";

tempAddress = tempAddress + rs.getString(4) + ", " + rs.getString(5) + "#";

}

room = tempRoomNo.split("#");

roomID = tempRoomID.split("#");

houseID = tempHouse.split("#");

address = tempAddress.split("#");

for(int i = 0; i < roomID.length; i++)

{

rs = stmt.executeQuery("select booking\_date, leaving\_date from customer\_record where customer\_record.customerID = '"+customerID+"' and booking = 'r" + roomID[i] + "';");

System.out.println("room ID from All (else) : " + roomID[i]);

while(rs.next())

{

tempBooking = tempBooking + rs.getString(1) + "#";

tempLeaving = tempLeaving + rs.getString(2) + "#";

}

}

booking = tempBooking.split("#");

leaving = tempLeaving.split("#");

if(check)

{

for(int i = 0; i < roomID.length; i++)

{

list.add(new HouseModalTable(Integer.toString(counter), "Room No.(" + roomID[i] + ") HouseID (" + houseID[i] + ")", address[i], booking[i], leaving[i]));

counter++;

}

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("LeaveDetailedCustomerInfo.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

LeaveDetailedCustomerInfoController controller = loader.getController();

controller.setParameter(empID, customerCNIC, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

combo.getItems().addAll("ALL", "LUXURY", "ECONOMY");

combo.setValue("ALL");

srNoC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

booking.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("booking"));

address.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("address"));

bookingDate.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("bookingDate"));

leavingDate.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("leavingDate"));

}

}

## Leaved Customer Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class LeavedCustomerController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<Customer> tableView;

@FXML private TableColumn<Customer, String> srNoC;

@FXML private TableColumn<Customer, String> customerNameC;

@FXML private TableColumn<Customer, String> cnicC;

@FXML private ComboBox combo;

@FXML private TextField searchBar;

@FXML private Button search;

@FXML private Button DetailView;

@FXML private Label label;

@FXML private Label label2;

int total\_customer = 0;

String empID = "";

String prevPageID = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from customer\_deleted;");

while(rs.next())

total\_customer++;

label.setText("TOTAL CUSTOMERS : " + total\_customer);

setTableData();

}

public void setParameter(String empID, String s) throws SQLException

{

this.empID = empID;

prevPageID = s;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from customer\_deleted;");

while(rs.next())

total\_customer++;

label.setText("TOTAL CUSTOMERS : " + total\_customer);

setTableData();

}

public void userClickOnTable()

{

this.DetailView.setDisable(false);

}

public void searchButtonPushed() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(combo.getValue().toString().equals("NAME"))

{

int counter = 1;

String name[] = searchBar.getText().split(" ");

String firstName = "";

String lastName = "";

if(name.length > 1)

{

firstName = name[0];

lastName = name[1];

}

else if(name.length == 1)

{

firstName = name[0];

}

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer\_deleted where first\_name = '"+ firstName +"' and last\_name = '"+ lastName +"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("FIRST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer\_deleted where first\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered First Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("LAST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer\_deleted where last\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Enterd Last Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CNIC"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer\_deleted where CNIC = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered CNIC is Not Present in our Database");

}

else if(combo.getValue().toString().equals("E-MAIL"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer\_deleted where email = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered E-MAIL is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CUSTOMER ID"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer\_deleted where customerID = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Customer ID is Not Present in our Database");

}

else

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer\_deleted;");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

}

tableView.setItems(list);

}

public void setTableData() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer\_deleted;");

while(rs.next())

{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}

tableView.setItems(list);

}

public void detailedViewButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("LeaveDetailedCustomerInfo.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

LeaveDetailedCustomerInfoController controller = loader.getController();

controller.setParameter(empID, tableView.getSelectionModel().getSelectedItem(), prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Customer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

DetailView.setDisable(true);

combo.getItems().addAll("NAME", "FIRST NAME", "LAST NAME","CNIC", "E-MAIL", "CUSTOMER ID", "Show All Data");

combo.setValue("Enter Your Desired Input for Search ");

srNoC.setCellValueFactory(new PropertyValueFactory<Customer, String>("srNo"));

customerNameC.setCellValueFactory(new PropertyValueFactory<Customer, String>("customerName"));

cnicC.setCellValueFactory(new PropertyValueFactory<Customer, String>("cnic"));

}

}

## Leaved Employee Controller 2:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class LeavedEmployee2Controller implements Initializable {

@FXML private ImageView logo;

@FXML private Label name;

@FXML private Label gender;

@FXML private Label dob;

@FXML private Label cnic;

@FXML private Label leavingDate;

@FXML private Label salary;

@FXML private Label joiningDate;

@FXML private Label mob1;

@FXML private Label mob2;

@FXML private Label mob3;

@FXML private Label address1\_1;

@FXML private Label address1\_2;

@FXML private Label address2\_1;

@FXML private Label address2\_2;

@FXML private Label address3\_1;

@FXML private Label address3\_2;

String empID = "";

String employeeCNIC = "";

String employeeID = "";

public void setParameter(String empID, String c)

{

this.empID = empID;

this.employeeCNIC = c;

setDetailDataOfUser();

}

public void setParameter(String empID, Employee c)

{

this.empID = empID;

this.employeeCNIC = c.getCnic();

setDetailDataOfUser();

}

public void setDetailDataOfUser()

{

try

{

System.out.println("Customer CNIC from Account : "+employeeCNIC);

cnic.setText(employeeCNIC);

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

//ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, CNIC, joining\_date from customer where email = '" + s + "';");

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, gender, DOB, salary, joiningDate, leavingDate,empoyeeleft.empID from empoyeeleft where empoyeeleft.CNIC = '" + employeeCNIC + "';");

boolean temp = rs.next();

name.setText(rs.getString(1) + " " + rs.getString(2));

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

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

System.out.println("\nDate Of Birth : "+dob.getText());

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

joiningDate.setText(rs.getString(6));

leavingDate.setText(rs.getString(7));

String employeeID = rs.getString(8);

rs = stmt.executeQuery("select distinct address, city\_name, country\_name from empoyeeleft, address, address\_has\_employee, country, city where address\_has\_employee.empID = empoyeeleft.empID and address\_has\_employee.addressID = address.addressID and address.City\_cityID = city.cityID and city.countryID = country.countryID and empoyeeleft.CNIC = '"+ employeeCNIC +"';");

temp = rs.next();

address1\_1.setText(rs.getString(1) + ", ");

address1\_2.setText(rs.getString(2) + ", " + rs.getString(3));

int counter = 1;

while(rs.next())

{

if(counter == 1)

{

address2\_1.setText(rs.getString(1) + ", ");

address2\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

else

{

address3\_1.setText(rs.getString(1) + ", ");

address3\_2.setText(rs.getString(2) + ", " + rs.getString(3));

}

counter++;

}

rs = stmt.executeQuery("select distinct phone\_number from empoyeeleft, employee\_phone where empoyeeleft.empID = employee\_phone.empID and empoyeeleft.CNIC = '"+employeeCNIC+"';");

temp = rs.next();

mob1.setText(rs.getString(1));

counter = 1;

while(rs.next())

{

if(counter == 1)

{

mob2.setText(rs.getString(1));

}

else

{

mob3.setText(rs.getString(1));

}

counter++;

}

}

catch(Exception e)

{

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

}

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("LeaveEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

LeaveEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Modify Employee Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ModifyEmployee1Controller implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<Employee> tableView;

@FXML private TableColumn<Employee, String> srNoC;

@FXML private TableColumn<Employee, String> customerNameC;

@FXML private TableColumn<Employee, String> cnicC;

@FXML private ComboBox combo;

@FXML private TextField searchBar;

@FXML private Button search;

@FXML private Button DetailView;

@FXML private Label label;

@FXML private Label label2;

int total\_customer = 0;

String empID = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from employee;");

while(rs.next())

total\_customer++;

label.setText("TOTAL Employees : " + total\_customer);

setTableData();

}

public void userClickOnTable()

{

this.DetailView.setDisable(false);

}

public void searchButtonPushed() throws SQLException

{

ObservableList<Employee> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(combo.getValue().toString().equals("NAME"))

{

int counter = 1;

String name[] = searchBar.getText().split(" ");

String firstName = "";

String lastName = "";

if(name.length > 1)

{

firstName = name[0];

lastName = name[1];

}

else if(name.length == 1)

{

firstName = name[0];

}

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where first\_name = '"+ firstName +"' and last\_name = '"+ lastName +"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("FIRST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where first\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered First Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("LAST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where last\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Enterd Last Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CNIC"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where CNIC = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered CNIC is Not Present in our Database");

}

else if(combo.getValue().toString().equals("EMPLOYEE ID"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee where employeeID = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered EMPLOYEE ID is Not Present in our Database");

}

else

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee;");

if(rs.next())

{

do{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

}

tableView.setItems(list);

}

public void setTableData() throws SQLException

{

ObservableList<Employee> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from employee;");

while(rs.next())

{

list.add(new Employee(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}

tableView.setItems(list);

}

public void detailedViewButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ModifyEmployee2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ModifyEmployee2Controller controller = loader.getController();

controller.setParameter(empID, tableView.getSelectionModel().getSelectedItem());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("EmpployeeMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

EmpployeeMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

DetailView.setDisable(true);

combo.getItems().addAll("NAME", "FIRST NAME", "LAST NAME","CNIC", "EMPLOYEE ID", "Show All Data");

combo.setValue("Enter Your Desired Input for Search ");

srNoC.setCellValueFactory(new PropertyValueFactory<Employee, String>("srNo"));

customerNameC.setCellValueFactory(new PropertyValueFactory<Employee, String>("employeeName"));

cnicC.setCellValueFactory(new PropertyValueFactory<Employee, String>("cnic"));

}

}

## Modify Employee 2 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.time.LocalDate;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.RadioButton;

import javafx.scene.control.TextField;

import javafx.scene.control.ToggleGroup;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ModifyEmployee2Controller implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private TextField firstName;

@FXML private TextField lastName;

@FXML private DatePicker dob;

@FXML private TextField cnic;

//For Radio Button

@FXML private RadioButton male;

@FXML private RadioButton female;

@FXML private ToggleGroup favLang; //this is for grouping the upper 2 radio button, so that only one button can be selected at a time

@FXML private TextField mob1;

@FXML private TextField mob2;

@FXML private TextField mob3;

@FXML private TextField address1\_city;

@FXML private TextField address1\_country;

@FXML private TextField address1\_address;

@FXML private TextField address2\_city;

@FXML private TextField address2\_country;

@FXML private TextField address2\_address;

@FXML private TextField address3\_city;

@FXML private TextField address3\_country;

@FXML private TextField address3\_address;

String prevAddress1ID = "";

String prevAddress2ID = "";

String prevAddress3ID = "";

String prevMob1 = "";

String prevMob2 = "";

String prevMob3 = "";

@FXML private Label label;

String CNIC = "";

String empID = "";

String gender = "";

String employeeID = "";

public void setParameter(String empID, Employee s)

{

try

{

this.empID = empID;

CNIC = s.getCnic();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String employeeID = "";

ResultSet rs = stmt.executeQuery("select \* from employee where CNIC = '"+CNIC+"';");

if(rs.next())

{

employeeID = rs.getString(1);

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

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

cnic.setText(rs.getString(6));

dob.setValue(Checks.stringToLocalDateConverter(rs.getString(5)));

gender = rs.getString(4);

if(gender.equals("Male"))

favLang.selectToggle(male);

else

favLang.selectToggle(female);

}

int counter = 1;

rs = stmt.executeQuery("select distinct address, city.city\_name, country.country\_name, address.addressID from address, city, country, employee, address\_has\_employee where employee.empID = address\_has\_employee.empID and address.addressID = address\_has\_employee.addressID and address.City\_cityID = city.cityID and city.countryID = country.countryID and employee.CNIC = '"+CNIC+"';;");

while(rs.next())

{

if(counter == 1)

{

address1\_address.setText(rs.getString(1));

address1\_city.setText(rs.getString(2));

address1\_country.setText(rs.getString(3));

prevAddress1ID = rs.getString(4);

}

else if(counter == 2)

{

address2\_address.setVisible(true);

address2\_city.setVisible(true);

address2\_country.setVisible(true);

address2\_address.setText(rs.getString(1));

address2\_city.setText(rs.getString(2));

address2\_country.setText(rs.getString(3));

prevAddress2ID = rs.getString(4);

}

else

{

address3\_address.setVisible(true);

address3\_city.setVisible(true);

address3\_country.setVisible(true);

address3\_address.setText(rs.getString(1));

address3\_city.setText(rs.getString(2));

address3\_country.setText(rs.getString(3));

prevAddress3ID = rs.getString(4);

}

counter++;

}

counter = 1;

rs = stmt.executeQuery("select phone\_number from employee natural join employee\_phone where employee.CNIC = '"+CNIC+"';");

while(rs.next())

{

if(counter == 1)

{

mob1.setText(rs.getString(1));

prevMob1 = mob1.getText();

}

else if(counter == 2)

{

mob2.setVisible(true);

mob2.setText(rs.getString(1));

prevMob2 = mob2.getText();

}

else

{

mob3.setVisible(true);

mob3.setText(rs.getString(1));

prevMob3 = mob3.getText();

}

counter++;

}

}

catch(Exception e)

{

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

}

}

public boolean cnicCheckData(String cnic) //this method verify the unity of cnic Entered By User

{

boolean check = true; //return false when cnic

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select CNIC from employee where CNIC = '" + cnic + "';");

if(rs.next())

check = false;

}

catch(Exception ex)

{

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

}

return check;

}

public boolean cnicCheck(String s) throws SQLException

{

boolean check = false;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String cnic = "";

ResultSet rs = stmt.executeQuery("select CNIC from employee where CNIC = '"+CNIC+"';");

if(rs.next())

cnic = rs.getString(1);

if(cnic.equals(s))

check = true;

return check;

}

public void updateData(ActionEvent event) throws IOException, SQLException

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(Checks.alphaCheckIgnoreSpace(firstName.getText()) && Checks.alphaCheckIgnoreSpace(lastName.getText()))

{

if(Checks.cnicCheck(cnic.getText()))

{

if(cnicCheckData(cnic.getText()) || cnicCheck(cnic.getText()))

{

String date = dob.getValue().toString();

String dateData[] = date.split("-");

System.out.println("Year : "+ dateData[0]);

System.out.println("MOnth : "+ dateData[1]);

System.out.println("Day : "+ dateData[2]);

if(Checks.ageCheck(Integer.parseInt(dateData[0]), Integer.parseInt(dateData[1]), Integer.parseInt(dateData[2])))

{

//System.out.println(favLang.getSelectedToggle().toString());

if(favLang.getSelectedToggle().equals(male))

gender = "Male";

else

gender = "Female";

System.out.println(gender);

String sql = "UPDATE `mydb`.`employee` SET `first\_name` = '"+firstName.getText()+"', `last\_name` = '"+lastName.getText()+"', `gender` = '"+gender+"', `DOB` = '"+date+"', `CNIC` = '"+cnic.getText()+"' WHERE (`CNIC` = '"+CNIC+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

CNIC = cnic.getText();

if(address1\_address.getText().equals("") || address1\_city.getText().equals("") || address1\_country.getText().equals(""))

label.setText("You Cannot Leave any Address Field Empty!");

else

addressUpdate();

if(mob1.getText().equals(""))

label.setText("You Cannot Leave Mobile Field Empty!");

else

mobileUpdate();

saveAlert(event);

//MObile update cpmes here

}

else

label.setText("Your Age shoudn't be Under 18 !");

}

else

label.setText("Your Entered CNIC is already registered!");

}

else

label.setText("Please Enter CNIC in Format like '12345-1234567-1' !");

}

else

label.setText("First Name should only consist of Alphabets !");

}

public void addressUpdate() throws SQLException

{

System.out.println("=====================================================================================");

System.out.println("=========================== ADDRESSS 1 =============================================");

System.out.println("=====================================================================================");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String customerID = "";

ResultSet rs = stmt.executeQuery("Select empID from employee where CNIC = '"+CNIC+"';");

if(rs.next())

employeeID = rs.getString(1);

String countryID = "";

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address1\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

else

{

String sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+address1\_country.getText()+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address1\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

}

System.out.println("CountryID : " + countryID);

String cityID = "";

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address1\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

String sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+address1\_city.getText()+"', '"+countryID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address1\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

}

System.out.println("City ID : " + cityID);

String addressID = "";

rs = stmt.executeQuery("Select addressID from address where address = '"+address1\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

String sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+address1\_address.getText()+"', '"+cityID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select addressID from address where address = '"+address1\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

}

System.out.println("Employee ID : " +employeeID);

System.out.println("AddressID : " +addressID);

System.out.println("Previous Address ID : " + prevAddress1ID);

String sql = "UPDATE `mydb`.`address\_has\_employee` SET `addressID` = '"+addressID+"' WHERE (`addressID` = '"+prevAddress1ID+"') and (`empID` = '"+employeeID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

System.out.println("=====================================================================================");

System.out.println("=========================== ADDRESSS 2 =============================================");

System.out.println("=====================================================================================");

if(address2\_address.isVisible())

{

if(address1\_address.getText().equals("") || address1\_city.getText().equals("") || address1\_country.getText().equals(""))

label.setText("You Cannot Leave Previous Address Field Empty !");

else

{

countryID = "";

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address2\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+address2\_country.getText()+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address2\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

}

System.out.println("Country ID : " + countryID);

cityID = "";

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address2\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+address2\_city.getText()+"', '"+countryID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address2\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

}

System.out.println("City ID : " + cityID);

addressID = "";

rs = stmt.executeQuery("Select addressID from address where address = '"+address2\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+address2\_address.getText()+"', '"+cityID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select addressID from address where address = '"+address2\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

}

System.out.println("Employee ID : " + employeeID);

System.out.println("Address ID : " + addressID);

System.out.println("Previous AddressID : " + prevAddress2ID);

if(prevAddress2ID.equals(""))

{

sql = "INSERT INTO `mydb`.`address\_has\_employee` (`addressID`, `empID`) VALUES ('"+addressID+"', '"+employeeID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

}

else

{

sql = "UPDATE `mydb`.`address\_has\_employee` SET `addressID` = '"+addressID+"' WHERE (`addressID` = '"+prevAddress2ID+"') and (`empID` = '"+employeeID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

}

}

}

System.out.println("=====================================================================================");

System.out.println("============================== ADDRESSS 3 ==========================================");

System.out.println("=====================================================================================");

if(address3\_address.isVisible() || !address3\_address.getText().equals(""))

{

System.out.println("Inside of address 3");

if(address2\_address.getText().equals("") || address2\_city.getText().equals("") || address2\_country.getText().equals(""))

label.setText("You Cannot Leave Previous Address Field Empty !");

else

{

System.out.println("Purely Inside of Address 3");

countryID = "";

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address3\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+address3\_country.getText()+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select countryID from country where country\_name = '"+address3\_country.getText()+"';");

if(rs.next())

countryID = rs.getString(1);

}

System.out.println("Country ID : " + countryID);

cityID = "";

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address3\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+address3\_city.getText()+"', '"+countryID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("Select cityID from city where city\_name = '"+address3\_city.getText()+"' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

}

System.out.println("City ID : " + cityID);

addressID = "";

rs = stmt.executeQuery("Select addressID from address where address = '"+address3\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+address3\_address.getText()+"', '"+cityID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

rs = stmt.executeQuery("Select addressID from address where address = '"+address3\_address.getText()+"' and City\_cityID = '"+cityID+"';");

if(rs.next())

addressID = rs.getString(1);

}

System.out.println("Address ID : " + addressID);

System.out.println("CustomerID : " +customerID);

System.out.println("Previous Address ID : " + prevAddress3ID);

if(prevAddress3ID.equals(""))

{

sql = "INSERT INTO `mydb`.`address\_has\_employee` (`addressID`, `empID`) VALUES ('"+addressID+"', '"+employeeID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

}

else

{

sql = "UPDATE `mydb`.`address\_has\_employee` SET `addressID` = '"+addressID+"' WHERE (`addressID` = '"+prevAddress3ID+"') and (`empID` = '"+employeeID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

System.out.println("Affected Rows : " + rowsAffected);

}

}

}

}

public void mobileUpdate() throws SQLException

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String customerID = "";

ResultSet rs = stmt.executeQuery("Select empID from employee where CNIC = '"+CNIC+"';");

if(rs.next())

employeeID = rs.getString(1);

String sql = "UPDATE `mydb`.`employee\_phone` SET `phone\_number` = '"+mob1.getText()+"' WHERE (`phone\_number` = '"+prevMob1+"') and (`empID` = '"+employeeID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

if(mob2.isVisible())

{

if(prevMob2.equals(""))

{

sql = "INSERT INTO `mydb`.`employee\_phone` (`empID`, `phone\_number`) VALUES ('"+employeeID+"', '"+mob2.getText()+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

}

else

{

sql = "UPDATE `mydb`.`employee\_phone` SET `phone\_number` = '"+mob2.getText()+"' WHERE (`phone\_number` = '"+prevMob2+"') and (`empID` = '"+employeeID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

}

}

if(mob3.isVisible())

{

if(prevMob3.equals(""))

{

sql = "INSERT INTO `mydb`.`employee\_phone` (`empID`, `phone\_number`) VALUES ('"+employeeID+"', '"+mob3.getText()+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

}

else

{

sql = "UPDATE `mydb`.`employee\_phone` SET `phone\_number` = '"+mob3.getText()+"' WHERE (`phone\_number` = '"+prevMob3+"') and (`empID` = '"+employeeID+"');";

preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

rowsAffected = preparedStatement.executeUpdate();

}

}

}

public void addMobileNumber()

{

if(!mob2.isVisible() && !mob1.getText().equals(""))

mob2.setVisible(true);

else if(!mob3.isVisible() && !mob2.getText().equals(""))

mob3.setVisible(true);

else

{

if(mob2.isVisible())

label.setText("Already Visible");

else if(mob1.getText().equals(""))

label.setText("You Cannot Leave Mobile Number Field Empty");

if(mob3.isVisible())

label.setText("Already Visible");

else if(mob2.getText().equals(""))

label.setText("You Cannot Left Mobile Number Field Empty");

if(mob2.isVisible() && mob3.isVisible())

label.setText("You Can Only Add Three numbers");

}

}

public void addAddress()

{

if(address3\_address.isVisible())

label.setText("You Cannot Add more than 3 Addresses!");

if(address2\_address.isVisible())

{

if(address2\_address.getText().equals("") || address2\_city.getText().equals("") || address2\_country.getText().equals(""))

{

label.setText("You Cannot Leave Any previous Address Filed Empty ! ");

}

else

{

address3\_address.setVisible(true);

address3\_city.setVisible(true);

address3\_country.setVisible(true);

}

}

else

{

if(address1\_address.getText().equals("") || address1\_city.getText().equals("") || address1\_country.getText().equals(""))

{

label.setText("You Cannot Leave Any previous Address Filed Empty ! ");

}

else

{

address2\_address.setVisible(true);

address2\_city.setVisible(true);

address2\_country.setVisible(true);

}

}

}

public void saveAlert(ActionEvent event) throws IOException, SQLException //this is INFORMATION

{

Alert a1 = new Alert(Alert.AlertType.INFORMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a1.setTitle("Save Changes");

a1.setContentText("Your Changes have been Saved ! ");

//a1.setHeaderText("Basic ALERT 2 "); //you can also set header

a1.showAndWait();

backButtonPushed(event);

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ModifyEmployee1.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

ModifyEmployee1Controller controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

//System.out.println("Username in back button of Account : " + username.getText());

}

@Override

public void initialize(URL url, ResourceBundle rb)

{

logo = new ImageView("logo1.png");

logo2 = new ImageView("account.png");

favLang = new ToggleGroup();

male.setToggleGroup(favLang);

female.setToggleGroup(favLang);

address2\_address.setVisible(false);

address2\_city.setVisible(false);

address2\_country.setVisible(false);

address3\_address.setVisible(false);

address3\_city.setVisible(false);

address3\_country.setVisible(false);

mob2.setVisible(false);

mob3.setVisible(false);

// TODO

}

}

## Receptionist Account Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Optional;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.ButtonType;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ReceptionistAccountController implements Initializable {

@FXML private ImageView logo;

String empID = "";

String cnic = "";

String mob = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select CNIC, phone\_number from employee, employee\_phone where employee.empID = '"+empID+"';");

rs.next();

cnic = rs.getString(1);

mob = rs.getString(2);

}

public void logOutButtonPushed(ActionEvent event) throws IOException, SQLException

{

Alert a1 = new Alert(Alert.AlertType.CONFIRMATION); /\*Alert types \*

1) INFORMATION

2) WARNING

3) ERROR

4) CONFIRMATION\*/

a1.setTitle("LOGOUT");

a1.setContentText("ARE YOU SURE YOU WANT TO LOGOUT? ");

a1.setHeaderText(null); //you can also set header

Optional<ButtonType> result = a1.showAndWait();

if(result.get() == ButtonType.OK) //type can also be "OK" as well

{

System.out.println("OKAY");

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistLogin.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

{

System.out.println("Cancel");

}

}

public void changePasswordButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ForgetPassword2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ForgetPassword2Controller controller = loader.getController();

controller.setParameter("ReceptionistAccount", empID, cnic, mob);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ReceptionistProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Receptionist Login Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.PrintWriter;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import java.util.Scanner;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.CheckBox;

import javafx.scene.control.Label;

import javafx.scene.control.PasswordField;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ReceptionistLoginController implements Initializable {

@FXML private Label label;

/\*\*

\* Initializes the controller class.

\*/

@FXML private ImageView msgIcon;

@FXML private ImageView lockIcon;

@FXML private ImageView close;

@FXML private ImageView logo; // logo for Guest House

@FXML private ImageView logo2; //receptionist logo

@FXML private TextField username;

@FXML private PasswordField password;

@FXML

public void close\_app(MouseEvent event)

{

System.exit(0);

}

@FXML private CheckBox rememberMe;

String empID = "";

//function for CHECK BOX

public void checkBox(String uName, String pass) throws FileNotFoundException

{

PrintWriter pw = new PrintWriter(new FileOutputStream("receptionist.txt"));

if(rememberMe.isSelected())

{

pw.println(uName + " " + pass);

System.out.println("WRITTEN \n");

}

else

{

pw.println();

System.out.println("\nNOT Written\n");

}

pw.close();

}

public void getCreditials() throws FileNotFoundException

{

Scanner reader = new Scanner(new FileInputStream("receptionist.txt"));

if(reader.hasNextLine())

{

String credentials[] = reader.nextLine().split(" ");

username.setText(credentials[0]);

password.setText(credentials[1]);

rememberMe.setSelected(true);

}

reader.close();

}

public void loginButtonPushed(ActionEvent event) throws IOException

{

if(!(username.getText().equals("") || password.getText().equals("")) && loginAuthentication())

{

String tempUsername = username.getText();

String tempPassword = password.getText();

checkBox(tempUsername, tempPassword);

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ReceptionistProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

{

if(username.getText().equals("") || password.getText().equals(""))

label.setText("You Cannot Leave Any of the Field Empty!");

if(!loginAuthentication())

label.setText("You Are NOT Registered in Our DATABASE !");

}

//REST OF THE WORk

}

public boolean loginAuthentication() //this method return true when login successfull else return false

{

boolean check = false;

String usrnm = username.getText().substring(5, username.getText().length());

empID = usrnm;

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT empID, password from receptionist where empID = '" + usrnm + "' and " + " password = '" + password.getText() + "';");

boolean temp = rs.next();

if(temp)

check = true;

//System.out.println("EMal : " + rs.getString(1) + "\t\t" + "Password : " + rs.getString(2));

// if(temp && rs.getString(1).equals(usrnm) && rs.getString(2).equals(password.getText()))

// {

// check = true;

// System.out.println("EMal : " + rs.getString(1) + "\t\t" + "Password : " + rs.getString(2));

// }

}

catch(Exception e)

{

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

}

return check;

}

//This Method is for moving to page/Screen for reseting password

public void forgetPasswordButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ForgetPassword.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ForgetPasswordController controller = loader.getController();

controller.setParameter("ReceptionistLogin.fxml");

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

Parent tableViewParent = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

msgIcon = new ImageView("id.png");

lockIcon = new ImageView("lock.png");

close = new ImageView("close.png");

logo = new ImageView("logo1.png");

logo2 = new ImageView("receptionistLogo.png");

try

{

getCreditials();

}

catch(Exception e)

{

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

}

}

}

## Receptionist Profile Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ReceptionistProfileController implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private ImageView welcome;

@FXML private ImageView house;

@FXML private ImageView room;

@FXML private ImageView customer;

@FXML private ImageView employee;

@FXML private ImageView complaint;

@FXML private ImageView account;

@FXML private Label nameLabel;

String empID = "";

String firstName = "";

String lastName = "";

public void setParameter(String s)

{

empID = s;

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select first\_name, last\_name from employee where empID = '" + empID + "';");

rs.next();

firstName = rs.getString(1);

lastName = rs.getString(2);

System.out.println(firstName + " " + lastName);

nameLabel.setText(firstName + " " + lastName);

}

catch(Exception e)

{

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

}

}

//============================================== IMAGE Methods =====================================================

@FXML

public void homeImagePressed(MouseEvent event) throws IOException // this function is call when home image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("HouseMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void roomImagePressed(MouseEvent event) throws IOException // this function is call when room image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RoomMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void customerImagePressed(MouseEvent event) throws IOException // this function is call when service image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Customer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void employeeImagePressed(MouseEvent event) throws IOException // this function is call when bill image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("EmpployeeMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

EmpployeeMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void complaintImagePressed(MouseEvent event) throws IOException, SQLException // this function is call when complaint image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ResponseComplaint1.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

System.out.println("EMP ID in Profile : " + empID);

ResponseComplaint1Controller controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

@FXML

public void accountImagePressed(MouseEvent event) throws IOException, SQLException // this function is call when account image is pressed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistAccount.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

System.out.println("Username in Profile : " + empID);

ReceptionistAccountController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//------------------------------------------ END Image Methods -----------------------------------------------------------

//============================================ BUTTON METHODS ============================================================

public void homeButtonPressed(ActionEvent event) throws IOException // this function is call when home Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("HouseMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

HouseMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void roomButtonPressed(ActionEvent event) throws IOException // this function is call when room Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RoomMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void customerButtonPressed(ActionEvent event) throws IOException // this function is call when service Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Customer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void employeeButtonPressed(ActionEvent event) throws IOException // this function is call when bill Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("EmpployeeMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

EmpployeeMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void complaintButtonPressed(ActionEvent event) throws IOException, SQLException // this function is call when complaint Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ResponseComplaint1.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ResponseComplaint1Controller controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void accountButtonPressed(ActionEvent event) throws IOException, SQLException // this function is call when acount Button is pushed

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistAccount.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ReceptionistAccountController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//--------------------------------------- END BUTTON METHODS ----------------------------------------------------------

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("receptionistLogo.png");

welcome = new ImageView("welcome3.png");

house = new ImageView("houseR.png");

room = new ImageView("roomR.png");

customer = new ImageView("customerR.png");

employee = new ImageView("employeeR.png");

complaint = new ImageView("complaint.png");

account = new ImageView("account.png");

}

}

## Register Customer Controller:

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.control.TextArea;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

public class RegisterCustomer2Controller implements Initializable

{

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private TextField country1;

@FXML private TextField country2;

@FXML private TextField city1;

@FXML private TextField city2;

@FXML private TextArea address1;

@FXML private TextArea address2;

@FXML private Label label;

String reg1[] = new String[7];

String reg2[] = new String[6];

public void setParameter(String[] reg1)

{

this.reg1 = reg1;

}

public void setParameter(String[] reg1, String[] reg2)

{

this.reg1 = reg1;

this.reg2 = reg2;

country1.setText(reg2[0]);

country2.setText(reg2[3]);

city1.setText(reg2[1]);

city2.setText(reg2[4]);

address1.setText(reg2[2]);

address2.setText(reg2[5]);

}

//This Method is for moving back to previous page/Screen (RegisterCustomer Screen) from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RegisterCustomer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

RegisterCustomerController controller = loader.getController();

controller.setParameter(reg1);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving to next page/Screen which is RegisterCustomer3

public void nextButtonPushedRegister2(ActionEvent event) throws IOException

{

if(!(country1.getText().equals("") || city1.getText().equals("") || address1.getText().equals("")))

{

if(Checks.alphaCheckIgnoreSpace(country1.getText()))

{

if(Checks.alphaCheckIgnoreSpace(country2.getText()) || country2.getText().equals(""))

{

if(Checks.alphaCheckIgnoreSpace(city1.getText()))

{

if(Checks.alphaCheckIgnoreSpace(city2.getText()) || city2.getText().equals(""))

{

reg2[0] = country1.getText();

reg2[1] = city1.getText();

reg2[2] = address1.getText();

reg2[3] = country2.getText();

reg2[4] = city2.getText();

reg2[5] = address2.getText();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RegisterCustomer3.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RegisterCustomer3Controller controller = loader.getController();

controller.setParameter(reg1, reg2);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("City 2 Name only contains alphabets");

}

else

label.setText("City 1 Name only contains alphabets");

}

else

label.setText("Country 2 Name only contains alphabets");

}

else

label.setText("Country 1 Name only contains alphabets");

}

else

{

label.setText("You Cannot Leave Any of the Field Empty! (Except Other mobile number)");

}

}

@Override

public void initialize(URL url, ResourceBundle rb)

{

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerLogo.png");

}

}

## Register Customer 3 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Label;

import javafx.scene.control.PasswordField;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class RegisterCustomer3Controller implements Initializable

{

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private TextField username;

@FXML private PasswordField password1;

@FXML private PasswordField password2;

@FXML private Label label;

String reg1[] = new String[7];

String reg2[] = new String[6];

public void setParameter(String[] reg1, String[] reg2)

{

this.reg1 = reg1;

this.reg2 = reg2;

}

//This Method is for moving back to previous page/Screen (RegisterCustomer2 Screen) from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RegisterCustomer2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

RegisterCustomer2Controller controller = loader.getController();

controller.setParameter(reg1, reg2);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void getRegisterButtonPressed(ActionEvent event) throws IOException

{

if(!(username.getText().equals("") || password1.getText().equals("") || password2.getText().equals("")) && (password1.getText().equals(password2.getText())))

{

try

{

if(Checks.emailCheck(username.getText()))

{

if(Checks.emailCheckData(username.getText()))

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

//

// int up = stmt.executeUpdate();

//

String sql = "INSERT INTO `mydb`.`customer` (`first\_name`, `last\_name`, `gender`, `DOB`, `CNIC`, `email`, `password`, `joining\_date`) VALUES ('"+reg1[0]+"', '"+reg1[1]+"', '"+reg1[4]+"', '"+reg1[3]+"', '"+reg1[2]+"', '"+username.getText()+"', '"+password1.getText()+"', '"+Checks.getCurrentdate()+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

ResultSet rs = stmt.executeQuery("select customerID from customer where email = '" + username.getText() + "';");

rs.next();

String customerID = rs.getString(1);

sql = "INSERT INTO `mydb`.`customer\_phone` (`customerID`, `phone\_number`) VALUES ('"+customerID+"', '"+reg1[5]+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

if(!reg1[6].equals(""))

{

sql = "INSERT INTO `mydb`.`customer\_phone` (`customerID`, `phone\_number`) VALUES ('"+customerID+"', '"+reg1[6]+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

}

String countryID = "";

rs = stmt.executeQuery("select countryID from country where country\_name = '" + reg2[0] + "';");

if(rs.next())

countryID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+reg2[0]+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select countryID from country where country\_name = '" + reg2[0] + "';");

rs.next();

countryID = rs.getString(1);

}

String cityID = "";

rs = stmt.executeQuery("select cityID from city where city\_name = '" + reg2[1] + "' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+reg2[1]+"', '"+countryID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select cityID from city where city\_name = '" + reg2[1] + "' and countryID = '"+countryID+"';");

rs.next();

cityID = rs.getString(1);

}

String addressID = "";

rs = stmt.executeQuery("select addressID from address where City\_cityID = '" + cityID + "' and address = '"+reg2[2]+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+reg2[2]+"', '"+cityID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select addressID from address where address = '" + reg2[2] + "' and City\_cityID = '"+cityID+"';");

rs.next();

addressID = rs.getString(1);

}

sql = "INSERT INTO `mydb`.`customer\_address` (`addressID`, `customerID`) VALUES ('"+addressID+"', '"+ customerID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

if(!reg2[3].equals("") && !reg2[4].equals("") && !reg2[5].equals(""))

{

countryID = "";

rs = stmt.executeQuery("select countryID from country where country\_name = '" + reg2[3] + "';");

if(rs.next())

countryID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+reg2[3]+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select countryID from country where country\_name = '" + reg2[3] + "';");

rs.next();

countryID = rs.getString(1);

}

cityID = "";

rs = stmt.executeQuery("select cityID from city where city\_name = '" + reg2[4] + "' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+reg2[4]+"', '"+countryID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select cityID from city where city\_name = '" + reg2[4] + "' and countryID = '"+countryID+"';");

rs.next();

cityID = rs.getString(1);

}

addressID = "";

rs = stmt.executeQuery("select addressID from address where City\_cityID = '" + cityID + "' and address = '"+reg2[5]+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+reg2[5]+"', '"+cityID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select addressID from address where address = '" + reg2[5] + "' and City\_cityID = '"+cityID+"';");

rs.next();

addressID = rs.getString(1);

}

sql = "INSERT INTO `mydb`.`customer\_address` (`addressID`, `customerID`) VALUES ('"+addressID+"', '"+ customerID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

}

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("Successfully Register ");

a1.setContentText("Your Data has been registered successfully, Now you can login through your account !");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerLogin.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("This Username Already have registered, Please Entered Another one !");

}

else

label.setText("Please Enter Correct Email ! (like abcxyz@gmail.com)");

}

catch(SQLException e)

{

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

}

}

else

{

if(username.getText().equals("") || password1.getText().equals("") || password2.getText().equals(""))

label.setText("You Cannot Leave Any of the Field Empty!");

else if(!(password1.getText().equals(password2.getText())))

label.setText("Re-Entered Password Did'nt Match!");

}

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerLogo.png");

}

}

## Register Customer Controller:

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.RadioButton;

import javafx.scene.control.TextField;

import javafx.scene.control.ToggleGroup;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

public class RegisterCustomerController implements Initializable {

String radioButtonValue = "";

@FXML private ImageView logo;

@FXML private ImageView logo2;

//For Radio Button

@FXML private RadioButton male;

@FXML private RadioButton female;

@FXML private ToggleGroup favLang; //this is for grouping the upper 2 radio button, so that only one button can be selected at a time

//for getting Date

@FXML private DatePicker dateOfBirth;

//TextFields

@FXML private TextField firstName;

@FXML private TextField lastName;

@FXML private TextField cnic;

@FXML private TextField mob1; //mobile number 1

@FXML private TextField mob2;

@FXML private Label label; //label for wrong input

String reg1[] = new String[7];

public void setParameter(String[] reg1)

{

this.reg1 = reg1;

firstName.setText(reg1[0]);

lastName.setText(reg1[1]);

cnic.setText(reg1[2]);

dateOfBirth.setValue(Checks.stringToLocalDateConverter(reg1[3]));

if(reg1[4].equalsIgnoreCase("Male"))

favLang.selectToggle(male);

else if(reg1[4].equalsIgnoreCase("Female"))

favLang.selectToggle(female);

mob1.setText(reg1[5]);

mob2.setText(reg1[6]);

}

//For getting RadioButton's Value

public void radioButtonView()

{

if(favLang.getSelectedToggle().equals(male))

this.radioButtonValue = male.getText();

else if(favLang.getSelectedToggle().equals(female))

this.radioButtonValue = female.getText();

}

//This Method is for moving back to previous page/Screen (Customer Screen) from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

Parent tableViewParent = FXMLLoader.load(getClass().getResource("CustomerLogin.fxml"));

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving to next page/Screen which is RegisterCustomer2

public void nextButtonPushedRegister(ActionEvent event) throws IOException

{

radioButtonView(); //call just to initalize the value of RadioButtons

if(Checks.alphaCheckIgnoreSpace(firstName.getText()) && !firstName.getText().equals("") && Checks.alphaCheckIgnoreSpace(lastName.getText()) && !lastName.getText().equals("")&& Checks.cnicCheck(cnic.getText()) && !cnic.getText().equals("")&& Checks.mobCheck(mob1.getText()) && !(dateOfBirth.getValue().equals("")) && !radioButtonValue.equals("") && !mob1.getText().equals("") && (Checks.mobCheck(mob2.getText()) || mob2.getText().equals("")))

{

if(Checks.cnicCheckData(cnic.getText()))

{

if(Checks.mobCheckData(mob1.getText()))

{

if(Checks.mobCheckData(mob2.getText()) || mob2.getText().equals(""))

{

String DOB[] = dateOfBirth.getValue().toString().split("-");

if(Checks.ageCheck(Integer.parseInt(DOB[0]), Integer.parseInt(DOB[1]), Integer.parseInt(DOB[2])))

{

reg1[0] = firstName.getText();

reg1[1] = lastName.getText();

reg1[2] = cnic.getText();

reg1[3] = dateOfBirth.getValue().toString();

reg1[4] = radioButtonValue;

reg1[5] = mob1.getText();

reg1[6] = mob2.getText();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RegisterCustomer2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

RegisterCustomer2Controller controller = loader.getController();

controller.setParameter(reg1);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("You Cannot Register ! Your Age is Under 18!!!");

}

else

label.setText("This Mobile 2 is already registered");

}

else

label.setText("This Mobile 1 is already registered");

}

else

label.setText("This CNIC Already Registered !!!");

}

else

{

if(firstName.getText().equals("") || lastName.getText().equals("") || cnic.getText().equals("") || mob1.getText().equals("") || (dateOfBirth.getValue().equals("")) || radioButtonValue == null)

{

label.setText("You Cannot Leave Any of the Field Empty! (Except Other mobile number)");

// System.out.println(firstName.getText());

// System.out.println(lastName.getText());

// System.out.println(cnic.getText());

// System.out.println(mob1.getText());

// System.out.println(dateOfBirth.getValue());

// System.out.println(radioButtonValue);

}

if(!Checks.mobCheck(mob2.getText()))

label.setText("Please! Enter Mobile Number like 03035405615 (Only 11 digits)");

if(!Checks.alphaCheckIgnoreSpace(firstName.getText()))

label.setText("Please! Enter Only Alpha Characters in First Name Field");

if(!Checks.alphaCheckIgnoreSpace(lastName.getText()))

label.setText("Please! Enter Only Alpha Characters in Last Name Field");

if(!Checks.cnicCheck(cnic.getText()))

label.setText("Please! Enter CNIC in Format 12345-1234567-1 (Use only digits and '-')");

if(!Checks.mobCheck(mob1.getText()))

label.setText("Please! Enter Mobile Number like 03035405615 (Only 11 digits)");

}

}

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerLogo.png");

//for RADIO Button and there toggle group

favLang = new ToggleGroup();

male.setToggleGroup(favLang);

female.setToggleGroup(favLang);

favLang.selectToggle(male);

}

}

## Register Employee 2 Controller

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.control.TextArea;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class RegisterEmployee2Controller implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private TextField country1;

@FXML private TextField country2;

@FXML private TextField city1;

@FXML private TextField city2;

@FXML private TextArea address1;

@FXML private TextArea address2;

@FXML private Label label;

String reg1[] = new String[8];

String reg2[] = new String[6];

String empID = "";

public void setParameter(String[] reg1)

{

this.reg1 = reg1;

}

public void setParameter(String[] reg1, String[] reg2, String empID)

{

this.empID = empID;

this.reg1 = reg1;

this.reg2 = reg2;

country1.setText(reg2[0]);

country2.setText(reg2[3]);

city1.setText(reg2[1]);

city2.setText(reg2[4]);

address1.setText(reg2[2]);

address2.setText(reg2[5]);

}

//This Method is for moving back to previous page/Screen (RegisterCustomer Screen) from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RegisterEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

RegisterEmployeeController controller = loader.getController();

controller.setParameter(reg1, empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving to next page/Screen which is RegisterCustomer3

public void nextButtonPushedRegister2(ActionEvent event) throws IOException

{

if(!(country1.getText().equals("") || city1.getText().equals("") || address1.getText().equals("")))

{

if(Checks.alphaCheckIgnoreSpace(country1.getText()))

{

if(Checks.alphaCheckIgnoreSpace(country2.getText()) || country2.getText().equals(""))

{

if(Checks.alphaCheckIgnoreSpace(city1.getText()))

{

if(Checks.alphaCheckIgnoreSpace(city2.getText()) || city2.getText().equals(""))

{

reg2[0] = country1.getText();

reg2[1] = city1.getText();

reg2[2] = address1.getText();

reg2[3] = country2.getText();

reg2[4] = city2.getText();

reg2[5] = address2.getText();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RegisterEmployee3.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RegisterEmployee3Controller controller = loader.getController();

controller.setParameter(reg1, reg2, empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("City 2 Name only contains alphabets");

}

else

label.setText("City 1 Name only contains alphabets");

}

else

label.setText("Country 2 Name only contains alphabets");

}

else

label.setText("Country 1 Name only contains alphabets");

}

else

{

label.setText("You Cannot Leave Any of the Field Empty! (Except Other mobile number)");

}

}

@Override

public void initialize(URL url, ResourceBundle rb)

{

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerLogo.png");

}

}

## Register Employee 3 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.PasswordField;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class RegisterEmployee3Controller implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private ComboBox branch;

@FXML private ComboBox type;

@FXML private PasswordField password1;

@FXML private PasswordField password2;

@FXML private Label label;

@FXML private Label label2;

@FXML private Label labelP;

@FXML private Label label2P;

String reg1[] = new String[7];

String reg2[] = new String[6];

String empID = "";

public void setParameter(String[] reg1, String[] reg2, String empID)

{

this.empID = empID;

this.reg1 = reg1;

this.reg2 = reg2;

}

//This Method is for moving back to previous page/Screen (RegisterCustomer2 Screen) from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RegisterEmployee2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

RegisterEmployee2Controller controller = loader.getController();

controller.setParameter(reg1, reg2, empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void typeComboSelect()

{

if(type.getValue().toString().equalsIgnoreCase("Receptionist"))

{

password1.setVisible(true);

password2.setVisible(true);

label2.setVisible(true);

labelP.setVisible(true);

label2P.setVisible(true);

}

else

{

password1.setVisible(false);

password2.setVisible(false);

label2.setVisible(false);

labelP.setVisible(false);

label2P.setVisible(false);

}

}

public void getRegisterButtonPressed(ActionEvent event) throws IOException, SQLException

{

String employeeID = "";

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

//

// int up = stmt.executeUpdate();

//

if(password1.isVisible() && password2.isVisible())

{

if(password1.getText().equals(password2.getText()))

{

dataAddition();

ResultSet rs = stmt.executeQuery("select empID from employee where CNIC = '" + reg1[2] + "';");

rs.next();

employeeID = rs.getString(1);

String sql = "INSERT INTO `mydb`.`receptionist` (`empID`, `password`) VALUES ('"+employeeID+"', '"+password1.getText()+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

}

else

label.setText("Password Didn't match ! ");

}

else

{

dataAddition();

}

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("Successfully Register ");

a1.setContentText("New Employee has been registered successfully !");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

AdminEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void dataAddition() throws SQLException

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branchID from branch where branch\_name = '"+branch.getValue().toString()+"';");

rs.next();

String branchID = rs.getString(1);

String sql = "INSERT INTO `mydb`.`employee` (`first\_name`, `last\_name`, `gender`, `DOB`, `CNIC`, `joinDate`, `salary`, `branchID`) VALUES ('"+reg1[0]+"', '"+reg1[1]+"', '"+reg1[4]+"', '"+reg1[3]+"', '"+reg1[2]+"', '"+Checks.getCurrentdate()+"', '"+reg1[5]+"', '"+branchID+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql); //This Will Prevent From SQL Injection

int rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select empID from employee where CNIC = '" + reg1[2] + "';");

rs.next();

String customerID = rs.getString(1);

sql = "INSERT INTO `mydb`.`employee\_phone` (`empID`, `phone\_number`) VALUES ('"+customerID+"', '"+reg1[6]+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

if(!reg1[7].equals(""))

{

sql = "INSERT INTO `mydb`.`employee\_phone` (`empID`, `phone\_number`) VALUES ('"+customerID+"', '"+reg1[7]+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

}

String countryID = "";

rs = stmt.executeQuery("select countryID from country where country\_name = '" + reg2[0] + "';");

if(rs.next())

countryID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+reg2[0]+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select countryID from country where country\_name = '" + reg2[0] + "';");

rs.next();

countryID = rs.getString(1);

}

String cityID = "";

rs = stmt.executeQuery("select cityID from city where city\_name = '" + reg2[1] + "' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+reg2[1]+"', '"+countryID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select cityID from city where city\_name = '" + reg2[1] + "' and countryID = '"+countryID+"';");

rs.next();

cityID = rs.getString(1);

}

String addressID = "";

rs = stmt.executeQuery("select addressID from address where City\_cityID = '" + cityID + "' and address = '"+reg2[2]+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+reg2[2]+"', '"+cityID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select addressID from address where address = '" + reg2[2] + "' and City\_cityID = '"+cityID+"';");

rs.next();

addressID = rs.getString(1);

}

sql = "INSERT INTO `mydb`.`address\_has\_employee` (`addressID`, `empID`) VALUES ('"+addressID+"', '"+ customerID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

if(!reg2[3].equals("") && !reg2[4].equals("") && !reg2[5].equals(""))

{

countryID = "";

rs = stmt.executeQuery("select countryID from country where country\_name = '" + reg2[3] + "';");

if(rs.next())

countryID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`country` (`country\_name`) VALUES ('"+reg2[3]+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select countryID from country where country\_name = '" + reg2[3] + "';");

rs.next();

countryID = rs.getString(1);

}

cityID = "";

rs = stmt.executeQuery("select cityID from city where city\_name = '" + reg2[4] + "' and countryID = '"+countryID+"';");

if(rs.next())

cityID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`city` (`city\_name`, `countryID`) VALUES ('"+reg2[4]+"', '"+countryID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select cityID from city where city\_name = '" + reg2[4] + "' and countryID = '"+countryID+"';");

rs.next();

cityID = rs.getString(1);

}

addressID = "";

rs = stmt.executeQuery("select addressID from address where City\_cityID = '" + cityID + "' and address = '"+reg2[5]+"';");

if(rs.next())

addressID = rs.getString(1);

else

{

sql = "INSERT INTO `mydb`.`address` (`address`, `City\_cityID`) VALUES ('"+reg2[5]+"', '"+cityID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

rs = stmt.executeQuery("select addressID from address where address = '" + reg2[5] + "' and City\_cityID = '"+cityID+"';");

rs.next();

addressID = rs.getString(1);

}

sql = "INSERT INTO `mydb`.`address\_has\_employee` (`addressID`, `empID`) VALUES ('"+addressID+"', '"+ customerID+"');";

preparedStatement = conn.prepareStatement(sql);

rowsAffected = preparedStatement.executeUpdate();

}

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

// TODO

password1.setVisible(false);

password2.setVisible(false);

label2.setVisible(false);

labelP.setVisible(false);

label2P.setVisible(false);

branch.getItems().addAll("ISB Branch 1", "ISB Branch 2", "LHR Branch", "KHI Branch", "PEW Branch 1", "PEW Branch 2");

type.getItems().addAll("housekeeping", "repairer", "sweeper", "cook", "laundryman", "Receptionist");

branch.setValue("ISB Branch 1");

type.setValue("housekeeping");

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerLogo.png");

}

}

## Register Employee Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.RadioButton;

import javafx.scene.control.TextField;

import javafx.scene.control.ToggleGroup;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class RegisterEmployeeController implements Initializable {

String radioButtonValue = "";

@FXML private ImageView logo;

@FXML private ImageView logo2;

//For Radio Button

@FXML private RadioButton male;

@FXML private RadioButton female;

@FXML private ToggleGroup favLang; //this is for grouping the upper 2 radio button, so that only one button can be selected at a time

//for getting Date

@FXML private DatePicker dateOfBirth;

//TextFields

@FXML private TextField firstName;

@FXML private TextField lastName;

@FXML private TextField cnic;

@FXML private TextField salary;

@FXML private TextField mob1; //mobile number 1

@FXML private TextField mob2;

@FXML private Label label; //label for wrong input

String reg1[] = new String[8];

String empID = "";

public void setParameter(String[] reg1, String empID)

{

this.empID = empID;

this.reg1 = reg1;

firstName.setText(reg1[0]);

lastName.setText(reg1[1]);

cnic.setText(reg1[2]);

dateOfBirth.setValue(Checks.stringToLocalDateConverter(reg1[3]));

if(reg1[4].equalsIgnoreCase("Male"))

favLang.selectToggle(male);

else if(reg1[4].equalsIgnoreCase("Female"))

favLang.selectToggle(female);

salary.setText(reg1[5]);

mob1.setText(reg1[6]);

mob2.setText(reg1[7]);

}

public void setParameter(String empID)

{

this.empID = empID;

}

//For getting RadioButton's Value

public void radioButtonView()

{

if(favLang.getSelectedToggle().equals(male))

this.radioButtonValue = male.getText();

else if(favLang.getSelectedToggle().equals(female))

this.radioButtonValue = female.getText();

}

//This Method is for moving back to previous page/Screen (Customer Screen) from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AdminEmployee.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

AdminEmployeeController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving to next page/Screen which is RegisterCustomer2

public void nextButtonPushedRegister(ActionEvent event) throws IOException

{

radioButtonView(); //call just to initalize the value of RadioButtons

if(Checks.alphaCheckIgnoreSpace(firstName.getText()) && !firstName.getText().equals("") && Checks.alphaCheckIgnoreSpace(lastName.getText()) && !lastName.getText().equals("")&& Checks.cnicCheck(cnic.getText()) && !cnic.getText().equals("")&& Checks.mobCheck(mob1.getText()) && !(dateOfBirth.getValue().equals("")) && !radioButtonValue.equals("") && !mob1.getText().equals("") && (Checks.mobCheck(mob2.getText()) || mob2.getText().equals("")))

{

if(Checks.cnicCheckData(cnic.getText()))

{

if(Checks.mobCheckData(mob1.getText()))

{

if(Checks.mobCheckData(mob2.getText()) || mob2.getText().equals(""))

{

String DOB[] = dateOfBirth.getValue().toString().split("-");

if(Checks.ageCheck(Integer.parseInt(DOB[0]), Integer.parseInt(DOB[1]), Integer.parseInt(DOB[2])))

{

if(Checks.digitCheck((salary.getText())))

{

if(Integer.parseInt(salary.getText()) >= 20000)

{

reg1[0] = firstName.getText();

reg1[1] = lastName.getText();

reg1[2] = cnic.getText();

reg1[3] = dateOfBirth.getValue().toString();

reg1[4] = radioButtonValue;

reg1[5] = salary.getText();

reg1[6] = mob1.getText();

reg1[7] = mob2.getText();

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RegisterEmployee2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

//

RegisterEmployee2Controller controller = loader.getController();

controller.setParameter(reg1);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

else

label.setText("Salary can NOT be less than 20000!!!");

}

else

label.setText("Salary can only be a Digit Entity!!!");

}

else

label.setText("You Cannot Register ! Your Age is Under 18!!!");

}

else

label.setText("This Mobile 2 is already registered");

}

else

label.setText("This Mobile 1 is already registered");

}

else

label.setText("This CNIC Already Registered !!!");

}

else

{

if(firstName.getText().equals("") || lastName.getText().equals("") || cnic.getText().equals("") || mob1.getText().equals("") || (dateOfBirth.getValue().equals("")) || radioButtonValue == null)

{

label.setText("You Cannot Leave Any of the Field Empty! (Except Other mobile number)");

// System.out.println(firstName.getText());

// System.out.println(lastName.getText());

// System.out.println(cnic.getText());

// System.out.println(mob1.getText());

// System.out.println(dateOfBirth.getValue());

// System.out.println(radioButtonValue);

}

if(!Checks.mobCheck(mob2.getText()))

label.setText("Please! Enter Mobile Number like 03035405615 (Only 11 digits)");

if(!Checks.alphaCheckIgnoreSpace(firstName.getText()))

label.setText("Please! Enter Only Alpha Characters in First Name Field");

if(!Checks.alphaCheckIgnoreSpace(lastName.getText()))

label.setText("Please! Enter Only Alpha Characters in Last Name Field");

if(!Checks.cnicCheck(cnic.getText()))

label.setText("Please! Enter CNIC in Format 12345-1234567-1 (Use only digits and '-')");

if(!Checks.mobCheck(mob1.getText()))

label.setText("Please! Enter Mobile Number like 03035405615 (Only 11 digits)");

}

}

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("customerLogo.png");

//for RADIO Button and there toggle group

favLang = new ToggleGroup();

male.setToggleGroup(favLang);

female.setToggleGroup(favLang);

favLang.selectToggle(male);

mob1.setEditable(true);

}

}

## Response Complaint 1 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ResponseComplaint1Controller implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<Customer> tableView;

@FXML private TableColumn<Customer, String> srNoC;

@FXML private TableColumn<Customer, String> complaintC;

@FXML private TableColumn<Customer, String> dateC;

@FXML private ComboBox combo;

@FXML private Button DetailView;

@FXML private Label label;

@FXML private Label label2;

int total\_customer = 0;

String empID = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from complaint;");

while(rs.next())

total\_customer++;

label.setText("TOTAL COMPLAINTS : " + total\_customer);

setTableData();

}

public void userClickOnTable()

{

this.DetailView.setDisable(false);

}

public void updatedComboBox() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(combo.getValue().toString().equals("FOOD"))

{

ResultSet rs = stmt.executeQuery("select complaint.complaintID, complaint.complaint, complaint.complaint\_date from complaint, complaint\_category where complaint\_category.complaintCategoryID = complaint.complaintCategoryID and complaint\_category.category = 'food' and complaint.complaintID not in (select complaintID from response);");

while(rs.next())

{

list.add(new Customer(rs.getString(1), rs.getString(2), rs.getString(3)));

}

}

else if(combo.getValue().toString().equals("LAUNDRY"))

{

ResultSet rs = stmt.executeQuery("select complaint.complaintID, complaint.complaint, complaint.complaint\_date from complaint, complaint\_category where complaint\_category.complaintCategoryID = complaint.complaintCategoryID and complaint\_category.category = 'laundry' and complaint.complaintID not in (select complaintID from response);");

while(rs.next())

{

list.add(new Customer(rs.getString(1), rs.getString(2), rs.getString(3)));

}

}

else if(combo.getValue().toString().equals("CLEANING"))

{

ResultSet rs = stmt.executeQuery("select complaint.complaintID, complaint.complaint, complaint.complaint\_date from complaint, complaint\_category where complaint\_category.complaintCategoryID = complaint.complaintCategoryID and complaint\_category.category = 'cleaning' and complaint.complaintID not in (select complaintID from response);");

while(rs.next())

{

list.add(new Customer(rs.getString(1), rs.getString(2), rs.getString(3)));

}

}

else if(combo.getValue().toString().equals("MISMANAGEMENT"))

{

ResultSet rs = stmt.executeQuery("select complaint.complaintID, complaint.complaint, complaint.complaint\_date from complaint, complaint\_category where complaint\_category.complaintCategoryID = complaint.complaintCategoryID and complaint\_category.category = 'mismanagement' and complaint.complaintID not in (select complaintID from response);");

while(rs.next())

{

list.add(new Customer(rs.getString(1), rs.getString(2), rs.getString(3)));

}

}

else if(combo.getValue().toString().equals("HOUSE CONDITION"))

{

ResultSet rs = stmt.executeQuery("select complaint.complaintID, complaint.complaint, complaint.complaint\_date from complaint, complaint\_category where complaint\_category.complaintCategoryID = complaint.complaintCategoryID and complaint\_category.category = 'house condition' and complaint.complaintID not in (select complaintID from response);");

while(rs.next())

{

list.add(new Customer(rs.getString(1), rs.getString(2), rs.getString(3)));

}

}

else if(combo.getValue().toString().equals("OTHERS"))

{

ResultSet rs = stmt.executeQuery("select complaint.complaintID, complaint.complaint, complaint.complaint\_date from complaint, complaint\_category where complaint\_category.complaintCategoryID = complaint.complaintCategoryID and complaint\_category.category = 'other' and complaint.complaintID not in (select complaintID from response);");

while(rs.next())

{

list.add(new Customer(rs.getString(1), rs.getString(2), rs.getString(3)));

}

}

else

{

ResultSet rs = stmt.executeQuery("select complaintID, complaint, complaint\_date from complaint where complaint.complaintID not in (select complaintID from response);");

while(rs.next())

{

list.add(new Customer(rs.getString(1), rs.getString(2), rs.getString(3)));

}

}

tableView.setItems(list);

}

public void setTableData() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select complaintID, complaint, complaint\_date from complaint where complaint.complaintID not in (select complaintID from response);");

while(rs.next())

{

list.add(new Customer(rs.getString(1), rs.getString(2), rs.getString(3)));

}

tableView.setItems(list);

}

public void giveResponseButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ResponseComplaint2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ResponseComplaint2Controller controller = loader.getController();

controller.setParameter(empID, tableView.getSelectionModel().getSelectedItem());

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ReceptionistProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

DetailView.setDisable(true);

combo.getItems().addAll("FOOD", "LAUNDRY", "CLEANING","MISMANAGEMENT", "HOUSE CONDITION", "OTHERS", "ALL");

combo.setValue("ALL");

srNoC.setCellValueFactory(new PropertyValueFactory<Customer, String>("srNo"));

complaintC.setCellValueFactory(new PropertyValueFactory<Customer, String>("customerName"));

dateC.setCellValueFactory(new PropertyValueFactory<Customer, String>("cnic"));

}

}

## Response Complaint 2 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.Label;

import javafx.scene.control.TextArea;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ResponseComplaint2Controller implements Initializable {

@FXML private ImageView logo;

@FXML private TextArea complaintText;

@FXML private TextArea responseText;

@FXML private Label label;

String empID = "";

String complaintID = "";

String complaint = "";

String complaintDate = "";

public void setParameter(String empID, Customer c)

{

this.empID = empID;

this.complaint = c.getCustomerName();

this.complaintID = c.getSrNo();

this.complaintDate = c.getCnic();

for(int i = 0; i < complaint.length(); i++)

{

if(i % 30 == 0)

{

complaint = complaint.substring(0, i) + "\n" + complaint.substring(i, complaint.length());

}

}

complaintText.setText(complaint);

}

public void submitResponse(ActionEvent event) throws IOException, SQLException

{

if(!responseText.getText().equals(""))

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String sql = "INSERT INTO `mydb`.`response` (`response`, `complaintID`, `responseDate`) VALUES ('"+responseText.getText()+"', '"+complaintID+"', '"+Checks.getCurrentdate()+"');";

PreparedStatement preparedStatement = conn.prepareStatement(sql);

int rowsAffected = preparedStatement.executeUpdate();

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("Response Submission !");

a1.setContentText("Your Response Has Been Submitted !");

a1.setHeaderText(null); //you can also set header

a1.showAndWait();

backButtonPushed(event);

}

else

label.setText("You Cannot Submit Empty Response !");

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ResponseComplaint1.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ResponseComplaint1Controller controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

complaintText.setEditable(false);

}

}

## Room Bookees Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.ComboBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class RoomBookeesController implements Initializable

{

@FXML private ImageView logo;

@FXML private TableView<HouseModalTable> tableView;

@FXML private TableColumn<HouseModalTable, String> houseName;

@FXML private TableColumn<HouseModalTable, String> branchNameC; //this will consider as the House Name

@FXML private TableColumn<HouseModalTable, String> locationC;

@FXML private TableColumn<HouseModalTable, String> cityC;

@FXML private TableColumn<HouseModalTable, String> categoryC;

@FXML private TableColumn<HouseModalTable, String> statusC;

@FXML private ComboBox branch;

@FXML private ComboBox category;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void showResultButtonPushed(ActionEvent event) throws IOException, SQLException

{

ObservableList<HouseModalTable> list = FXCollections.observableArrayList();

System.out.println("I am in the SEt Tabel DAta Method");

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(category.getValue().toString().equals("ALL"))

{

String tempRoom = "";

String tempHouse = "";

String tempBranch = "";

String tempLocation = "";

String tempCity = "";

String tempType = "";

String tempRoomID = "";

String room[];

String roomID[];

String house[];

String branch\_name[];

String location[];

String city[];

String houseType[];

ResultSet rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type, room.roomID from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type like 'roomsOnly%';");

while(rs.next())

{

tempRoom = tempRoom + rs.getString(1) + "#";

tempHouse = tempHouse + rs.getString(2) + "#";

tempBranch = tempBranch + rs.getString(3) + "#";

tempLocation = tempLocation + rs.getString(4) + "#";

tempCity = tempCity + rs.getString(5) + "#";

tempType = tempType + rs.getString(6) + "#";

tempRoomID = tempRoomID + rs.getString(7) + "#";

}

room = tempRoom.split("#");

roomID = tempRoomID.split("#");

house = tempHouse.split("#");

branch\_name = tempBranch.split("#");

location = tempLocation.split("#");

city = tempCity.split("#");

houseType = tempType.split("#");

for(int i = 0; i < roomID.length; i++)

{

rs = stmt.executeQuery("SELECT booking, count(\*) as Total\_Bookees from customer\_record where booking = 'r"+roomID[i]+"' group by booking;");

if(rs.next())

{

list.add(new HouseModalTable(room[i], house[i], branch\_name[i], location[i] + ", " + city[i], houseType[i], rs.getString(2)));

}

else

{

list.add(new HouseModalTable(room[i], house[i], branch\_name[i], location[i] + ", " + city[i], houseType[i], "0"));

}

}

}

else

{

String tempRoom = "";

String tempHouse = "";

String tempBranch = "";

String tempLocation = "";

String tempCity = "";

String tempType = "";

String tempRoomID = "";

String room[];

String roomID[];

String house[];

String branch\_name[];

String location[];

String city[];

String houseType[];

ResultSet rs = stmt.executeQuery("select distinct room.room\_number, house.houseID, branch\_name, location, city\_name, house.huose\_Type, room.roomID from house, branch, city, location, room where room.houseID = house.houseID and branch.branch\_name = '"+branch.getValue().toString()+"' and house.branchID = branch.branchID and branch.locationID = location.locationID and location.cityID = city.cityID and house.huose\_Type = 'roomsOnly"+category.getValue().toString()+"';");

while(rs.next())

{

tempRoom = tempRoom + rs.getString(1) + "#";

tempHouse = tempHouse + rs.getString(2) + "#";

tempBranch = tempBranch + rs.getString(3) + "#";

tempLocation = tempLocation + rs.getString(4) + "#";

tempCity = tempCity + rs.getString(5) + "#";

tempType = tempType + rs.getString(6) + "#";

tempRoomID = tempRoomID + rs.getString(7) + "#";

}

room = tempRoom.split("#");

roomID = tempRoomID.split("#");

house = tempHouse.split("#");

branch\_name = tempBranch.split("#");

location = tempLocation.split("#");

city = tempCity.split("#");

houseType = tempType.split("#");

for(int i = 0; i < roomID.length; i++)

{

rs = stmt.executeQuery("SELECT booking, count(\*) as Total\_Bookees from customer\_record where booking = 'r"+roomID[i]+"' group by booking;");

if(rs.next())

{

list.add(new HouseModalTable(room[i], house[i], branch\_name[i], location[i] + ", " + city[i], houseType[i], rs.getString(2)));

}

else

{

list.add(new HouseModalTable(room[i], house[i], branch\_name[i], location[i] + ", " + city[i], houseType[i], "0"));

}

}

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RoomMain.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomMainController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb){

// TODO

branch.getItems().addAll("ISB Branch 1", "ISB Branch 2", "PEW Branch 1", "PEW Branch 2", "LHR Branch", "KHI Branch");

category.getItems().addAll("ALL", "LUXURY", "ECONOMY");

String branchID = "";

String branchName = "";

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select branchID from employee where empID = '"+ empID +"';");

while(rs.next())

{

branchID = rs.getString(1);

}

rs = stmt.executeQuery("select branch\_name from branch where branchID = '" + branchID + "';");

while(rs.next())

{

branchName = rs.getString(1);

}

}

catch(SQLException e)

{

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

}

branch.setValue(branchName);

category.setValue("ALL");

houseName.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("srNo"));

branchNameC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("branchName"));

locationC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("location"));

cityC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("city"));

categoryC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("category"));

statusC.setCellValueFactory(new PropertyValueFactory<HouseModalTable, String>("status"));

logo = new ImageView("logo1.png");

}

}

## Room Controller:

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

public class RoomController implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private ImageView viewRoom;

@FXML private ImageView bookRoom;

String usernameOfLoggedInPerson;

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

}

public void availableRoomButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableRoom.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableRoomController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void availableRoomImagePushed(MouseEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableRoom.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableRoomController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void bookRoomButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookRoom.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BookRoomController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void bookRoomImagePushed(MouseEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookRoom.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BookRoomController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerProfileController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("room.png");

viewRoom = new ImageView("availableRooms.png");

bookRoom = new ImageView("bookARoom.png");

}

}

## Room Main Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class RoomMainController implements Initializable {

@FXML private ImageView logo;

String empID = "";

public void setParameter(String empID)

{

this.empID = empID;

}

public void allRoomButtonPressed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AllRoom.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AllRoomController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void bookedRoomButtonPressed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("BookedRooms.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

BookedRoomsController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void availableRoomButtonPressed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("AvailableHousesFromMainHouses.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

AvailableHousesFromMainHousesController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void allRoomBookesButtonPressed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("RoomBookees.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

RoomBookeesController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ReceptionistProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ReceptionistProfileController controller = loader.getController();

controller.setParameter(empID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

}

}

## Service Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Alert;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TextArea;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

import verifications.Checks;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ServiceController implements Initializable {

@FXML private ImageView logo;

@FXML private ImageView logo2;

@FXML private ImageView housekeeping;

@FXML private ImageView cleaning;

@FXML private ImageView food;

@FXML private ImageView laundry;

@FXML private ImageView repair;

@FXML private Label label;

@FXML private ComboBox combo;

String usernameOfLoggedInPerson;

String customerID = "";

String houseID = "";

String roomID = "";

String branch = "";

public void setParameter(String s)

{

usernameOfLoggedInPerson = s;

System.out.println(s + "\n" + usernameOfLoggedInPerson);

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

System.out.println("Customer ID : " + customerID);

ResultSet rs = stmt.executeQuery("select customerID from customer where email = '" + usernameOfLoggedInPerson + "';");

while(rs.next())

customerID = rs.getString(1);

String house = "";

String room = "";

String b = "";

rs = stmt.executeQuery("select house.houseID, branch\_name from house\_booking, house, branch where house\_booking.houseID = house.houseID and branch.branchID = house.branchID and house\_booking.customerID = '"+ customerID +"';");

while(rs.next())

{

house = rs.getString(1);

b = rs.getString(2);

String temp = "house No." + house + " | " + b;

combo.getItems().add(temp);

}

rs = stmt.executeQuery("select room.room\_number, house.houseID, branch\_name, room.roomID from room\_booking, room, house, branch where room\_booking.roomID = room.roomID and room.houseID = house.houseID and branch.branchID = house.branchID and room\_booking.customerID = '"+ customerID +"';");

while(rs.next())

{

room = rs.getString(4);

b = rs.getString(3);

String temp = "Room No." + rs.getString(1) + " (Room ID : " + room + "), House No." + rs.getString(2) + " | " + b;

combo.getItems().add(temp);

}

}

catch(Exception e)

{

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

}

}

public void houseKeeping()

{

if(!combo.getValue().toString().equals("Select Your Booking First"))

{

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String billsID = "";

if(combo.getValue().toString().substring(0, 4).equals("Room"))

{

int counted = 0;

String s = combo.getValue().toString();

roomID = s.substring(s.indexOf("(")+11, s.indexOf(")"));

branch = s.substring(s.indexOf("|")+2, s.length());

ResultSet rs = stmt.executeQuery("select counts from bills, services where bills.serviceID = services.serviceID and service\_name = 'housekeeping' and customerID = '"+customerID+"' and bills.booking = 'r"+roomID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '1') and (`booking` = 'r"+roomID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '1', '1', 'r"+roomID+"');");

}

rs = stmt.executeQuery("Select counts, billsID from bills where customerID = '"+customerID+"' and serviceID = '1' and booking = 'r"+roomID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

billsID = rs.getString(2);

int bills = 200 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'r"+roomID+"');");

System.out.println("BRANCH : "+branch);

String empID = Checks.getEmpID("housekeeping", branch);

if(!empID.equals(""))

{

System.out.println("EMP ID : " + empID);

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

else

{

int counted = 0;

String s1 = combo.getValue().toString();

houseID = s1.substring(s1.indexOf(".")+1, s1.indexOf("|") - 1);

branch = s1.substring(s1.indexOf("|")+2, s1.length());

ResultSet rs = stmt.executeQuery("select counts, billsID from bills, services where bills.serviceID = services.serviceID and service\_name = 'housekeeping' and customerID = '"+customerID+"'and bills.booking = 'h"+houseID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

billsID = rs.getString(2);

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '1') and (`booking` = 'h"+houseID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '1', '"+1+"', 'h"+houseID+"');");

}

rs = stmt.executeQuery("Select counts from bills where customerID = '"+customerID+"' and serviceID = '1' and booking = 'h"+houseID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

int bills = 200 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'h"+houseID+"');");

String empID = Checks.getEmpID("housekeeping", branch);

if(!empID.equals(""))

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

label.setText("");

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("HOUSE KEEPING Request");

a1.setContentText("Request Accepted you will get the service after 30 mints");

a1.setHeaderText("For Details, Select Show Details!"); //you can also set header

String temp = "You have Requested For HOUSEKEEPING SERVICE !\nThe Service Charges will be 200 Rupees\nThe Billing Amount has been added to your Bills\nThe Service will be provide to you After Half an Hour\n";

TextArea ta = new TextArea(temp);

ta.setEditable(false);

a1.getDialogPane().setExpandableContent(ta);

a1.showAndWait();

}

catch(Exception e)

{

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

}

}

else

label.setText("Please First Select Your Booking");

}

public void food()

{

if(!combo.getValue().toString().equals("Select Your Booking First"))

{

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String billsID = "";

if(combo.getValue().toString().substring(0, 4).equals("Room"))

{

int counted = 0;

String s = combo.getValue().toString();

roomID = s.substring(s.indexOf("(")+11, s.indexOf(")"));

branch = s.substring(s.indexOf("|")+2, s.length());

ResultSet rs = stmt.executeQuery("select counts from bills, services where bills.serviceID = services.serviceID and service\_name = 'food' and customerID = '"+customerID+"' and bills.booking = 'r"+roomID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '4') and (`booking` = 'r"+roomID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '4', '1', 'r"+roomID+"');");

}

rs = stmt.executeQuery("Select counts, billsID from bills where customerID = '"+customerID+"' and serviceID = '4' and booking = 'r"+roomID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

billsID = rs.getString(2);

int bills = 250 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'r"+roomID+"');");

System.out.println("BRANCH : "+branch);

String empID = Checks.getEmpID("housekeeping", branch);

if(!empID.equals(""))

{

System.out.println("EMP ID : " + empID);

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

else

{

int counted = 0;

String s1 = combo.getValue().toString();

houseID = s1.substring(s1.indexOf(".")+1, s1.indexOf("|") - 1);

branch = s1.substring(s1.indexOf("|")+2, s1.length());

ResultSet rs = stmt.executeQuery("select counts, billsID from bills, services where bills.serviceID = services.serviceID and service\_name = 'food' and customerID = '"+customerID+"'and bills.booking = 'h"+houseID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

billsID = rs.getString(2);

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '4') and (`booking` = 'h"+houseID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '4', '"+1+"', 'h"+houseID+"');");

}

rs = stmt.executeQuery("Select counts from bills where customerID = '"+customerID+"' and serviceID = '4' and booking = 'h"+houseID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

int bills = 250 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'h"+houseID+"');");

String empID = Checks.getEmpID("cook", branch);

if(!empID.equals(""))

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

label.setText("");

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("FOOD Request");

a1.setContentText("Request Accepted you will get the service after 30 mints");

a1.setHeaderText("For Details, Select Show Details!"); //you can also set header

String temp = "You have Requested For FOOD SERVICE !\nThe Service Charges will be 250 Rupees\nThe Billing Amount has been added to your Bills\nThe Service will be provide to you After Half an Hour\n";

TextArea ta = new TextArea(temp);

ta.setEditable(false);

a1.getDialogPane().setExpandableContent(ta);

a1.showAndWait();

}

catch(Exception e)

{

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

}

}

else

label.setText("Please First Select Your Booking");

}

public void cleaning() //for cleaner must use term sweeper

{

if(!combo.getValue().toString().equals("Select Your Booking First"))

{

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String billsID = "";

if(combo.getValue().toString().substring(0, 4).equals("Room"))

{

int counted = 0;

String s = combo.getValue().toString();

roomID = s.substring(s.indexOf("(")+11, s.indexOf(")"));

branch = s.substring(s.indexOf("|")+2, s.length());

ResultSet rs = stmt.executeQuery("select counts from bills, services where bills.serviceID = services.serviceID and service\_name = 'cleaning' and customerID = '"+customerID+"' and bills.booking = 'r"+roomID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '3') and (`booking` = 'r"+roomID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '3', '1', 'r"+roomID+"');");

}

rs = stmt.executeQuery("Select counts, billsID from bills where customerID = '"+customerID+"' and serviceID = '3' and booking = 'r"+roomID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

billsID = rs.getString(2);

int bills = 100 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'r"+roomID+"');");

System.out.println("BRANCH : "+branch);

String empID = Checks.getEmpID("sweeper", branch);

if(!empID.equals(""))

{

System.out.println("EMP ID : " + empID);

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

else

{

int counted = 0;

String s1 = combo.getValue().toString();

houseID = s1.substring(s1.indexOf(".")+1, s1.indexOf("|") - 1);

branch = s1.substring(s1.indexOf("|")+2, s1.length());

ResultSet rs = stmt.executeQuery("select counts, billsID from bills, services where bills.serviceID = services.serviceID and service\_name = 'cleaning' and customerID = '"+customerID+"'and bills.booking = 'h"+houseID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

billsID = rs.getString(2);

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '3') and (`booking` = 'h"+houseID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '3', '"+1+"', 'h"+houseID+"');");

}

rs = stmt.executeQuery("Select counts from bills where customerID = '"+customerID+"' and serviceID = '3' and booking = 'h"+houseID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

int bills = 100 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'h"+houseID+"');");

String empID = Checks.getEmpID("sweeper", branch);

if(!empID.equals(""))

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

label.setText("");

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("CLEANING Request");

a1.setContentText("Request Accepted you will get the service after 30 mints");

a1.setHeaderText("For Details, Select Show Details!"); //you can also set header

String temp = "You have Requested For CLEANING SERVICE !\nThe Service Charges will be 100 Rupees\nThe Billing Amount has been added to your Bills\nThe Service will be provide to you After Half an Hour\n";

TextArea ta = new TextArea(temp);

ta.setEditable(false);

a1.getDialogPane().setExpandableContent(ta);

a1.showAndWait();

}

catch(Exception e)

{

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

}

}

else

label.setText("Please First Select Your Booking");

}

public void repairing()

{

if(!combo.getValue().toString().equals("Select Your Booking First"))

{

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String billsID = "";

if(combo.getValue().toString().substring(0, 4).equals("Room"))

{

int counted = 0;

String s = combo.getValue().toString();

roomID = s.substring(s.indexOf("(")+11, s.indexOf(")"));

branch = s.substring(s.indexOf("|")+2, s.length());

ResultSet rs = stmt.executeQuery("select counts from bills, services where bills.serviceID = services.serviceID and service\_name = 'repair' and customerID = '"+customerID+"' and bills.booking = 'r"+roomID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '2') and (`booking` = 'r"+roomID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '2', '1', 'r"+roomID+"');");

}

rs = stmt.executeQuery("Select counts, billsID from bills where customerID = '"+customerID+"' and serviceID = '2' and booking = 'r"+roomID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

billsID = rs.getString(2);

int bills = 150 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'r"+roomID+"');");

System.out.println("BRANCH : "+branch);

String empID = Checks.getEmpID("repairer", branch);

if(!empID.equals(""))

{

System.out.println("EMP ID : " + empID);

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

else

{

int counted = 0;

String s1 = combo.getValue().toString();

houseID = s1.substring(s1.indexOf(".")+1, s1.indexOf("|") - 1);

branch = s1.substring(s1.indexOf("|")+2, s1.length());

ResultSet rs = stmt.executeQuery("select counts, billsID from bills, services where bills.serviceID = services.serviceID and service\_name = 'repair' and customerID = '"+customerID+"'and bills.booking = 'h"+houseID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

billsID = rs.getString(2);

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '2') and (`booking` = 'h"+houseID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '2', '"+1+"', 'h"+houseID+"');");

}

rs = stmt.executeQuery("Select counts from bills where customerID = '"+customerID+"' and serviceID = '2' and booking = 'h"+houseID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

int bills = 150 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'h"+houseID+"');");

String empID = Checks.getEmpID("repairer", branch);

if(!empID.equals(""))

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

label.setText("");

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("REPAIR Request");

a1.setContentText("Request Accepted you will get the service after 30 mints");

a1.setHeaderText("For Details, Select Show Details!"); //you can also set header

String temp = "You have Requested For REPAIR SERVICE !\nThe Service Charges will be 150 Rupees\nThe Billing Amount has been added to your Bills\nThe Service will be provide to you After Half an Hour\n";

TextArea ta = new TextArea(temp);

ta.setEditable(false);

a1.getDialogPane().setExpandableContent(ta);

a1.showAndWait();

}

catch(Exception e)

{

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

}

}

else

label.setText("Please First Select Your Booking");

}

public void laundry()

{

if(!combo.getValue().toString().equals("Select Your Booking First"))

{

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String billsID = "";

if(combo.getValue().toString().substring(0, 4).equals("Room"))

{

int counted = 0;

String s = combo.getValue().toString();

roomID = s.substring(s.indexOf("(")+11, s.indexOf(")"));

branch = s.substring(s.indexOf("|")+2, s.length());

ResultSet rs = stmt.executeQuery("select counts from bills, services where bills.serviceID = services.serviceID and service\_name = 'laundry' and customerID = '"+customerID+"' and bills.booking = 'r"+roomID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '5') and (`booking` = 'r"+roomID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '5', '1', 'r"+roomID+"');");

}

rs = stmt.executeQuery("Select counts, billsID from bills where customerID = '"+customerID+"' and serviceID = '5' and booking = 'r"+roomID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

billsID = rs.getString(2);

int bills = 100 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'r"+roomID+"');");

System.out.println("BRANCH : "+branch);

String empID = Checks.getEmpID("laundryman", branch);

if(!empID.equals(""))

{

System.out.println("EMP ID : " + empID);

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

else

{

int counted = 0;

String s1 = combo.getValue().toString();

houseID = s1.substring(s1.indexOf(".")+1, s1.indexOf("|") - 1);

branch = s1.substring(s1.indexOf("|")+2, s1.length());

ResultSet rs = stmt.executeQuery("select counts, billsID from bills, services where bills.serviceID = services.serviceID and service\_name = 'laundry' and customerID = '"+customerID+"'and bills.booking = 'h"+houseID+"';");

if(rs.next())

{

String counts = rs.getString(1);

counted = Integer.parseInt(counts) + 1;

billsID = rs.getString(2);

int uq = stmt.executeUpdate("UPDATE `mydb`.`bills` SET `counts` = '"+counted+"' WHERE (`customerID` = '"+customerID+"') and (`serviceID` = '5') and (`booking` = 'h"+houseID+"');");

}

else

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`bills` (`customerID`, `serviceID`, `counts`, `booking`) VALUES ('"+customerID+"', '5', '"+1+"', 'h"+houseID+"');");

}

rs = stmt.executeQuery("Select counts from bills where customerID = '"+customerID+"' and serviceID = '5' and booking = 'h"+houseID+"';");

rs.next();

int countTemp = Integer.parseInt(rs.getString(1)) \* counted;

int bills = 100 \* counted; //Bills of service multiplied by counts for that bills;

int up = stmt.executeUpdate("UPDATE `mydb`.`customer\_record` SET `bills` = '"+bills+"' WHERE (`customerID` = '"+customerID+"') and (`booking` = 'h"+houseID+"');");

String empID = Checks.getEmpID("laundryman", branch);

if(!empID.equals(""))

{

int uq = stmt.executeUpdate("INSERT INTO `mydb`.`service\_record` (`billsID`, `empID`, `date`, `time`) VALUES ('"+billsID+"', '"+empID+"', '"+Checks.getCurrentdate()+"', '"+Checks.getCurrentTime()+"');");

}

else

System.out.println("No Employee is Available");

}

label.setText("");

Alert a1 = new Alert(Alert.AlertType.INFORMATION);

a1.setTitle("LAUNDRY Request");

a1.setContentText("Request Accepted you will get the service after 30 mints");

a1.setHeaderText("For Details, Select Show Details!"); //you can also set header

String temp = "You have Requested For LAUNDRY SERVICE !\nThe Service Charges will be 100 Rupees\nThe Billing Amount has been added to your Bills\nThe Service will be provide to you After Half an Hour\n";

TextArea ta = new TextArea(temp);

ta.setEditable(false);

a1.getDialogPane().setExpandableContent(ta);

a1.showAndWait();

}

catch(Exception e)

{

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

}

}

else

label.setText("Please First Select Your Booking");

}

//This Method is for moving back to previous page/Screen from which this scene/screen is called

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("CustomerProfile.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerProfileController controller = loader.getController();

controller.setParameter(usernameOfLoggedInPerson);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

logo2 = new ImageView("room.png");

housekeeping = new ImageView("housekeeping.png");

cleaning = new ImageView("cleaning.png");

food = new ImageView("food.png");

laundry = new ImageView("laundry.png");

repair = new ImageView("repair.png");

combo.setValue("Select Your Booking First");

// try

// {

//

// }

// catch(Exception e)

// {

//

// }

}

}

## Service Given To Customer 2 Controller:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ServicesGivenToCustomer2Controller implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<Customer> tableView;

@FXML private TableColumn<Customer, String> srNoC;

@FXML private TableColumn<Customer, String> serviceC;

@FXML private TableColumn<Customer, String> dateC;

@FXML private TableColumn<Customer, String> timeC;

@FXML private Label label;

@FXML private Label label2;

int total\_customer = 0;

String empID = "";

String customerCNIC = "";

String prevPageID = "";

public void setParameter(String empID, Customer customerCNIC) throws SQLException

{

this.empID = empID;

this.customerCNIC = customerCNIC.getCnic();

setTableData();

}

public void setParameter(String empID, Customer customerCNIC, String s) throws SQLException

{

this.empID = empID;

this.customerCNIC = customerCNIC.getCnic();

this.prevPageID = s;

setTableData();

}

public void setParameter(String empID, String customerCNIC, String s) throws SQLException

{

this.empID = empID;

this.customerCNIC = customerCNIC;

this.prevPageID = s;

setTableData();

}

public void setTableData() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int counter = 1;

ResultSet rs = stmt.executeQuery("select customerID from customer where CNIC = '"+customerCNIC+"';");

rs.next();

String customerID = rs.getString(1);

rs = stmt.executeQuery("select services.service\_name, service\_record.date, service\_record.time from bills, service\_record, services where services.serviceID = bills.serviceID and bills.billsID = service\_record.billsID and bills.customerID = '"+customerID+"';");

while(rs.next())

{

list.add(new Customer(Integer.toString(counter), rs.getString(1), rs.getString(2), rs.getString(3)));

counter++;

}

tableView.setItems(list);

}

public void backButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ServicesGivenToCustomer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ServicesGivenToCustomerController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

srNoC.setCellValueFactory(new PropertyValueFactory<Customer, String>("srNo"));

serviceC.setCellValueFactory(new PropertyValueFactory<Customer, String>("service"));

dateC.setCellValueFactory(new PropertyValueFactory<Customer, String>("date"));

timeC.setCellValueFactory(new PropertyValueFactory<Customer, String>("time"));

}

}

## Service Given To Customer:

/\*

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

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

\* and open the template in the editor.

\*/

package guesthousemanagement;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

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.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.stage.Stage;

/\*\*

\* FXML Controller class

\*

\* @author Hamza Barak

\*/

public class ServicesGivenToCustomerController implements Initializable {

@FXML private ImageView logo;

@FXML private TableView<Customer> tableView;

@FXML private TableColumn<Customer, String> srNoC;

@FXML private TableColumn<Customer, String> customerNameC;

@FXML private TableColumn<Customer, String> cnicC;

@FXML private ComboBox combo;

@FXML private TextField searchBar;

@FXML private Button search;

@FXML private Button DetailView;

@FXML private Label label;

@FXML private Label label2;

int total\_customer = 0;

String empID = "";

String prevPageID = "";

public void setParameter(String empID) throws SQLException

{

this.empID = empID;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from customer;");

while(rs.next())

total\_customer++;

label.setText("TOTAL CUSTOMERS : " + total\_customer);

setTableData();

}

public void setParameter(String empID, String s) throws SQLException

{

this.empID = empID;

this.prevPageID = s;

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select \* from customer;");

while(rs.next())

total\_customer++;

label.setText("TOTAL CUSTOMERS : " + total\_customer);

setTableData();

}

public void userClickOnTable()

{

this.DetailView.setDisable(false);

}

public void searchButtonPushed() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

if(combo.getValue().toString().equals("NAME"))

{

int counter = 1;

String name[] = searchBar.getText().split(" ");

String firstName = "";

String lastName = "";

if(name.length > 1)

{

firstName = name[0];

lastName = name[1];

}

else if(name.length == 1)

{

firstName = name[0];

}

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where first\_name = '"+ firstName +"' and last\_name = '"+ lastName +"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("FIRST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where first\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered First Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("LAST NAME"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where last\_name = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Enterd Last Name is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CNIC"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where CNIC = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered CNIC is Not Present in our Database");

}

else if(combo.getValue().toString().equals("E-MAIL"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where email = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered E-MAIL is Not Present in our Database");

}

else if(combo.getValue().toString().equals("CUSTOMER ID"))

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer where customerID = '"+ searchBar.getText()+"';");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

else

label2.setText("Your Entered Customer ID is Not Present in our Database");

}

else

{

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer;");

if(rs.next())

{

do{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}while(rs.next());

}

}

tableView.setItems(list);

}

public void setTableData() throws SQLException

{

ObservableList<Customer> list = FXCollections.observableArrayList();

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

int counter = 1;

ResultSet rs = stmt.executeQuery("select first\_name, last\_name, CNIC from customer;");

while(rs.next())

{

list.add(new Customer(Integer.toString(counter), rs.getString(1) + " " + rs.getString(2), rs.getString(3)));

counter++;

}

tableView.setItems(list);

}

public void seeGivenServicesButtonPushed(ActionEvent event) throws IOException, SQLException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("ServicesGivenToCustomer2.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

ServicesGivenToCustomer2Controller controller = loader.getController();

controller.setParameter(empID, tableView.getSelectionModel().getSelectedItem(), prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

public void backButtonPushed(ActionEvent event) throws IOException

{

FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("Customer.fxml"));

Parent tableViewParent = loader.load();

Scene tableViewScene = new Scene(tableViewParent);

String css = GuestHouseManagement.class.getResource("buttonStyles.css").toExternalForm();

tableViewScene.getStylesheets().add(css);

CustomerController controller = loader.getController();

controller.setParameter(empID, prevPageID);

Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();

window.setScene(tableViewScene);

window.show();

}

/\*\*

\* Initializes the controller class.

\*/

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

logo = new ImageView("logo1.png");

DetailView.setDisable(true);

combo.getItems().addAll("NAME", "FIRST NAME", "LAST NAME","CNIC", "E-MAIL", "CUSTOMER ID", "Show All Data");

combo.setValue("Enter Your Desired Input for Search ");

srNoC.setCellValueFactory(new PropertyValueFactory<Customer, String>("srNo"));

customerNameC.setCellValueFactory(new PropertyValueFactory<Customer, String>("customerName"));

cnicC.setCellValueFactory(new PropertyValueFactory<Customer, String>("cnic"));

}

}

## Checks

package verifications;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.SimpleDateFormat;

import java.time.LocalDate;

import java.time.Period;

import java.time.format.DateTimeFormatter;

import java.util.Date;

import java.util.Scanner;

public class Checks

{

//new password Verifier method, it should be called in order to verify the password for newly created account. or for to change passowrd

public static boolean newPasswordCheck(String password1, String password2)

{

if(password1.equals(password2))

return true;

else

return false;

}

//this method is to check for the correct input of First Name as it can contain spaces;

public static boolean alphaCheckIgnoreSpace(String name)

{

boolean check = true;

for(int i = 0; i < name.length(); i++)

{

if(!Character.isLetter(name.charAt(i)) && name.charAt(i) != ' ')

{

check = false;

break;

}

}

return check;

}

//this method is to check for the correct input of Last Name as it can contain spaces;

public static boolean alphaCheck(String name)

{

boolean check = true;

for(int i = 0; i < name.length(); i++)

{

if(!Character.isLetter(name.charAt(i)))

{

check = false;

break;

}

}

return check;

}

//this method is to check for the correct input of First Name as it can contain spaces;

public static boolean cnicCheck(String cnic)

{

boolean check = true; //0123456789 10 11 12 13 14

//37301-7256 8 3 0 - 3

if(cnic.length() == 15 && cnic.charAt(5) == '-' && cnic.charAt(13) == '-')

{

for(int i = 0; i < cnic.length(); i++)

{

if(!Character.isDigit(cnic.charAt(i)) && i != 5 && i != 13)

{

check = false;

break;

}

}

}

else

check = false;

return check;

}

//check the correct pattern of mobile number

public static boolean mobCheck(String num)

{

boolean check = true;

for(int i = 0; i < num.length(); i++)

{

if(num.length() != 11 || !Character.isDigit(num.charAt(i)))

{

check = false;

break;

}

}

return check;

}

public static String getCurrentdate()

{

LocalDate today = LocalDate.now();

int days = today.getDayOfMonth();

int month = today.getMonthValue();

int year = today.getYear();

String date = year + "-" + month + "-" + days;

return date;

}

public static Boolean checkLeavingDate(String date)

{

boolean check = false;

String date1[] = date.split("-");

LocalDate today = LocalDate.now();

int days = today.getDayOfMonth();

int month = today.getMonthValue();

int year = today.getYear();

if(!date.equals(today.toString()))

{

if(!(Integer.parseInt(date1[0]) < year))

{

if(!(Integer.parseInt(date1[1]) < month))

{

if(!(Integer.parseInt(date1[2]) <= days))

{

check = true;

}

}

}

}

return check;

}

public static boolean ageCheck(int bYear, int bMonth, int bDay)

{

boolean check = true;

LocalDate today = LocalDate.now(); //Today's date

LocalDate birthday = LocalDate.of(bYear, bMonth, bDay); //Birth date

Period period = Period.between(birthday, today);

//Now access the values as below

int days = period.getDays();

int month = period.getMonths();

int year = period.getYears();

if(year <= 18)

{

if(!(bYear == (today.getYear()-18) && ((month == 0 && days == 0) || bMonth > today.getMonthValue() || (bMonth == today.getMonthValue() && bDay >= today.getDayOfMonth()))))

check = false;

}

return check;

}

public static int ageReturner(int bYear, int bMonth, int bDay)

{

LocalDate today = LocalDate.now(); //Today's date

LocalDate birthday = LocalDate.of(bYear, bMonth, bDay); //Birth date

Period period = Period.between(birthday, today);

//Now access the values as below

int days = period.getDays();

int month = period.getMonths();

int year = period.getYears();

if(year <= 18)

{

if((bYear == (today.getYear()-18) && ((month == 0 && days == 0) || bMonth > today.getMonthValue() || (bMonth == today.getMonthValue() && bDay >= today.getDayOfMonth()))))

year = 18;

}

return year;

}

//verify either the entered value is digit or not

public static boolean digitCheck(String num)

{

boolean check = true;

for(int i = 0; i < num.length(); i++)

{

if(Character.isDigit(num.charAt(i)) != true)

{

check = false;

break;

}

}

return check;

}

public static LocalDate stringToLocalDateConverter(String date)

{

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");

//convert String to LocalDate

LocalDate localDate = LocalDate.parse(date, formatter);

return localDate;

}

public static String getCurrentTime()

{

SimpleDateFormat formatter = new SimpleDateFormat("hh:mm a");

Date date = new Date();

//System.out.println(formatter.format(date).toString());

return formatter.format(date).toString();

}

public static boolean mobCheckData(String mob) //this method verify the unity of mob number entered by user

{

boolean check = true;

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select phone\_number from customer\_phone where phone\_number = '" + mob + "';");

if(rs.next())

check = false;

}

catch(Exception ex)

{

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

}

return check;

}

public static boolean emailCheck(String email)

{

boolean check = false;

if(email.substring(email.length()-4, email.length()).equals(".com"))

{

for(int i = 0; i < email.length(); i++)

{

if(Character.isDigit(email.charAt(i)) || Character.isLetter(email.charAt(i)) ||email.charAt(i) == '.' || email.charAt(i) == '@' || email.charAt(i) == '\_')

check = true;

else

{

check = false;

break;

}

}

}

return check;

}

public static boolean emailCheckData(String email) //this method verify the unity of Email Entered By User

{

boolean check = true;

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select email from customer where email = '" + email + "';");

if(rs.next())

check = false;

}

catch(Exception ex)

{

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

}

return check;

}

//this function can throw an IOException

public static String getEmpID(String type, String branch) throws IOException//type can only be housekeeping, sweeper, laundryman, cook, repairer

{

String emp = "";

try

{

boolean bigCheck = true;

System.out.println("Branch in CHECKS : " + branch);

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

String tempEmp = "";

ResultSet rs = stmt.executeQuery("SELECT employee.empID from employee, employee\_type, type, branch where type.typeID = employee\_type.typeID and employee.empID = employee\_type.empID and branch.branchID = employee.branchID and type.type = '"+type+"' and branch.branch\_name = '"+branch+"';");

//rs = stmt.executeQuery("");

while(rs.next())

{

tempEmp = tempEmp + rs.getString(1) + "#";

}

String[] employeesDB = tempEmp.split("#");

int counted = 0;

Scanner reader = new Scanner(new FileInputStream(type + ".txt"));

if(reader.hasNextLine())

{

String employeesFile[] = reader.nextLine().split("#");

for(int i = 0; i < employeesDB.length; i++)

{

boolean check = true;

for(int j = 0; j < employeesFile.length; j++)

{

if(employeesDB[i].equals(employeesFile[j]))

{

check = false; // it get match

}

}

if(check)

{

emp = employeesDB[i];

counted = i;

bigCheck = false;

}

}

}

reader.close();

if(bigCheck)

{

emp = employeesDB[0];

PrintWriter pw = new PrintWriter(new FileOutputStream(type + ".txt"));

pw.print(employeesDB[0] + "#");

pw.close();

}

else

{

PrintWriter pw = new PrintWriter(new FileOutputStream(type + ".txt", true));

pw.print(employeesDB[counted] + "#");

pw.close();

}

}

catch(SQLException e)

{

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

}

return emp;

}

public static boolean cnicCheckData(String cnic) //this method verify the unity of cnic Entered By User

{

boolean check = true; //return false when cnic

try

{

Connection conn;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb?user=root&password=hamza");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("select CNIC from customer where CNIC = '" + cnic + "';");

if(rs.next())

check = false;

}

catch(Exception ex)

{

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

}

return check;

}

}