

PHUMULA HOTEL MANAGEMENT SYSTEM

Project Report

By: Letlhogonolo Chauke

Gideon Maloma

Khumo Mathe

Morongwa Leketi

201906152

201909412

201909409

201909407

Sefako Makgatho Health Sciences University 2023

INDEX

1.	Introduction	4
	Objective	5
	Feature	5
2.	Analysis	6
	History of Hotel Management System	6
	Cost Benefit Analysis	7
	Contribution of Hotel Management System	8
	Advantages of Hotel Management System	9
	Drawbacks in existing software	9
	Feasibility Study	10
	Requirement Specification	11
3.	Design Phase	13
	Class Diagram	13
	ER Diagram	14
	Flowchart	15
	Sequence Diagram	15
4.	Coding Phase with respective screenshots and database	16
5.	Testing Phase	124
6.	Future Scope	125
7.	References	126

INTRODUCTION

In our project “**Phumula Hotel Group Management System**” we have tried to show how the data/information in hotels is managed. This is just an overview of management in hotels. It manages and maintains the records of customers, rooms, employees and drivers in the hotel. The project is aimed to maintain the day-to-day state of admission/vacation of Residents, List of employees, room details etc. Main objective of this project is to provide a solution for hotels to manage most their work using computerized processes.

This software application will help admin to handle customer information, room allocation details, Payment details etc.

The rooms have different categories like single bed, double bed etc. so their charges and records will be maintained accordingly.

This software has been made in a user friendly interface, so that anyone can add, delete, update the entries and handle all the transactions easily. As a security we have provided Admin username and Password.

The project, Phumula Hotel Group Management System is a desktop-based application that allows the hotel manager to handle all hotel activities online. Interactive GUI and the ability to manage various rooms, employees, drivers and customers make this system very flexible and convenient. This application gives managers the power and flexibility to manage the entire system from a single online system. Hotel management project provides room booking, staff management and other necessary hotel management features. The system allows the manager to post available rooms in the system.

OBJECTIVE

The objective of the Phumula Hotel Group Management System is to develop a project which maintains booking of rooms, providing drivers, etc. It is a windows based application to be implemented in Java

The Purpose of the whole process is to ease the daily or regular activities of the Hotel Management into an automatic computerized retrievable process. The daily activities includes the Room activities, Entering details of the new customer check in, To allocate a room as per the customer need and interest, Recording the checkout time and details, Releasing or Empty of room and to record the process in a computer system for future.

Main objective of this project is to provide a solution for hotels to manage most their work using computerized processes. This software application will help admin to handle customer information, room allocation details, Payment details etc.

- Keeping records of admission of customers.
- Facilities provided by hotels are fully utilized in an effective and efficient manner.

FEATURES

1. Hotel Room Bookings and Reservation
2. Front Desk Operations
3. Managing Guest Profile
4. User Privilege and security control
5. Keeping a tab on guest check-in check-out status
6. Maintenance of hotel rooms.
7. Providing a user friendly interface.

ANALYSIS

Study Of Current System-

Current System of Hotel Management is fully based on paperwork. Hotel Management has to manage all records of customers and rooms on paper.

Problem and Weakness Of Current System-

Paperwork is a tedious job. There can be delay and the problem of human error in allocating rooms and providing resources.

HISTORY

Hotel Management Systems are able to automate the process of hotels. It is useful for authorities which keep track of all the users registered. The authority can hotel packages room detail and availability of rooms, online booking and other packages.

The following steps that give detailed information of the need for electronic managements are:

Performance

During the past several decades, the records are supposed to be manually handled for all activities. The manual handling of records is time consuming and highly prone to error. To improve the performance of the hotel management system, the computerized system had to be undertaken.

Efficiency

The basic need of this application is efficiency. The website should be efficient so that whatever the user submits any detail the application is updated immediately and automatically. These records will be useful for others instantly.

Control

The complete control of the electronic system is under the hands of authorized persons who have the password to access the system and illegal access is not permitted. Control is entirely in the hands of the administrator and the other members have the right to see the records not to change any transaction entry.

Security

Security is the main criteria for electronic hotel management systems. Since illegal access may corrupt the database and ensure protection of the stored data.

Cost Benefit Analysis

- Role Based Screen Access
- The roles can be any of the following:
- *Administrator Role* - is for provisioning of Register/Update/Remove/Transfer Contract.
- *Data Entry role* – for entering of basic form data
- *Supervisory role* – for review/approval/rejection/modification of data
- *Decision making role* – viewing of report and decision making

Contribution

Hotel Management Software is programmed to enable HR professionals for managing the data and information of the company's employees on a regular basis. The easy information regarding the employees is including contact details, investment details, and much more.

Accuracy:

Your system must be able to accommodate your employees and various customers or guests at your hotel. Once a good system is in place, you will have an accurate record of the check-in and check-out status of a customer. An automated system minimizes the possibility of human error, as the only real factor is whether employees remember to clock in and out.

Record-Keeping:

Another feature of a hotel management system is that it can be used to keep detailed and accurate records. These records can be stored in a main database on site, online or in an outside record-keeping facility. Record-keeping can help you monitor trends such as how much overtime you're paying and how many employees participate in your company retirement plans

ADVANTAGES OF PHUMULA HOTEL GROUP MANAGEMENT SYSTEM

- Sometime it happens that the rooms get booked soon when one visits the place therefore user can make advance booking using this system.
- It saves user time in searching a room.
- The system is useful as it calculates an exact cost of rooms for requested number of days.
- It saves organization resources and expenses.
- This system is effective and saves time and cost of users.
- It is less time consuming and provides maximum accuracy.

DRAWBACKS OF PHUMULA HOTEL GROUP MANAGEMENT SYSTEM

- The booking process usually requires a customer identity, which the system cannot detect.
- It requires a reliable internet connection

Feasibility Study

To analyze whether the software will meet organizational requirements Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed.

To determine whether the software can be implemented using the current technology and within the specified budget and schedule A feasibility study is a preliminary investigation of a proposed system to decide whether the system can run smoothly with the organization.

To determine whether the software can be integrated with other existing software. benefits that are expected and to decide will the organization go for it.

Three Types of feasibility study

1. Technical Feasibility.

Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required

2. Operational Feasibility.

Analyzes whether users will adapt to a new software. determines whether the organization is satisfied by the alternative solutions proposed by the software development team

3. Economical Feasibility

Cost of hardware, software, development team, and training. Feasible only if tangible and intangible benefits outweigh the cost. The cost for proposed online shopping system is outweighing the cost and efforts involved

REQUIREMENT SPECIFICATION

User Requirements

There should be software which allocates rooms automatically and maintains records of customers.

System Requirements

There should be a database backup of the hotel management system. There should be a Java supported framework for the system. Operating system should be WindowsXP or a higher version of windows.

User Interface

The User interface required to be developed for the system should be user friendly and attractive.

There are two sets of java APIs for graphics programming:

AWT (Abstract Windowing Toolkit) and Swing.

- ☐ AWT API was introduced in JDK 1.0. Most of the AWT components have become obsolete and should be replaced by newer Swing components.
- ☐ Swing API, a much more comprehensive set of graphics libraries that enhances the AWT, was introduced as part of Java Foundation Classes (JFC) after the release of JDK 1.1. JFC consists of Swing, Java2D, Accessibility, Internationalization, and Pluggable Look-and-Feel Support APIs. JFC was an add-on to JDK 1.1 but has been integrated into core Java since JDK 1.2.

For the development of this project both hardware and software were required. The hardware required is any machine compatible for running the softwares and resources required.

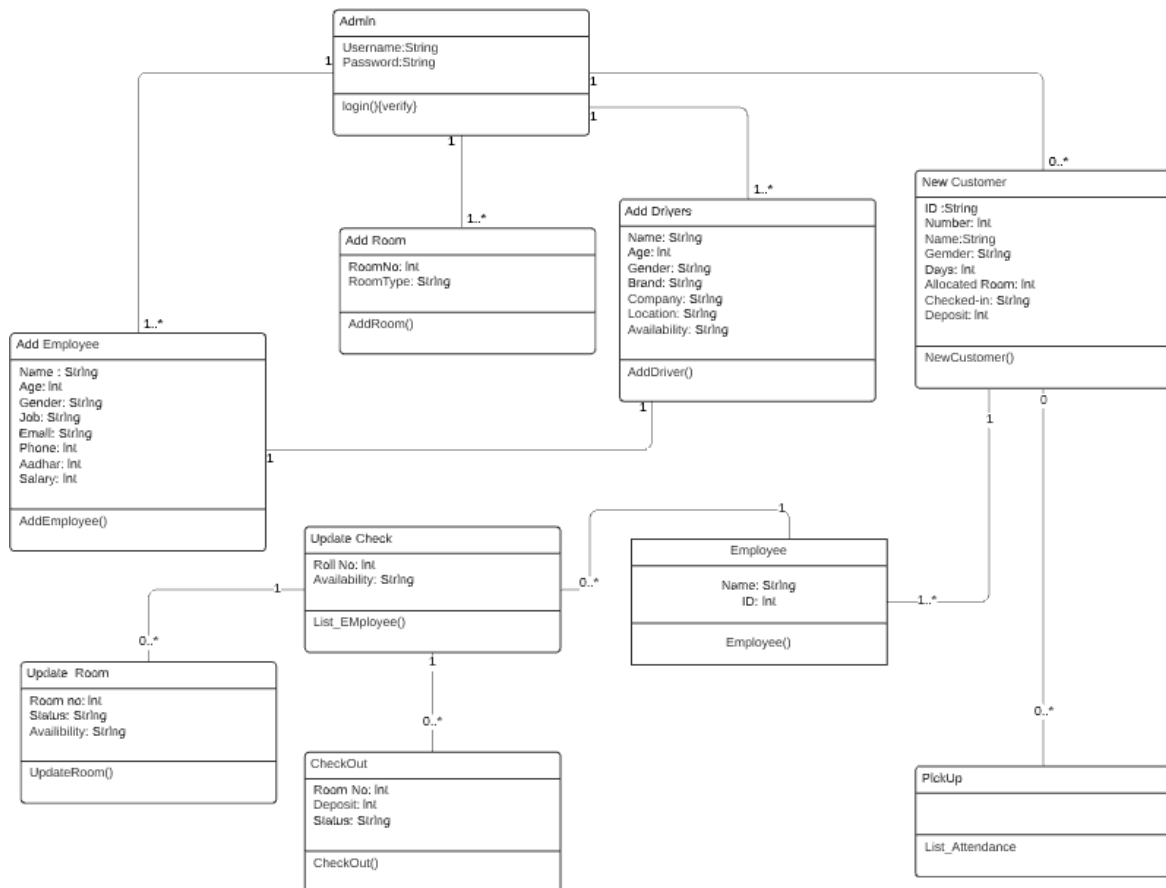
Software Requirements

- | | |
|--------------------|-------------------------|
| ● Operating System | Windows 8/10 |
| ● Front End | Java, Eclipse, Netbeans |
| ● Back End | MySQL |

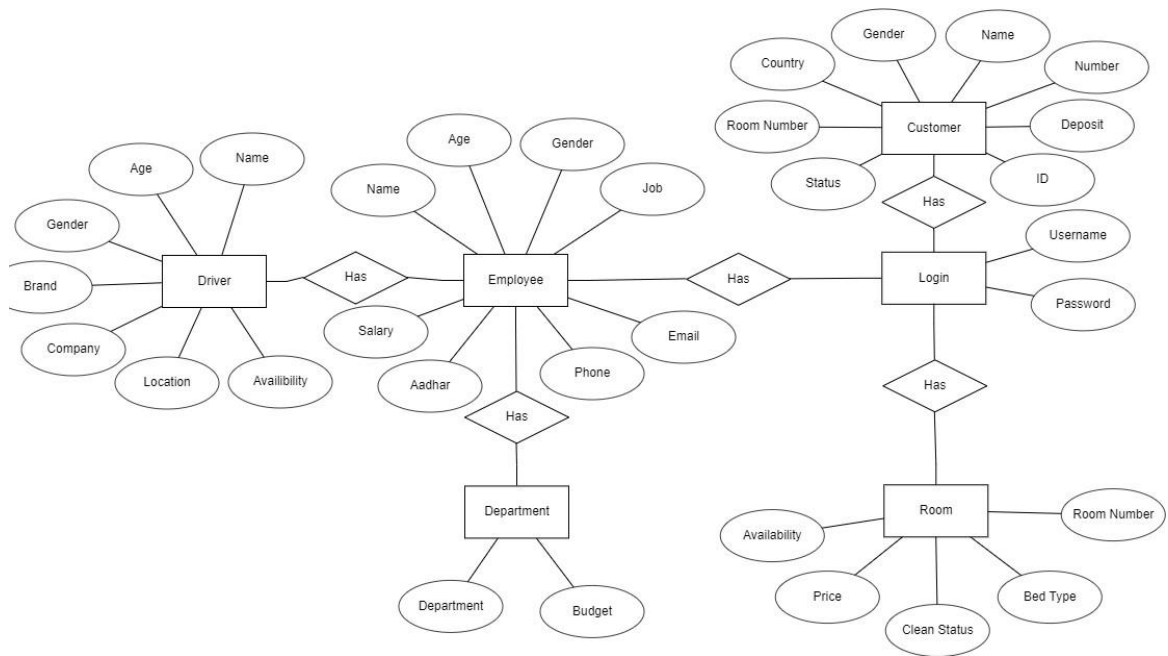
Hardware Requirements

- Desktop PC or a Laptop
- Printer
- Operating System – Windows 8/10
- Intel® Celeron/Core™ i3-6006U CPU @ 2.00GHz
- 4.00 GB RAM
- 64-bit operating system, x64 based processor
- Keyboard and Mouse

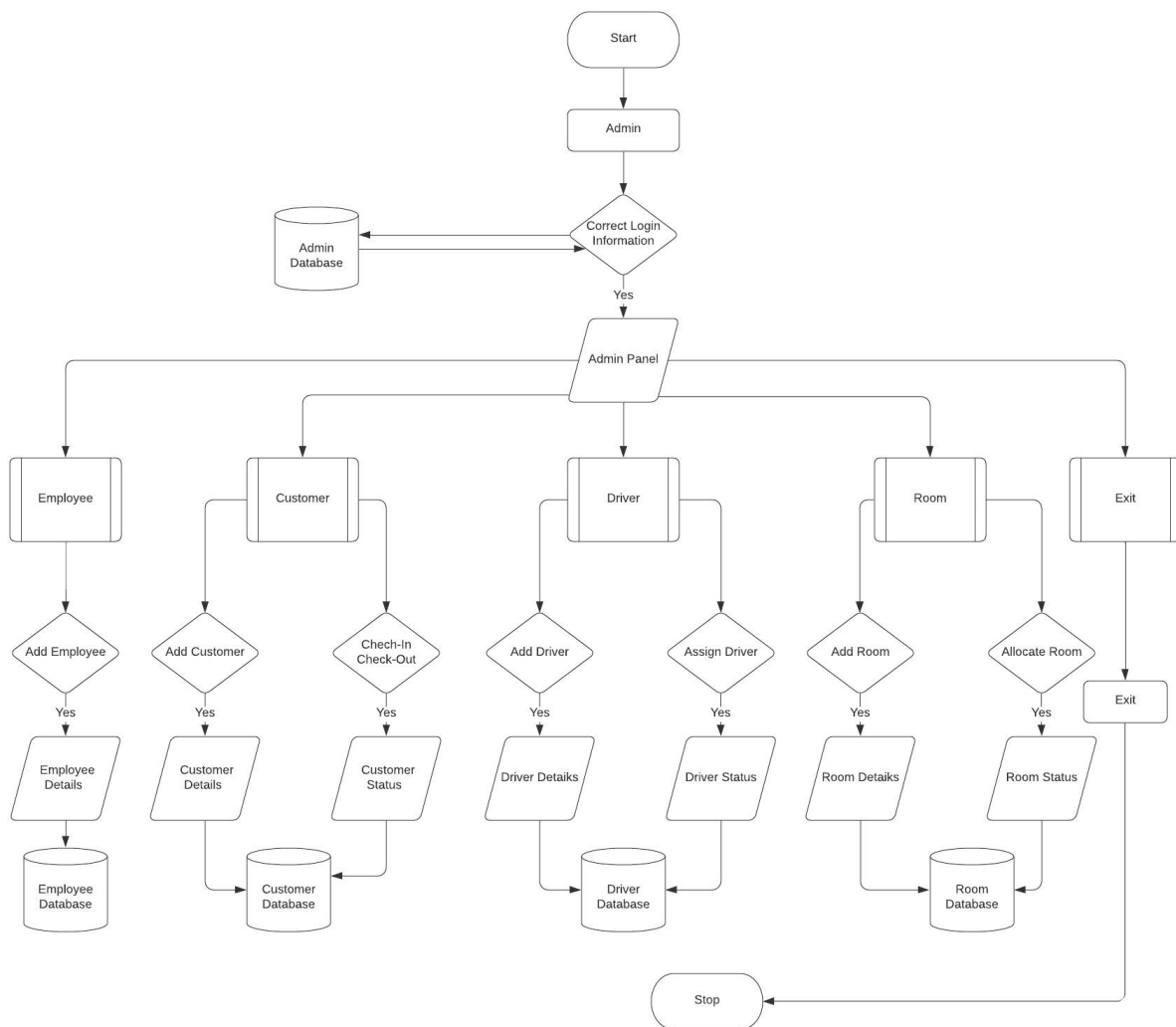
Class Diagram



ERDiagram



Flowchart



CODE

Connection

```
package hotel.management.system;

import java.sql.*;

public class conn {

    Connection c;
    Statement s;

    conn()
    {
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            c = DriverManager.getConnection("jdbc:mysql://localhost:3306/hotelmanagementsystem",
"root", "Menjukapoy123");
            s = c.createStatement();
        }

        catch (Exception e)
        {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        new conn();
    }
}
```

Hotel Management System (Main Page)

```
package hotel.management.system;
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*.*;

public class HotelManagementSystem extends JFrame implements ActionListener {

    HotelManagementSystem()
    {
        //setSize(1366, 565);
        //setLocation(100, 100);
        setBounds(100, 100, 1366, 565);
        setLayout(null);

        ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel
Management System\\src\\icons\\first.jpg");
        JLabel image = new JLabel(i1);
        image.setBounds(0, 0, 1366, 565);
        add(image);

        JLabel text = new JLabel("HOTEL MANAGEMENT SYSTEM");
        text.setBounds(20, 430, 1000, 90);
        text.setForeground(Color.WHITE);
        text.setFont(new Font("serif", Font.PLAIN, 50));
        image.add(text);

        JButton next = new JButton("Next");
        next.setBounds(1150, 450, 150, 50);
        next.setBackground(Color.WHITE);
        next.setForeground(Color.MAGENTA);

        next.addActionListener(this);

        next.setFont(new Font("serif", Font.PLAIN, 24));
        image.add(next);
```

```
setVisible(true);

while(true)
{
    text.setVisible(false);
    try
    {
        Thread.sleep(500);
    }

    catch(Exception e)
    {
        e.printStackTrace();
    }
    text.setVisible(true);

    try
    {
        Thread.sleep(500);
    }

    catch(Exception e)
    {
        e.printStackTrace();
    }
}
```

```
}

public void actionPerformed(ActionEvent e)
{
    setVisible(false);
    new Login();
}
```

```
public static void main(String[] args) {
    new HotelManagementSystem();
}
```



Login

```
package hotel.management.system;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;
import java.sql.*.*;

public class Login extends JFrame implements ActionListener {

    JTextField username;
    JPasswordField password;
    JButton login, cancel;

    Login()
    {
        getContentPane().setBackground(Color.WHITE);

        setLayout(null);
        JLabel user = new JLabel("Username");
        user.setBounds(40, 20, 100, 30);
        add(user);

        username = new JTextField();
        username.setBounds(150, 20, 150, 30);
        add(username);

        JLabel pass = new JLabel("Password");
        pass.setBounds(40, 70, 100, 30);
        add(pass);

        password = new JPasswordField();
        password.setBounds(150, 70, 150, 30);
        add(password);

        login = new JButton("Login");
        login.setBounds(40, 150, 120, 30);
        login.setBackground(Color.BLACK);
        login.setForeground(Color.WHITE);

        login.addActionListener(this);

        add(login);

        cancel = new JButton("Cancel");
        cancel.setBounds(180, 150, 120, 30);
        cancel.setBackground(Color.BLACK);
        cancel.setForeground(Color.WHITE);
```

```

        cancel.addActionListener(this);

        add(cancel);

        ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\second.jpg");
        Image i2 = i1.getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(350, 10, 200, 200);
        add(image);

        setBounds(500, 200, 600, 300);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        if (ae.getSource() == login) {

            String user = username.getText();
            String pass = password.getText();

            try
            {
                conn c = new conn();
                String query = "select * from login where  username = '" + user + "' and password = '"
+ pass + "'";

                ResultSet rs = c.s.executeQuery(query);

                if (rs.next())
                {
                    setVisible(false);
                    new Dashboard();
                }

                else
                {
                    JOptionPane.showMessageDialog(null, "Invalid username or password");
                    setVisible(false);
                }
            }

            catch (Exception e)
            {
                e.printStackTrace();
            }
        }

        else if (ae.getSource() == cancel)
        {
            setVisible(false);
        }
    }

```

```
}  
  
public static void main(String[] args) {  
    new Login();  
  
}  
}
```



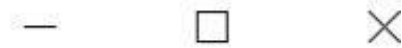
Login

Username

Password

Login

Cancel



Dashboard

```
package hotel.management.system;
```

```
import javax.swing.*.*;
```

```
import java.awt.*.*;
```

```
import java.awt.event.*;
```

```
public class Dashboard extends JFrame implements ActionListener {
```

```
    Dashboard()
```

```
    {
```

```
        setBounds(0, 0, 1550, 1000);
```

```
        setLayout(null);
```

```
        ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management System\\src\\icons\\third.jpg");
```

```
        Image i2 = i1.getImage().getScaledInstance(1550, 1000, Image.SCALE_DEFAULT);
```

```
        ImageIcon i3 = new ImageIcon(i2);
```

```
        JLabel image = new JLabel(i3);
```

```
        image.setBounds(0, 0, 1550, 1000);
```

```
        add(image);
```

```
        JLabel text = new JLabel("PHUMULA HOTEL GROUP WELCOMES YOU");
```

```
        text.setBounds(400, 80, 1000, 50);
```

```
        image.setFont(new Font("Tahoma", Font.PLAIN, 50));
```

```
        text.setForeground(Color.WHITE);
```

```
        image.add(text);
```

```
        JMenuBar mb = new JMenuBar();
```

```
        mb.setBounds(0, 0, 1550, 30);
```

```
        image.add(mb);
```

```
        JMenu hotel = new JMenu("HOTEL MANAGEMENT");
```

```
        hotel.setForeground(Color.RED);
```

```
        mb.add(hotel);
```

```
JMenuItem reception = new JMenuItem("RECEPTION");

reception.addActionListener(this);

hotel.add(reception);

JMenu admin = new JMenu("ADMIN");
admin.setForeground(Color.BLUE);
mb.add(admin);

JMenuItem addEmployee = new JMenuItem("ADD EMPLOYEE");

addEmployee.addActionListener(this);

admin.add(addEmployee);

JMenuItem addRooms = new JMenuItem("ADD ROOMS");

addRooms.addActionListener(this);

admin.add(addRooms);

JMenuItem addDrivers = new JMenuItem("ADD DRIVERS");

addDrivers.addActionListener(this);

admin.add(addDrivers);

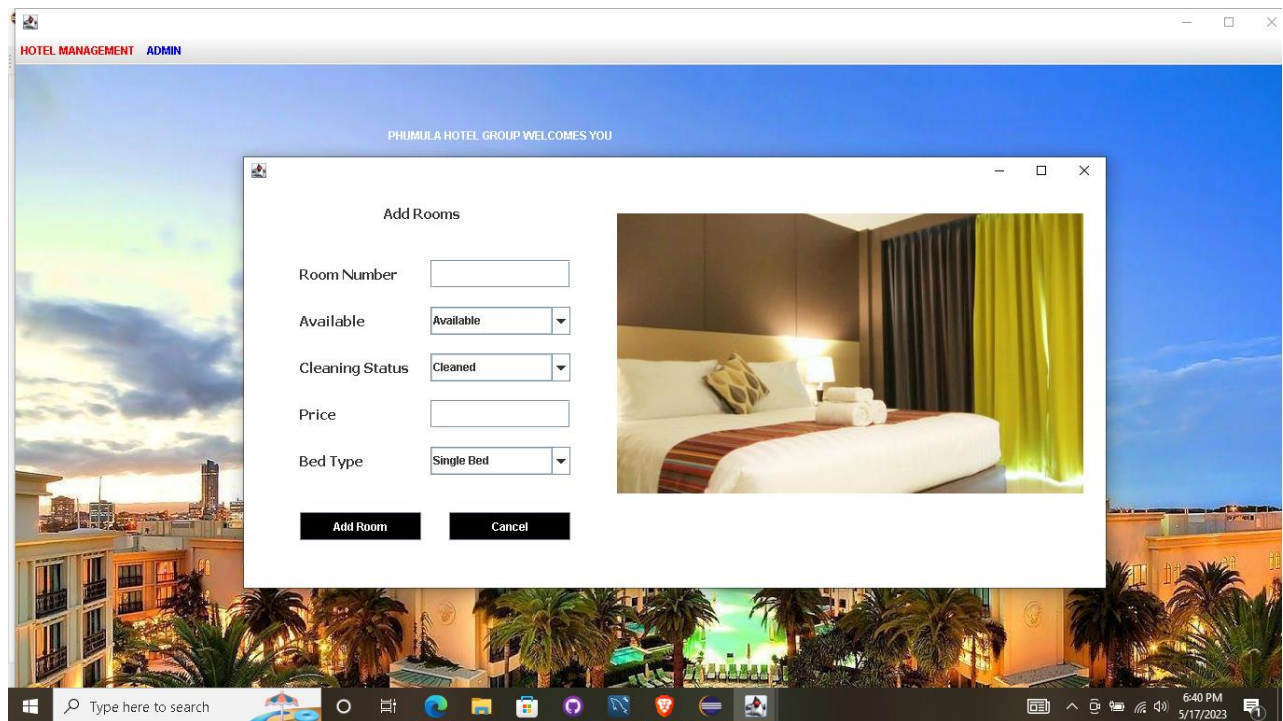
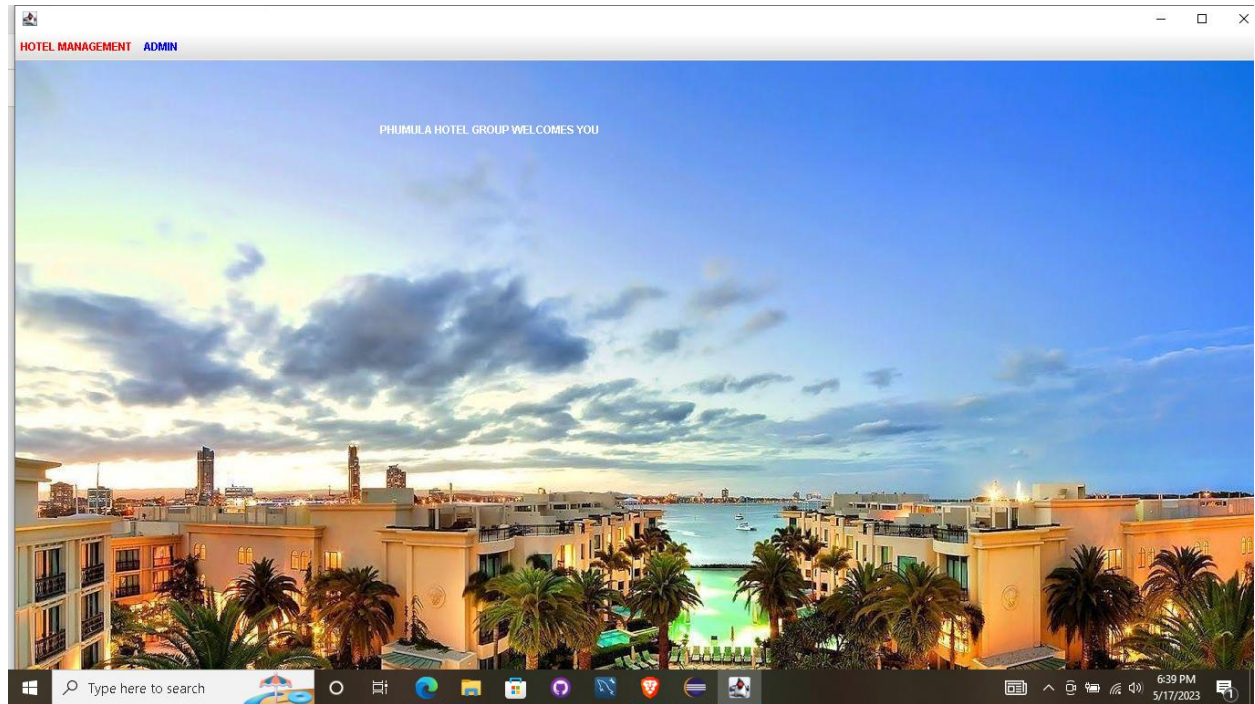
setVisible(true);

}

public void actionPerformed(ActionEvent ae)
{
    if (ae.getActionCommand().equals("ADD EMPLOYEE"))
    {
```

```
        new AddEmployee();
    }
    else if (ae.getActionCommand().equals("ADD ROOMS"))
    {
        new AddRoom();
    }
    else if (ae.getActionCommand().equals("ADD DRIVERS"))
    {
        new AddDrivers();
    }
    else if (ae.getActionCommand().equals("RECEPTION"))
    {
        new Reception();
    }
}

public static void main(String[] args)
{
    new Dashboard();
}
}
```



Reception

```
package hotel.management.system;

import javax.swing.*.*;

import java.sql.*;
import java.awt.event.*;
import java.awt.*.*;

public class Reception extends JFrame implements ActionListener {
    JButton newCustomer, rooms, department, allEmployee, customers, managerInfo, checkout, searchRoom,
    update, roomStatus, pickup, logout;

    Reception()
    {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        newCustomer = new JButton("New Customer Form");
        newCustomer.setBounds(10, 30, 200, 30);
        newCustomer.setBackground(Color.BLACK);
        newCustomer.setForeground(Color.WHITE);
        add(newCustomer);

        newCustomer.addActionListener(this);

        rooms = new JButton("Rooms");
        rooms.setBounds(10, 70, 200, 30);
        rooms.setBackground(Color.BLACK);
        rooms.setForeground(Color.WHITE);
        add(rooms);

        rooms.addActionListener(this);

        department = new JButton("Department");
        department.setBounds(10, 110, 200, 30);
        department.setBackground(Color.BLACK);
        department.setForeground(Color.WHITE);
        add(department);

        department.addActionListener(this);

        allEmployee = new JButton("All Employees");
        allEmployee.setBounds(10, 150, 200, 30);
        allEmployee.setBackground(Color.BLACK);
        allEmployee.setForeground(Color.WHITE);
        add(allEmployee);

        allEmployee.addActionListener(this);
```

```
customers = new JButton("Customer Info");
customers.setBounds(10, 190, 200, 30);
customers.setBackground(Color.BLACK);
customers.setForeground(Color.WHITE);
add(customers);

customers.addActionListener(this);

managerInfo = new JButton("Manager Info");
managerInfo.setBounds(10, 230, 200, 30);
managerInfo.setBackground(Color.BLACK);
managerInfo.setForeground(Color.WHITE);
add(managerInfo);

managerInfo.addActionListener(this);

checkout = new JButton("Checkout");
checkout.setBounds(10, 270, 200, 30);
checkout.setBackground(Color.BLACK);
checkout.setForeground(Color.WHITE);
add(checkout);

checkout.addActionListener(this);

update = new JButton("Update Status");
update.setBounds(10, 310, 200, 30);
update.setBackground(Color.BLACK);
update.setForeground(Color.WHITE);
add(update);

update.addActionListener(this);

roomStatus = new JButton("Update Room Status");
roomStatus.setBounds(10, 350, 200, 30);
roomStatus.setBackground(Color.BLACK);
roomStatus.setForeground(Color.WHITE);
add(roomStatus);

roomStatus.addActionListener(this);

pickup = new JButton("Pickup Service");
pickup.setBounds(10, 390, 200, 30);
pickup.setBackground(Color.BLACK);
pickup.setForeground(Color.WHITE);
add(pickup);

pickup.addActionListener(this);

searchRoom = new JButton("Search Room");
searchRoom.setBounds(10, 430, 200, 30);
searchRoom.setBackground(Color.BLACK);
searchRoom.setForeground(Color.WHITE);
add(searchRoom);
```

```
searchRoom.addActionListener(this);

logout = new JButton("Logout");
logout.setBounds(10, 470, 200, 30);
logout.setBackground(Color.BLACK);
logout.setForeground(Color.WHITE);
add(logout);

logout.addActionListener(this);

ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\fourth.jpg");
JLabel image = new JLabel(i1);
image.setBounds(250, 30, 500, 470);
add(image);

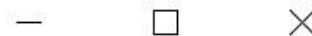
setBounds(350, 200, 800, 570);
setVisible(true);
}

public void actionPerformed(ActionEvent ae)
{
    if (ae.getSource() == newCustomer)
    {
        setVisible(false);
        new NewCustomer();
    }
    else if (ae.getSource() == rooms)
    {
        setVisible(false);
        new Room();
    }
    else if (ae.getSource() == department)
    {
        setVisible(false);
        new Department();
    }
    else if (ae.getSource() == allEmployee)
    {
        setVisible(false);
        new EmployeeInfo();
    }
    else if (ae.getSource() == managerInfo)
    {
        setVisible(false);
        new ManagerInfo();
    }
    else if (ae.getSource() == customers)
    {
        setVisible(false);
        new CustomerInfo();
    }
}
```

```
    else if (ae.getSource() == searchRoom)
    {
        setVisible(false);
        new SearchRoom();
    }
    else if (ae.getSource() == update)
    {
        setVisible(false);
        new UpdateCheck();
    }
    else if (ae.getSource() == roomStatus)
    {
        setVisible(false);
        new UpdateRoom();
    }
    else if (ae.getSource() == pickup)
    {
        setVisible(false);
        new Pickup();
    }
    else if (ae.getSource() == checkout)
    {
        setVisible(false);
        new Checkout();
    }
    else if (ae.getSource() == logout)
    {
        setVisible(false);
        System.exit(0);
    }
}

public static void main(String[] args) {
    new Reception();
}
```

}



- New Customer Form
- Room
- Department
- All Employee Info
- Customer Info
- Manager Info
- Check Out
- Update Check Status
- Update Room Status
- Pick up Service
- Search Room
- Log Out



New Customer Form

```
package hotel.management.system;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*.*;
import java.sql.*.*;
import java.util.Date;

public class NewCustomer extends JFrame implements ActionListener {

    JComboBox comboid;
    JTextField tfnumber, tfname, tfcountry, tfdeposit;
    JRadioButton rmale, rfemale;
    Choice croom;
    JLabel checkintime;
    JButton add, back;

    NewCustomer()
    {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel text = new JLabel("NEW CUSTOMER FORM");
        text.setBounds(100, 20, 300, 30);
        text.setFont(new Font("Raleway", Font.PLAIN, 20));
        add(text);

        JLabel lblid = new JLabel("ID");
        lblid.setBounds(35, 80, 100, 20);
        lblid.setFont(new Font("Raleway", Font.PLAIN, 20));
        add(lblid);

        String options[] = {"Smart ID card", "Passport", "Driving License", "Birth Certificate"};
        comboid = new JComboBox(options);
        comboid.setBounds(200, 80, 150, 25);
        comboid.setBackground(Color.WHITE);
        add(comboid);

        JLabel lblnumber = new JLabel("Number");
        lblnumber.setBounds(35, 120, 100, 20);
        lblnumber.setFont(new Font("Raleway", Font.PLAIN, 20));
        add(lblnumber);

        tfnumber = new JTextField();
        tfnumber.setBounds(200, 120, 150, 25);
        add(tfnumber);

        JLabel lblname = new JLabel("Name");
        lblname.setBounds(35, 160, 100, 20);
```

```
lblname.setFont(new Font("Raleway", Font.PLAIN, 20));
add(lblname);

tfname = new JTextField();
tfname.setBounds(200, 160, 150, 25);
add(tfname);

JLabel lblgender = new JLabel("Gender");
lblgender.setBounds(35, 200, 100, 20);
lblgender.setFont(new Font("Raleway", Font.PLAIN, 20));
add(lblgender);

rmale = new JRadioButton("Male");
rmale.setBackground(Color.WHITE);
rmale.setBounds(200, 200, 60, 25);
add(rmale);

rfemale = new JRadioButton("Female");
rfemale.setBackground(Color.WHITE);
rfemale.setBounds(260, 200, 70, 25);
add(rfemale);

ButtonGroup bg = new ButtonGroup();
bg.add(rmale);
bg.add(rfemale);

JLabel lblcountry = new JLabel("Country");
lblcountry.setBounds(35, 240, 100, 20);
lblcountry.setFont(new Font("Raleway", Font.PLAIN, 20));
add(lblcountry);

tfcountry = new JTextField();
tfcountry.setBounds(200, 240, 150, 25);
add(tfcountry);

JLabel lblroom = new JLabel("Room Number");
lblroom.setBounds(35, 280, 150, 20);
lblroom.setFont(new Font("Raleway", Font.PLAIN, 20));
add(lblroom);

croom = new Choice();

try
{
    conn conn = new conn();
    String query = "select * from room where availability = 'Available'";
    ResultSet rs = conn.s.executeQuery(query);

    while (rs.next())
    {
        croom.add(rs.getString("roomnumber"));
    }
}
```

```
catch (Exception e)
{
    e.printStackTrace();
}
croom.setBounds(200, 280, 150, 25);
add(croom);

JLabel lbltime = new JLabel("Check-in Time");
lbltime.setBounds(35, 320, 150, 20);
lbltime.setFont(new Font("Raleway", Font.PLAIN, 20));
add(lbltime);

Date date = new Date();

checkintime = new JLabel("" + date);
checkintime.setBounds(200, 320, 150, 25);
checkintime.setFont(new Font("Raleway", Font.PLAIN, 16));
add(checkintime);

JLabel lbldeposit = new JLabel("Deposit");
lbldeposit.setBounds(35, 360, 100, 20);
lbldeposit.setFont(new Font("Raleway", Font.PLAIN, 20));
add(lbldeposit);

tfdeposit = new JTextField();
tfdeposit.setBounds(200, 360, 150, 25);
add(tfdeposit);

add = new JButton("Add");
add.setBackground(Color.BLACK);
add.setForeground(Color.WHITE);
add.setBounds(50, 410, 120, 30);
add(add);

add.addActionListener(this);

back = new JButton("Back");
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);
back.setBounds(200, 410, 120, 30);
add(back);

back.addActionListener(this);

ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\fifth.png");
Image i2 = i1.getImage().getScaledInstance(300, 400, Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel image = new JLabel(i3);
image.setBounds(400, 50, 300, 400);
add(image);
```

```
        setBounds(350, 200, 800, 550);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        if (ae.getSource() == add)
        {
            String id = (String) comboid.getSelectedItem();
            String number = tfnumber.getText();
            String name = tfname.getText();
            String gender = null;

            if (rmale.isSelected())
            {
                gender = "Male";
            }
            else
            {
                gender = "Female";
            }
            String country = tfcountry.getText();
            String roomnumber = croom.getSelectedItem();
            String time = checkintime.getText();
            String deposit = tfdeposit.getText();

            try
            {
                String query = "insert into customer(document, number, name, gender, country, room, checkintime, deposit)
values('" + id + "', '" + number + "', '" + name + "', '" + gender + "', '" + country + "', '" + roomnumber + "', '" + time + "', '" + deposit + "')";
                String query2 = "update room set availability = 'Occupied' where roomnumber = '" + roomnumber + "'";

                conn conn = new conn();

                conn.s.executeUpdate(query);
                conn.s.executeUpdate(query2);

                JOptionPane.showMessageDialog(null, "New Customer Added Successfully");

                setVisible(false);
                new Reception();

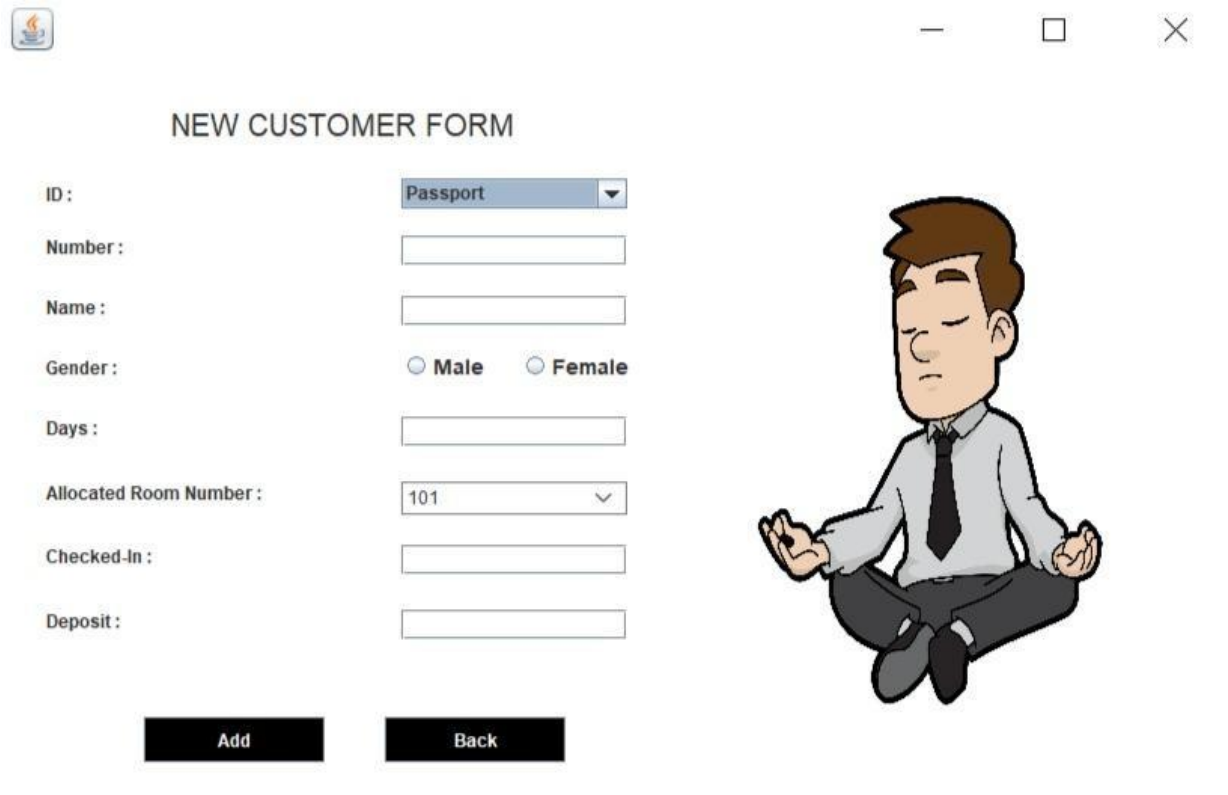
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
        else if (ae.getSource() == back)
        {
            setVisible(false);
        }
    }
}
```

```

    new Reception();
}

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

```



NEW CUSTOMER FORM

ID :

Number :

Name :

Gender : ☐ Male ☐ Female

Days :

Allocated Room Number :

Checked-In :

Deposit :

Room

```
package hotel.management.system;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import net.proteanit.sql.*;
```

```
public class Room extends JFrame implements ActionListener {
```

```
    JTable table;
    JButton back;
```

```
Room()
{
    getContentPane().setBackground(Color.WHITE);
    setLayout(null);

    ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\eight.jpg");
    Image i2 = i1.getImage().getScaledInstance(600, 600, Image.SCALE_DEFAULT);
    ImageIcon i3 = new ImageIcon(i2);
    JLabel image = new JLabel(i3);
    image.setBounds(500, 0, 600, 600);
    add(image);

    JLabel l1 = new JLabel("Room Number");
    l1.setBounds(10, 10, 100, 20);
    add(l1);

    JLabel l2 = new JLabel("Availability");
    l2.setBounds(120, 10, 100, 20);
    add(l2);

    JLabel l3 = new JLabel("Status");
    l3.setBounds(230, 10, 100, 20);
    add(l3);

    JLabel l4 = new JLabel("Price");
    l4.setBounds(330, 10, 100, 20);
    add(l4);

    JLabel l5 = new JLabel("Bed Type");
    l5.setBounds(410, 10, 100, 20);
    add(l5);

    table = new JTable();
    table.setBounds(0, 40, 500, 400);
    add(table);

    try
    {
        conn c = new conn();
        ResultSet rs = c.s.executeQuery("select * from room");
        table.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }

    back = new JButton("Back");
    back.setBounds(200, 500, 120, 30);
    back.setBackground(Color.BLACK);
    back.setForeground(Color.WHITE);
```

```

        back.addActionListener(this);

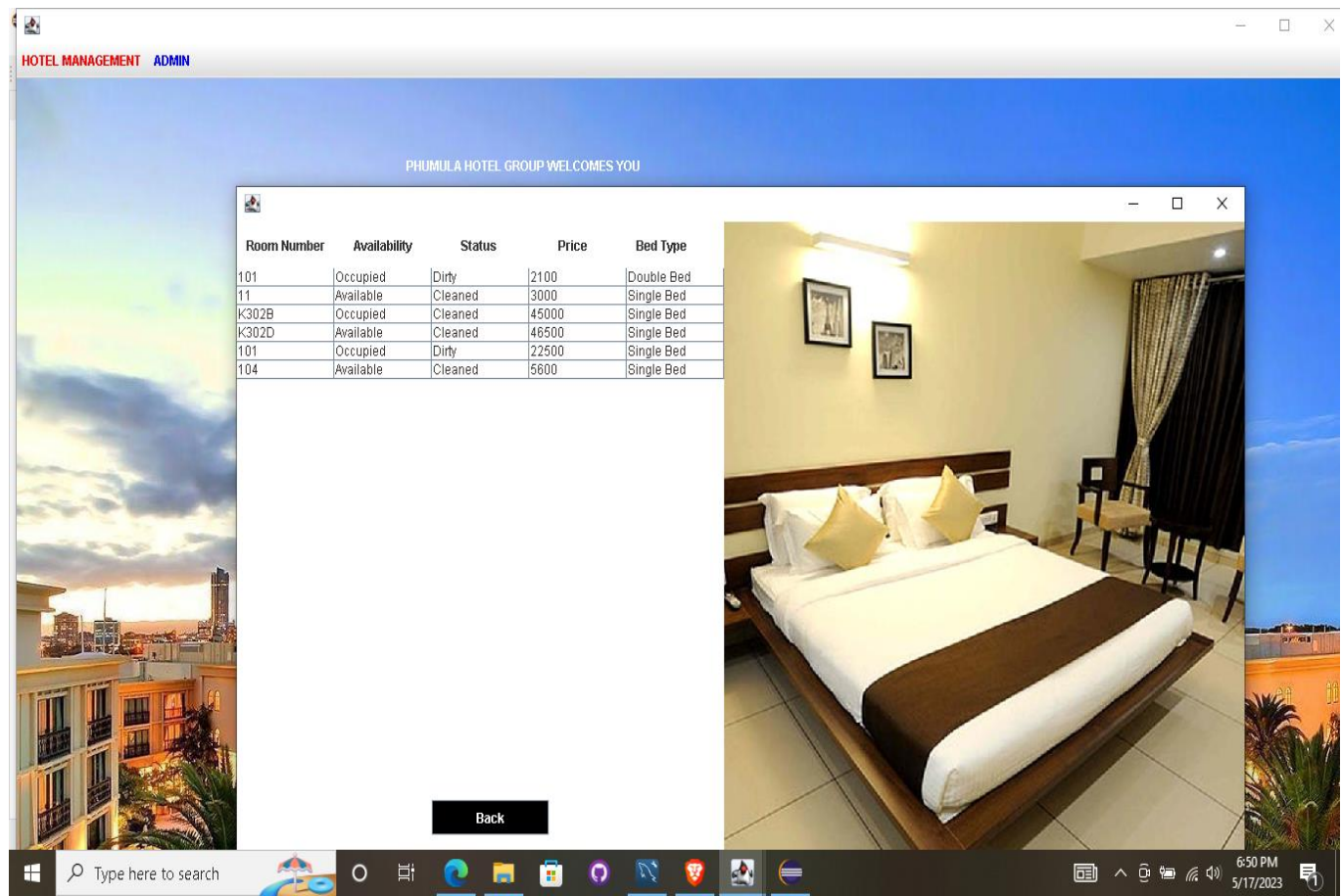
        add(back);

        setBounds(300, 200, 1050, 600);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        setVisible(false);
        new Reception();
    }

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

```



Department

```
package hotel.management.system;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;
import java.sql.*.*;
import net.proteanit.sql.*;

public class Department extends JFrame implements ActionListener {

    JTable table;
    JButton back;

    Department()
    {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel l1 = new JLabel("Department");
        l1.setBounds(180, 10, 100, 20);
        add(l1);

        JLabel l2 = new JLabel("Budget");
        l2.setBounds(420, 10, 100, 20);
        add(l2);

        table = new JTable();
        table.setBounds(0, 50, 700, 350);
        add(table);

        try
        {
            conn c = new conn();
            ResultSet rs = c.s.executeQuery("select * from department");
            table.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        back = new JButton("Back");
        back.setBounds(280, 400, 120, 30);
        back.setBackground(Color.BLACK);
```

```

        back.setForeground(Color.WHITE);

        back.addActionListener(this);

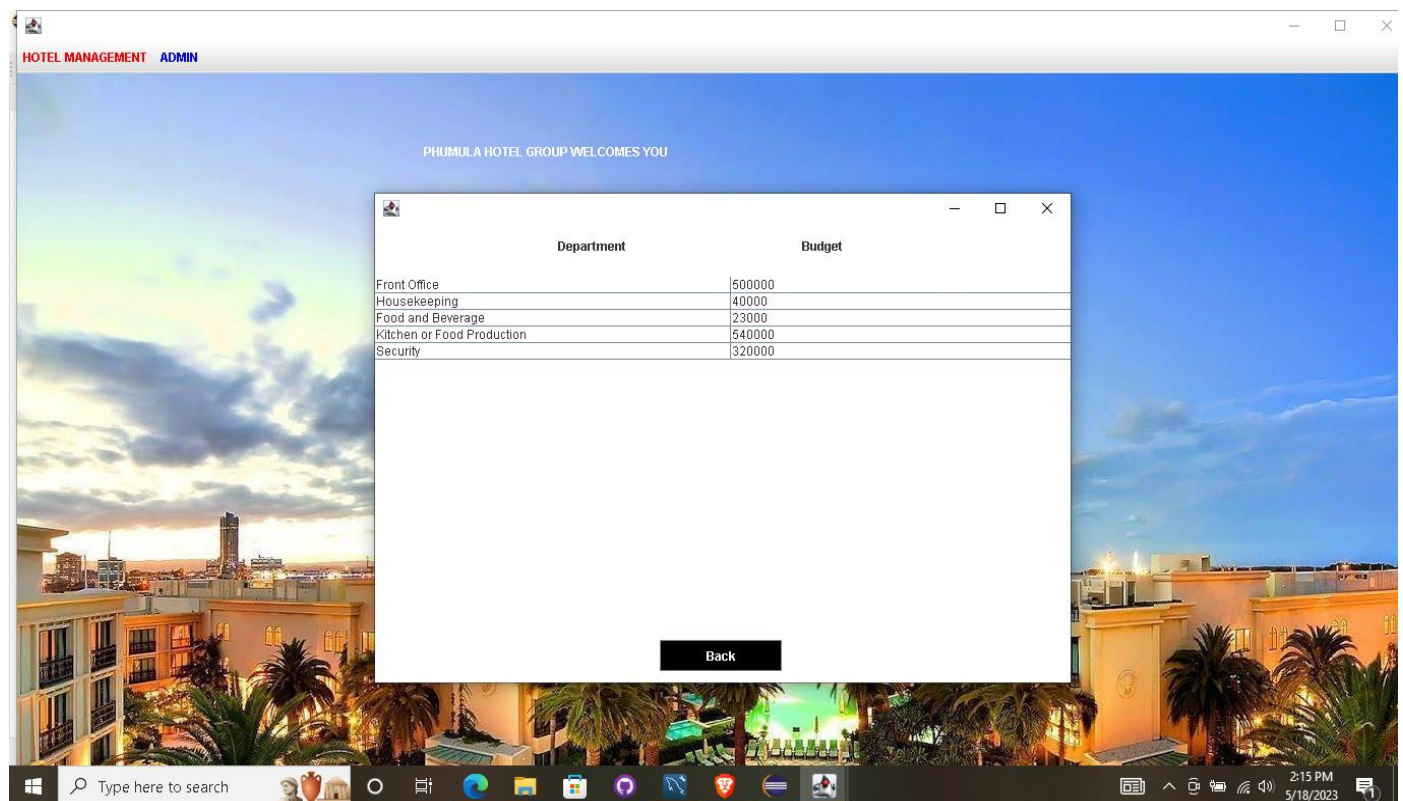
        add(back);

        setBounds(400, 200, 700, 480);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        setVisible(false);
        new Reception();
    }

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

```



Employee

```
package hotel.management.system;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;
import java.sql.*;
import net.proteanit.sql.*;

public class EmployeeInfo extends JFrame implements ActionListener {

    JTable table;
    JButton back;

    EmployeeInfo()
    {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel l1 = new JLabel("Name");
        l1.setBounds(40, 10, 100, 20);
        add(l1);

        JLabel l2 = new JLabel("Age");
        l2.setBounds(170, 10, 100, 20);
        add(l2);

        JLabel l3 = new JLabel("Gender");
        l3.setBounds(290, 10, 100, 20);
        add(l3);

        JLabel l4 = new JLabel("Job");
        l4.setBounds(400, 10, 100, 20);
        add(l4);

        JLabel l5 = new JLabel("Salary");
        l5.setBounds(540, 10, 100, 20);
        add(l5);

        JLabel l6 = new JLabel("Phone Num");
        l6.setBounds(640, 10, 100, 20);
        add(l6);

        JLabel l7 = new JLabel("Email");
        l7.setBounds(790, 10, 100, 20);
        add(l7);

        JLabel l8 = new JLabel("ID Num");
        l8.setBounds(910, 10, 100, 20);
        add(l8);
```

```
        table = new JTable();
        table.setBounds(0, 40, 1000, 400);
        add(table);

        try
        {
            conn c = new conn();
            ResultSet rs = c.s.executeQuery("select * from employee");
            table.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        back = new JButton("Back");
        back.setBounds(200, 500, 120, 30);
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);

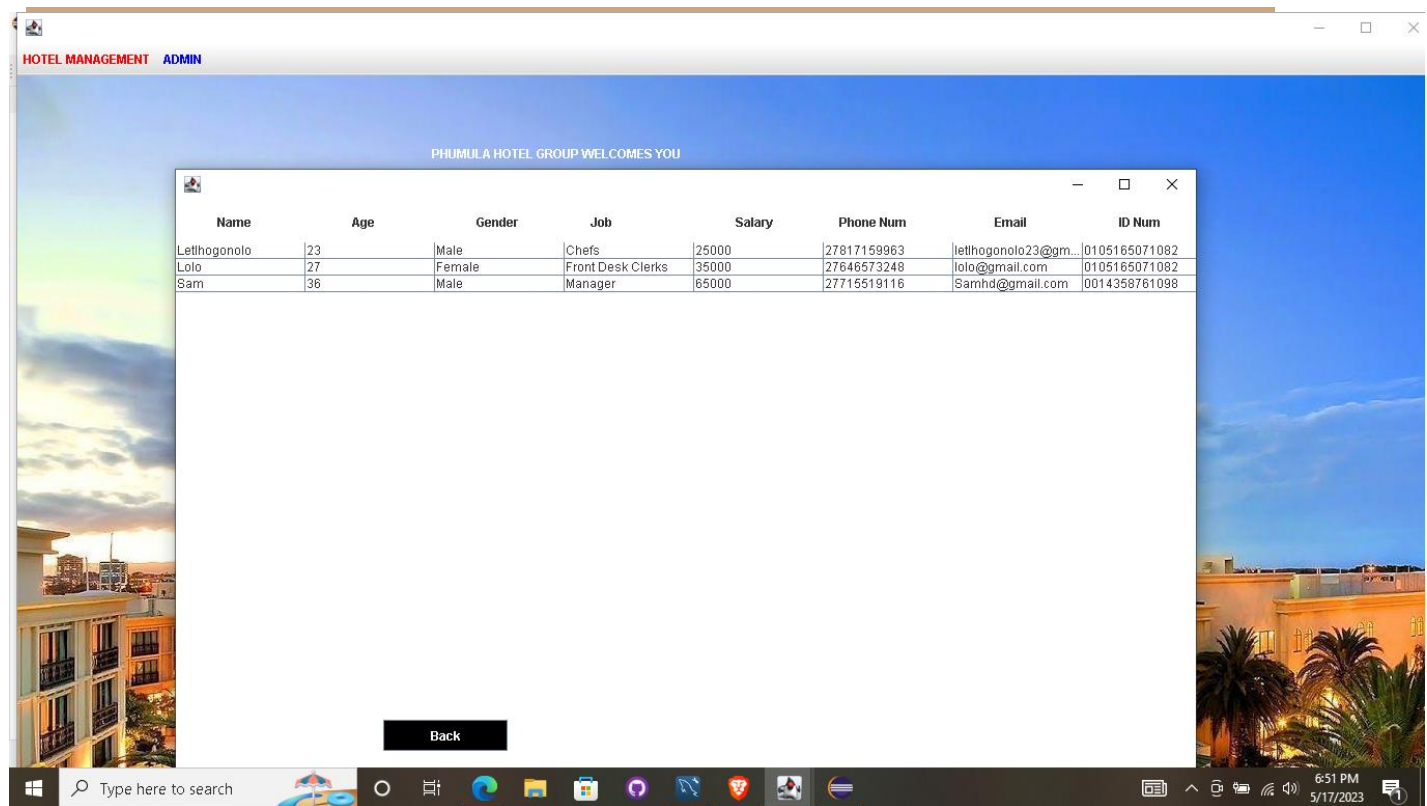
        back.addActionListener(this);

        add(back);

        setBounds(420, 200, 1000, 600);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        setVisible(false);
        new Reception();
    }

    public static void main(String[] args) {
        new EmployeeInfo();
    }
}
```



Customer Info

```
package hotel.management.system;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import net.proteanit.sql.*;
```

```
public class CustomerInfo extends JFrame implements ActionListener {
```

```
    JTable table;
    JButton back;
```

```
    CustomerInfo()
    {
```

```
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
```

```
        JLabel l1 = new JLabel("Document Type");
        l1.setBounds(10, 10, 100, 20);
```

```
add(11);

JLabel 12 = new JLabel("Number");
12.setBounds(160, 10, 100, 20);
add(12);

JLabel 13 = new JLabel("Name");
13.setBounds(290, 10, 100, 20);
add(13);

JLabel 14 = new JLabel("Gender");
14.setBounds(410, 10, 100, 20);
add(14);

JLabel 15 = new JLabel("Country");
15.setBounds(540, 10, 100, 20);
add(15);

JLabel 16 = new JLabel("Room Num");
16.setBounds(640, 10, 100, 20);
add(16);

JLabel 17 = new JLabel("Check-in time");
17.setBounds(760, 10, 100, 20);
add(17);

JLabel 18 = new JLabel("Deposit");
18.setBounds(900, 10, 100, 20);
add(18);

table = new JTable();
table.setBounds(0, 40, 1000, 400);
add(table);

try
{
    conn c = new conn();
    ResultSet rs = c.s.executeQuery("select * from customer");
    table.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (Exception e)
{
    e.printStackTrace();
}

back = new JButton("Back");
back.setBounds(200, 500, 120, 30);
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);

back.addActionListener(this);

add(back);
```

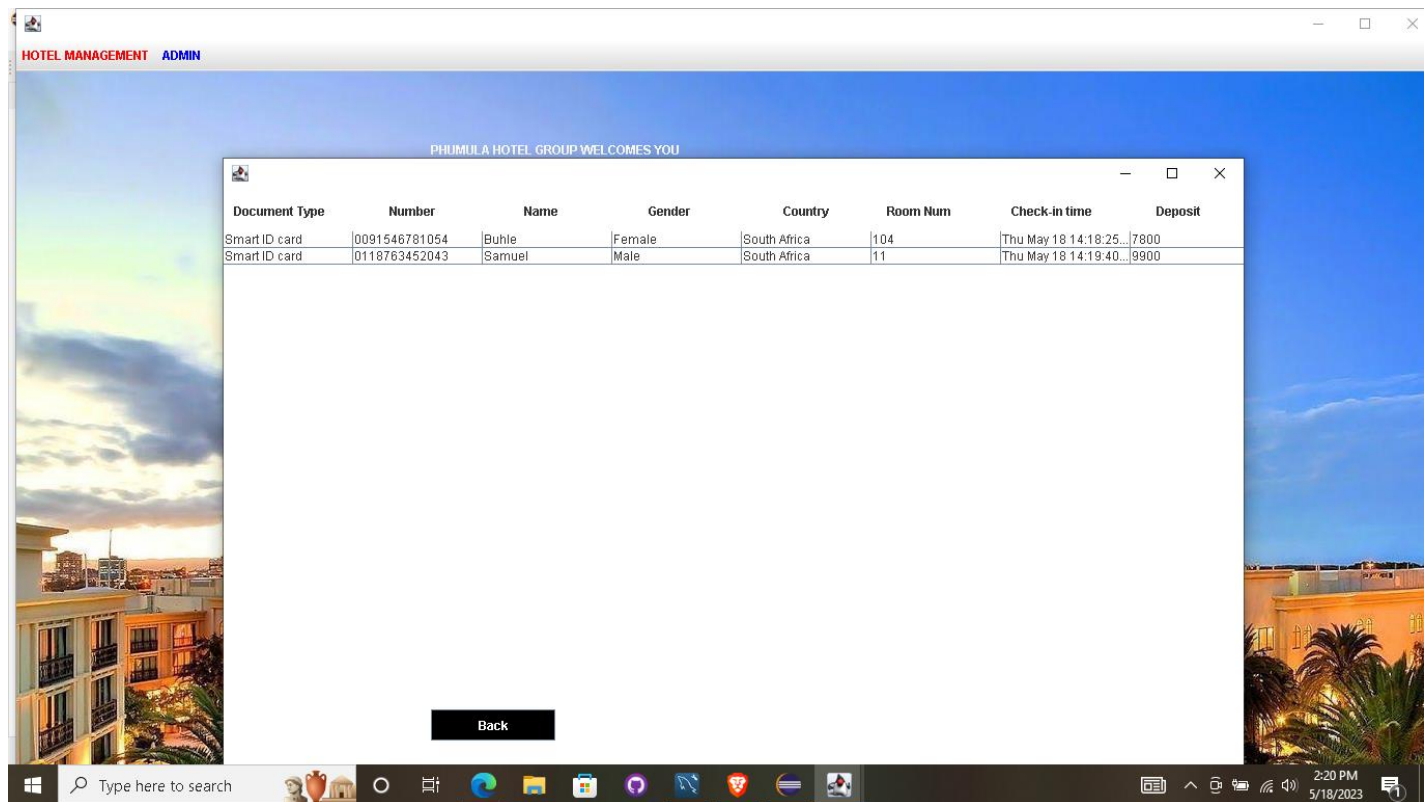
```

        setBounds(420, 200, 1000, 600);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        setVisible(false);
        new Reception();
    }

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

```



Manager Info

```
package hotel.management.system;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import net.proteanit.sql.*;

public class ManagerInfo extends JFrame implements ActionListener {

    JTable table;
    JButton back;

    ManagerInfo()
    {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel l1 = new JLabel("Name");
        l1.setBounds(40, 10, 100, 20);
        add(l1);

        JLabel l2 = new JLabel("Age");
        l2.setBounds(170, 10, 100, 20);
        add(l2);

        JLabel l3 = new JLabel("Gender");
        l3.setBounds(290, 10, 100, 20);
        add(l3);

        JLabel l4 = new JLabel("Job");
        l4.setBounds(400, 10, 100, 20);
        add(l4);

        JLabel l5 = new JLabel("Salary");
        l5.setBounds(540, 10, 100, 20);
        add(l5);

        JLabel l6 = new JLabel("Phone Num");
        l6.setBounds(640, 10, 100, 20);
        add(l6);

        JLabel l7 = new JLabel("Email");
        l7.setBounds(790, 10, 100, 20);
        add(l7);

        JLabel l8 = new JLabel("ID Num");
        l8.setBounds(910, 10, 100, 20);
        add(l8);

        table = new JTable();
        table.setBounds(0, 40, 1000, 400);
        add(table);
```

```
        try
        {
            conn c = new conn();
            ResultSet rs = c.s.executeQuery("select * from employee where job = 'Manager'");
            table.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        back = new JButton("Back");
        back.setBounds(200, 500, 120, 30);
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);

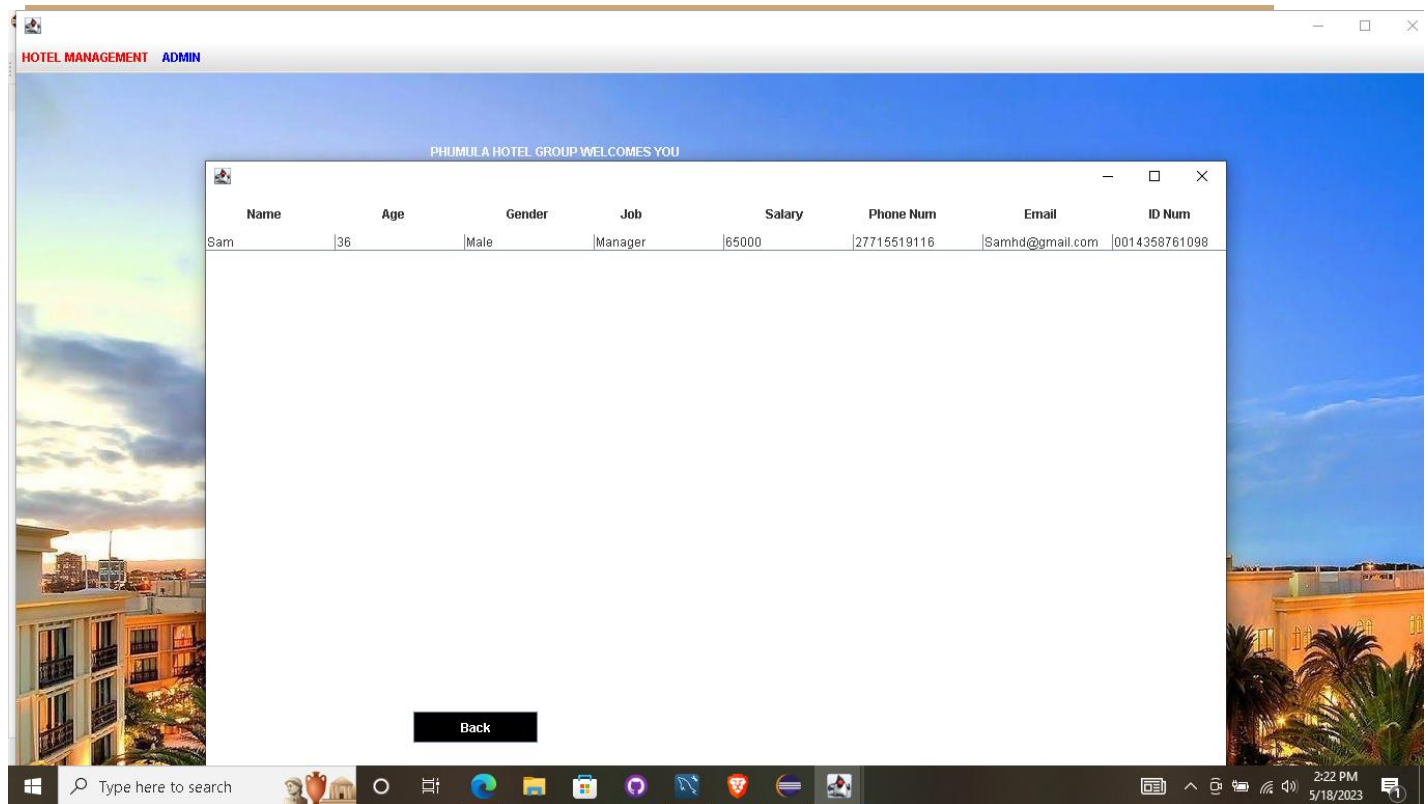
        back.addActionListener(this);

        add(back);

        setBounds(420, 200, 1000, 600);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        setVisible(false);
        new Reception();
    }

    public static void main(String[] args) {
        new ManagerInfo();
    }
}
```



CheckOut

```
package hotel.management.system;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.Date;
```

```
public class Checkout extends JFrame implements ActionListener {
```

```
    Choice ccustomer;
    JLabel lblroomnumber, lblcheckintime, lblcheckouttime;
    JButton checkout, back;
```

```
    Checkout()
    {
```

```
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
```

```
        JLabel text = new JLabel("Checkout");
        text.setBounds(100, 20, 100, 30);
```

```
text.setForeground(Color.BLUE);
text.setFont(new Font("Tahoma", Font.PLAIN, 20));
add(text);
```

```
JLabel lblid = new JLabel("Customer ID");
lblid.setBounds(30, 80, 100, 30);
add(lblid);
```

```
ccustomer = new Choice();
ccustomer.setBounds(150, 80, 150, 25);
add(ccustomer);
```

```
ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\tick.png");
```

```
Image i2 = i1.getImage().getScaledInstance(20, 20, Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel tick = new JLabel(i3);
tick.setBounds(310, 80, 20, 20);
add(tick);
```

```
JLabel lblroom = new JLabel("Room Number");
lblroom.setBounds(30, 130, 100, 30);
add(lblroom);
```

```
lblroomnumber = new JLabel();
lblroomnumber.setBounds(150, 130, 100, 30);
add(lblroomnumber);
```

```
JLabel lblcheckin = new JLabel("Check-in Time");
lblcheckin.setBounds(30, 180, 100, 30);
add(lblcheckin);
```

```
lblcheckintime = new JLabel();
lblcheckintime.setBounds(150, 180, 100, 30);
add(lblcheckintime);
```

```
JLabel lblcheckout = new JLabel("Check-out Time");
lblcheckout.setBounds(30, 230, 100, 30);
add(lblcheckout);
```

```
Date date = new Date();
lblcheckouttime = new JLabel("" + date);
lblcheckouttime.setBounds(150, 230, 150, 30);
add(lblcheckouttime);
```

```
checkout = new JButton("Checkout");
checkout.setBackground(Color.BLACK);
checkout.setForeground(Color.WHITE);
checkout.setBounds(30, 280, 120, 30);
add(checkout);
```

```
checkout.addActionListener(this);
```

```
back = new JButton("Back");
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);
back.setBounds(170, 280, 120, 30);
add(back);

back.addActionListener(this);

try
{
    conn c = new conn();
    ResultSet rs = c.s.executeQuery("select * from customer");

    while (rs.next())
    {
        ccustomer.add(rs.getString("number"));
        lblroomnumber.setText(rs.getString("room"));
        lblcheckintime.setText(rs.getString("checkintime"));
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
```

```
        ImageIcon i4 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\sixth.jpg");
        Image i5 = i4.getImage().getScaledInstance(400, 250, Image.SCALE_DEFAULT);
        ImageIcon i6 = new ImageIcon(i5);
        JLabel image = new JLabel(i6);
        image.setBounds(350, 50, 400, 250);
        add(image);

        setBounds(300, 200, 800, 400);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        if (ae.getSource() == checkout)
        {
            String query1 = "delete from customer where number = '"+ccustomer.getSelectedItem()+"'";
            String query2 = "update room set availability = 'Available' where roomnumber =
 '"+lblroomnumber.getText()+"'";

            try
            {
                conn c = new conn();
```

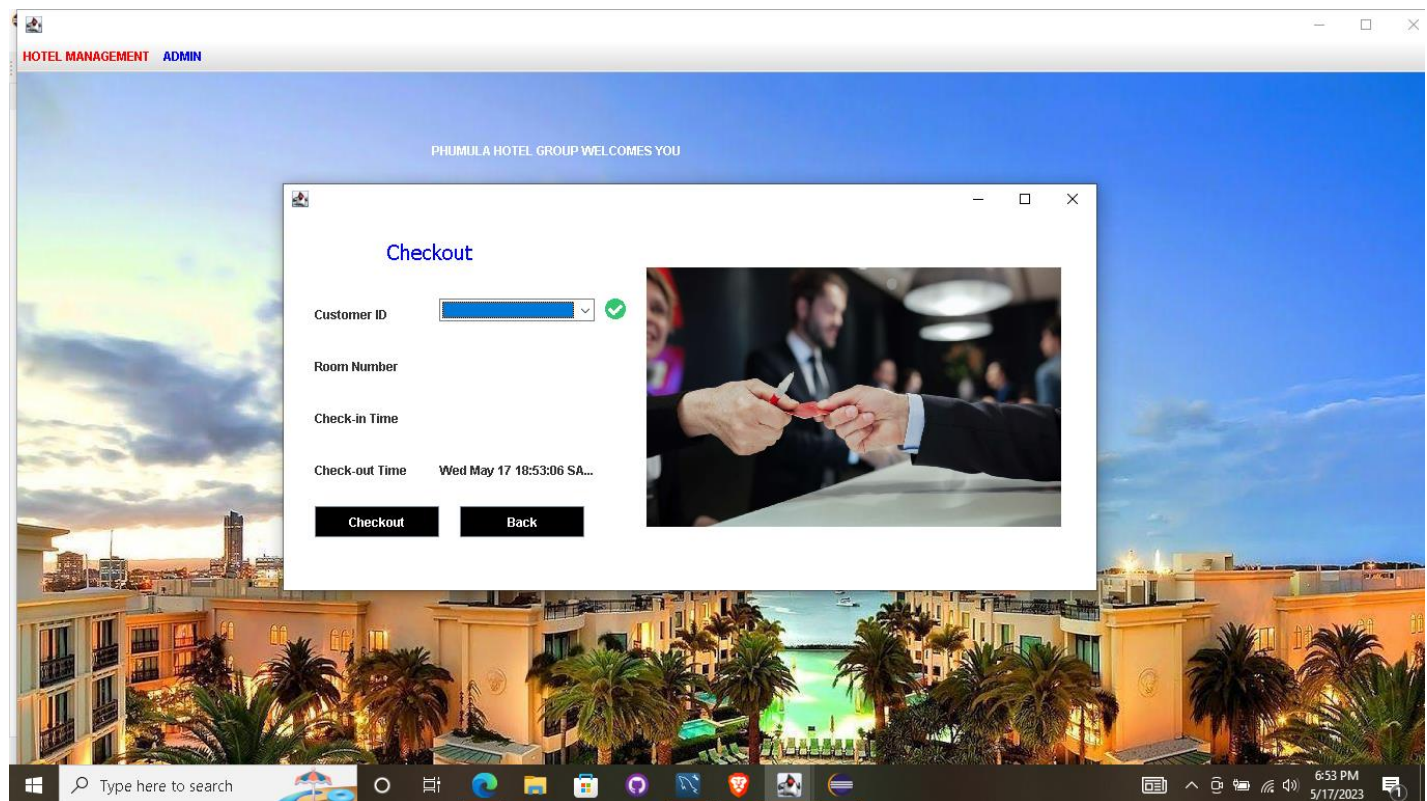
```

        c.s.executeUpdate(query1);
        c.s.executeUpdate(query2);

        JOptionPane.showMessageDialog(null, "Checkout Done");
        setVisible(false);
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
else
{
    setVisible(false);
    new Reception();
}
}

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

```



Update Check Status

```
package hotel.management.system;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*.*;
import java.sql.*.*;

public class UpdateCheck extends JFrame implements ActionListener {

    Choice ccustomer;
    JTextField tfroom, tfname, tfcheckin, tfpaid, tfpending;
    JButton check, update, back;

    UpdateCheck()
    {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel text = new JLabel("Update Room Status");
        text.setFont(new Font("Tahoma", Font.PLAIN, 20));
        text.setBounds(90, 20, 200, 30);
        text.setForeground(Color.BLUE);
        add(text);

        JLabel lblid = new JLabel("Customer ID");
        lblid.setBounds(30, 80, 100, 20);
        add(lblid);

        ccustomer = new Choice();
        ccustomer.setBounds(200, 80, 150, 25);
        add(ccustomer);

        try
        {
            conn c = new conn();
            ResultSet rs = c.s.executeQuery("select * from customer");

            while (rs.next())
            {
                ccustomer.add(rs.getString("number"));
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

```
}

JLabel lblroom = new JLabel("Room Number");
lblroom.setBounds(30, 120, 100, 20);
add(lblroom);

tfroom = new JTextField();
tfroom.setBounds(200, 120, 150, 25);
add(tfroom);

JLabel lblname = new JLabel("Name");
lblname.setBounds(30, 160, 100, 20);
add(lblname);

tfname = new JTextField();
tfname.setBounds(200, 160, 150, 25);
add(tfname);

JLabel lblcheckin = new JLabel("Check-in Time");
lblcheckin.setBounds(30, 200, 100, 20);
add(lblcheckin);

tfcheckin = new JTextField();
tfcheckin.setBounds(200, 200, 150, 25);
add(tfcheckin);

JLabel lblpaid = new JLabel("Amount Paid");
lblpaid.setBounds(30, 240, 100, 20);
add(lblpaid);

tfpaid = new JTextField();
tfpaid.setBounds(200, 240, 150, 25);
add(tfpaid);

JLabel lblpending = new JLabel("Pending Amount");
lblpending.setBounds(30, 280, 100, 20);
add(lblpending);

tfpending = new JTextField();
tfpending.setBounds(200, 280, 150, 25);
add(tfpending);

check = new JButton("Check");
check.setBackground(Color.BLACK);
check.setForeground(Color.WHITE);
check.setBounds(30, 340, 100, 30);
add(check);

check.addActionListener(this);

update = new JButton("Update");
update.setBackground(Color.BLACK);
update.setForeground(Color.WHITE);
```

```
        update.setBounds(150, 340, 100, 30);
        add(update);

        update.addActionListener(this);

        back = new JButton("Back");
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);
        back.setBounds(270, 340, 100, 30);
        add(back);

        back.addActionListener(this);

        ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\nine.jpg");
        JLabel image = new JLabel(i1);
        image.setBounds(400, 50, 500, 300);
        add(image);

        setBounds(300, 200, 900, 500);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        if (ae.getSource() == check)
        {
            String id = ccustomer.getSelectedItemAt();
            String query = "select * from customer where number = '"+id+"'";

            try
            {
                conn c = new conn();
                ResultSet rs = c.s.executeQuery(query);

                while (rs.next())
                {
                    tfroom.setText(rs.getString("room"));
                    tfname.setText(rs.getString("name"));
                    tfcheckin.setText(rs.getString("checkintime"));
                    tfpaid.setText(rs.getString("deposit"));
                }

                ResultSet rs2 = c.s.executeQuery("select * from room where roomnumber =
 '"+tfroom.getText()+"'");

                while (rs2.next())
                {
                    String price = rs2.getString("price");
                    int amountPaid = Integer.parseInt(price) - Integer.parseInt(tfpaid.getText());
                    tfpending.setText("" + amountPaid);
                }
            }
        }
    }
}
```

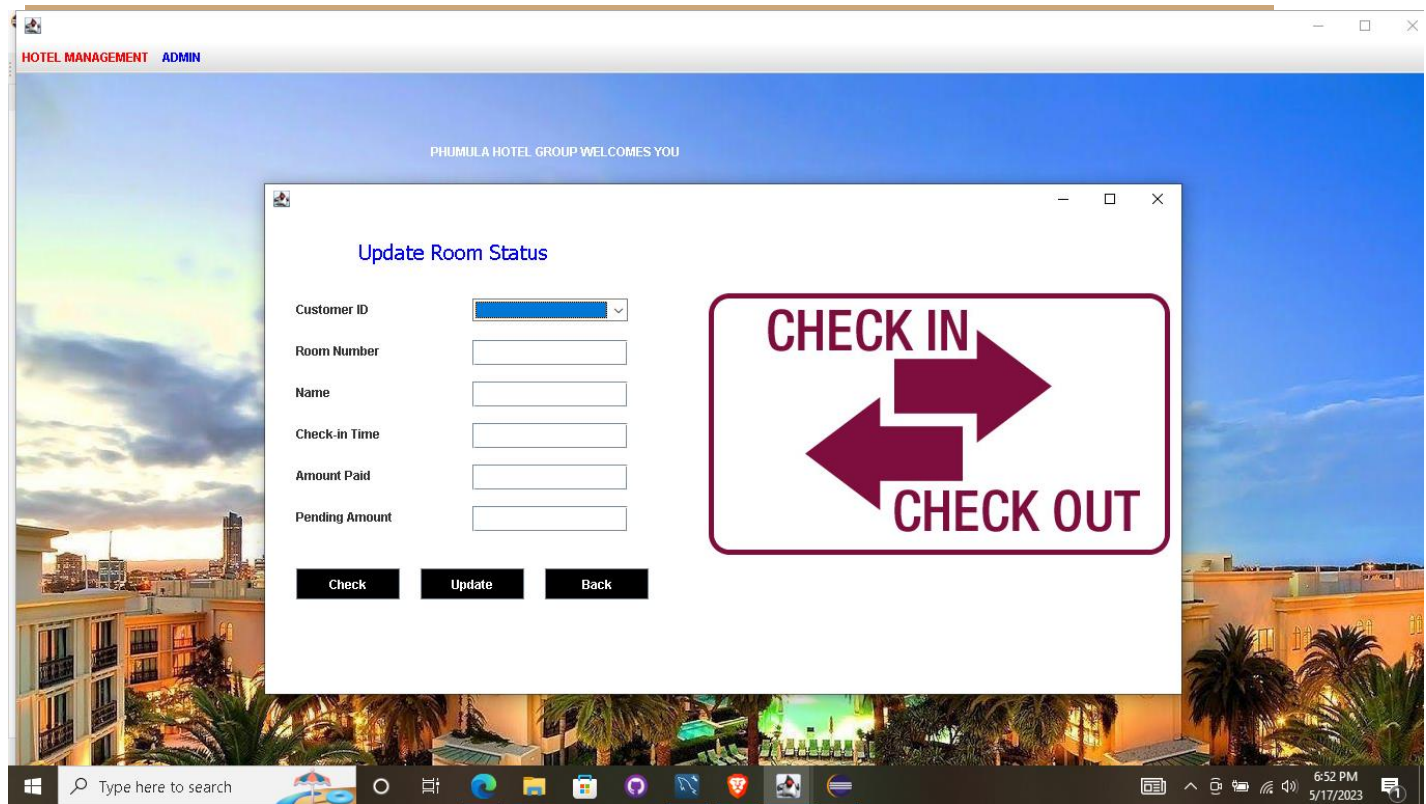
```
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    else if (ae.getSource() == update)
    {
        String number = ccustomer.getSelectedItemAt();
        String room = tfroom.getText();
        String name = tfname.getText();
        String checkin = tfcheckin.getText();
        String deposit = tfpaid.getText();

        try
        {
            conn c = new conn();
            c.s.executeUpdate("update customer set room = '"+room+"', name = '"+name+"',
checkintime = '"+checkin+"', deposit = '"+deposit+"' where number = '"+number+"'");

            JOptionPane.showMessageDialog(null, "Data Updated Successfully");

            setVisible(false);
            new Reception();
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    else
    {
        setVisible(false);
        new Reception();
    }
}

public static void main(String[] args) {
    new UpdateCheck();
}
}
```



Update Room Status

```
package hotel.management.system;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
```

```
public class UpdateRoom extends JFrame implements ActionListener {
```

```
    Choice ccustomer;
    JTextField tfroom, tfavailable, tfstatus, tfpaid, tfpending;
    JButton check, update, back;
```

```
    UpdateRoom()
    {
```

```
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
```

```
        JLabel text = new JLabel("Update Room Status");
        text.setFont(new Font("Tahoma", Font.PLAIN, 20));
```

```
text.setBounds(30, 20, 250, 30);
text.setForeground(Color.BLUE);
add(text);

JLabel lblid = new JLabel("Customer ID");
lblid.setBounds(30, 80, 100, 20);
add(lblid);

ccustomer = new Choice();
ccustomer.setBounds(200, 80, 150, 25);
add(ccustomer);

try
{
    conn c = new conn();
    ResultSet rs = c.s.executeQuery("select * from customer");

    while (rs.next())
    {
        ccustomer.add(rs.getString("number"));
    }
}
catch (Exception e)
{
    e.printStackTrace();
}

JLabel lblroom = new JLabel("Room Number");
lblroom.setBounds(30, 130, 100, 20);
add(lblroom);

tfroom = new JTextField();
tfroom.setBounds(200, 130, 150, 25);
add(tfroom);

JLabel lblavailable = new JLabel("Availability");
lblavailable.setBounds(30, 180, 100, 20);
add(lblavailable);

tfavailable = new JTextField();
tfavailable.setBounds(200, 180, 150, 25);
add(tfavailable);

JLabel lblstatus = new JLabel("Cleaning Status");
lblstatus.setBounds(30, 230, 100, 20);
add(lblstatus);

tfstatus = new JTextField();
tfstatus.setBounds(200, 230, 150, 25);
add(tfstatus);
```

```
        check = new JButton("Check");
        check.setBackground(Color.BLACK);
        check.setForeground(Color.WHITE);
        check.setBounds(30, 300, 100, 30);
        add(check);

        check.addActionListener(this);

        update = new JButton("Update");
        update.setBackground(Color.BLACK);
        update.setForeground(Color.WHITE);
        update.setBounds(150, 300, 100, 30);
        add(update);

        update.addActionListener(this);

        back = new JButton("Back");
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);
        back.setBounds(270, 300, 100, 30);
        add(back);

        back.addActionListener(this);

        ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\seventh.jpg");
        Image i2 = i1.getImage().getScaledInstance(500, 300, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(400, 50, 500, 300);
        add(image);

        setBounds(300, 200, 900, 450);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        if (ae.getSource() == check)
        {
            String id = ccustomer.getSelectedItemAt();
            String query = "select * from customer where number = '"+id+"'";

            try
            {
                conn c = new conn();
                ResultSet rs = c.s.executeQuery(query);

                while (rs.next())
                {
                    tfroom.setText(rs.getString("room"));
                }
            }
        }
    }
}
```

```
        ResultSet rs2 = c.s.executeQuery("select * from room where roomnumber =
"+tfroom.getText()+"");

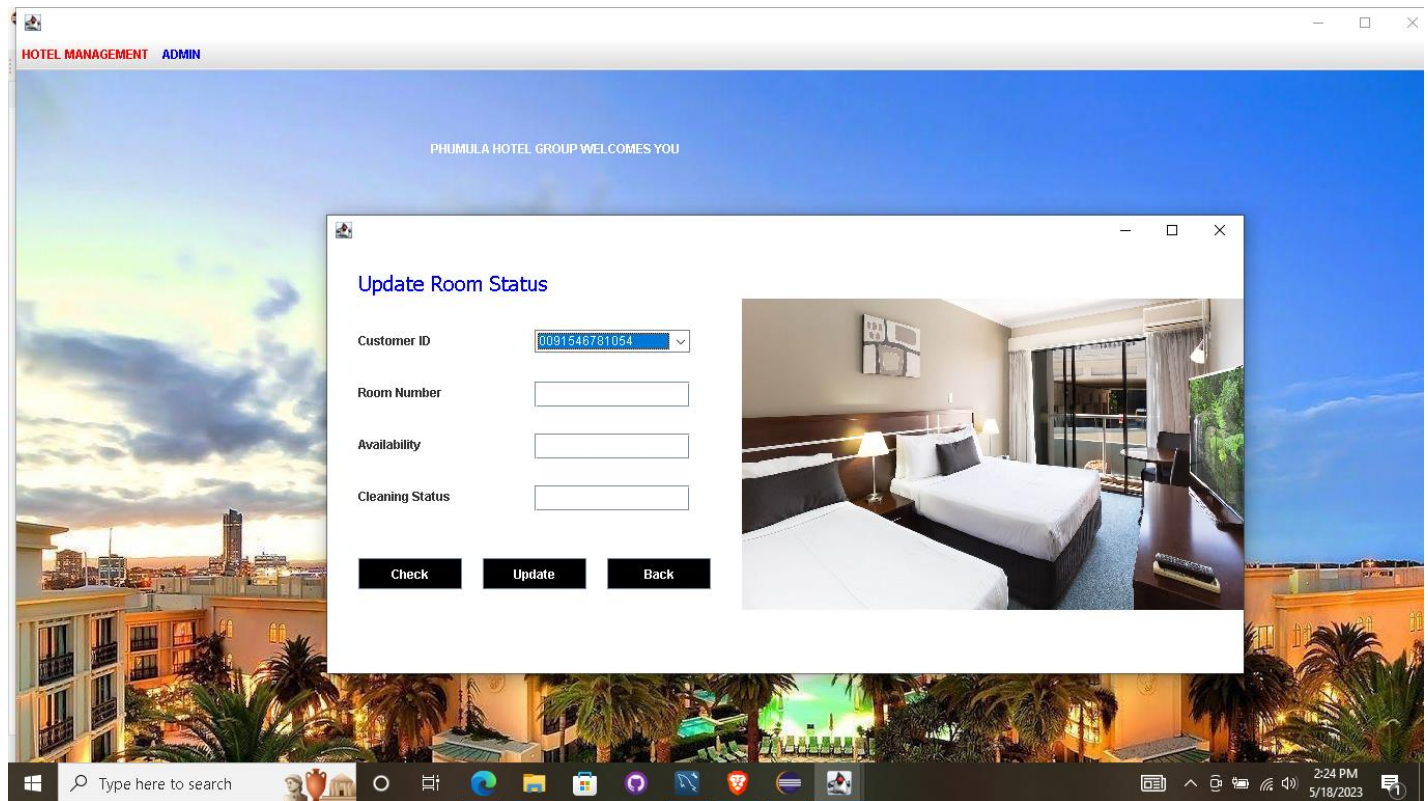
        while (rs2.next())
        {
            tfavailable.setText(rs2.getString("availability"));
            tfstatus.setText(rs2.getString("cleaning_status"));
        }
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
else if (ae.getSource() == update)
{
    String number = ccustomer.getSelectedItemAt();
    String room = tfroom.getText();
    String available = tfavailable.getText();
    String status = tfstatus.getText();

    try
    {
        conn c = new conn();
        c.s.executeUpdate("update room set availability = "+available+", cleaning_status =
"+status+" where roomnumber = "+room+"");

        JOptionPane.showMessageDialog(null, "Data Updated Successfully");

        setVisible(false);
        new Reception();
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
else
{
    setVisible(false);
    new Reception();
}
}

public static void main(String[] args) {
    new UpdateRoom();
}
}
```



Pick Up Service

```
package hotel.management.system;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import net.proteanit.sql.*;
```

```
public class Pickup extends JFrame implements ActionListener {
```

```
    JTable table;
    JButton submit, back;
    Choice typeofcar;
```

```
    Pickup()
    {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
```

```
JLabel text = new JLabel("Pickup Service");
text.setFont(new Font("Tahoma", Font.PLAIN, 20));
text.setBounds(400, 30, 200, 30);
add(text);
```

```
JLabel lblbed = new JLabel("Type Of Car");
lblbed.setBounds(50, 100, 100, 20);
add(lblbed);
```

```
typeofcar = new Choice();
typeofcar.setBounds(150, 100, 200, 25);
add(typeofcar);
```

```
try
{
    conn c = new conn();
    ResultSet rs = c.s.executeQuery("select * from driver");
    while (rs.next())
    {
        typeofcar.add(rs.getString("brand"));
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
```

```
JLabel l1 = new JLabel("Name");
l1.setBounds(30, 160, 100, 20);
add(l1);
```

```
JLabel l2 = new JLabel("Age");
l2.setBounds(210, 160, 100, 20);
add(l2);
```

```
JLabel l3 = new JLabel("Gender");
l3.setBounds(330, 160, 100, 20);
add(l3);
```

```
JLabel l4 = new JLabel("Company");
l4.setBounds(460, 160, 100, 20);
add(l4);
```

```
JLabel l5 = new JLabel("Brand");
l5.setBounds(630, 160, 100, 20);
add(l5);
```

```
JLabel l6 = new JLabel("Availability");
l6.setBounds(740, 160, 100, 20);
add(l6);
```

```
JLabel l7 = new JLabel("Location");
l7.setBounds(890, 160, 100, 20);
add(l7);

table = new JTable();
table.setBounds(0, 200, 1000, 300);
add(table);

try
{
    conn c = new conn();
    ResultSet rs = c.s.executeQuery("select * from driver");
    table.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (Exception e)
{
    e.printStackTrace();
}

submit = new JButton("Submit");
submit.setBounds(300, 520, 120, 30);
submit.setBackground(Color.BLACK);
submit.setForeground(Color.WHITE);

submit.addActionListener(this);

add(submit);

back = new JButton("Back");
back.setBounds(500, 520, 120, 30);
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);

back.addActionListener(this);

add(back);

setBounds(300, 200, 1000, 600);
setVisible(true);
}

public void actionPerformed(ActionEvent ae)
{
    if (ae.getSource() == submit)
    {
        try
        {
            String query = "select * from driver where brand = '"+typeofcar.getSelectedItemAt()+"'";

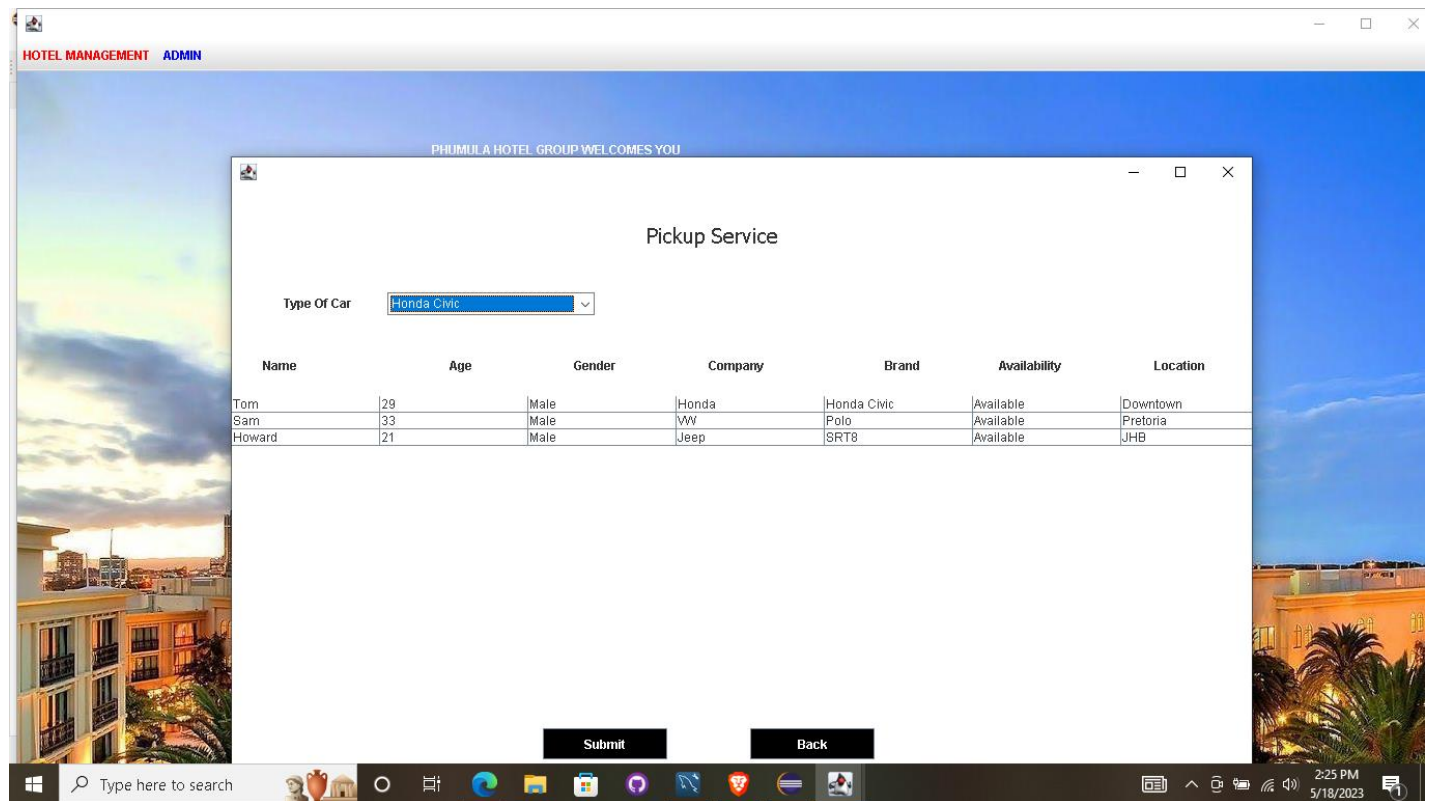
            conn conn = new conn();
            ResultSet rs;
            rs = conn.s.executeQuery(query);
            table.setModel(DbUtils.resultSetToTableModel(rs));
```

```

    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
else
{
    setVisible(false);
    new Reception();
}
}

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

```



Search Room

```

package hotel.management.system;

import javax.swing.*;

```

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import net.proteanit.sql.*;
```

```
public class SearchRoom extends JFrame implements ActionListener {
```

```
    JTable table;
    JButton submit, back;
    JComboBox bedType;
    JCheckBox available;
```

```
    SearchRoom()
```

```
    {
```

```
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
```

```
        JLabel text = new JLabel("Search for Room");
        text.setFont(new Font("Tahoma", Font.PLAIN, 20));
        text.setBounds(400, 30, 200, 30);
        add(text);
```

```
        JLabel lblbed = new JLabel("Bed Type");
        lblbed.setBounds(50, 100, 100, 20);
        add(lblbed);
```

```
        bedType = new JComboBox(new String[] { "Single bed", "Double bed", });
        bedType.setBounds(150, 100, 150, 25);
        bedType.setBackground(Color.WHITE);
        add(bedType);
```

```
        available = new JCheckBox("Only Dispalv Available");
        available.setBounds(650, 100, 150, 25);
        available.setBackground(Color.WHITE);
        add(available);
```

```
        JLabel l1 = new JLabel("Room Number");
        l1.setBounds(50, 160, 100, 20);
        add(l1);
```

```
        JLabel l2 = new JLabel("Availability");
        l2.setBounds(270, 160, 100, 20);
        add(l2);
```

```
        JLabel l3 = new JLabel("Cleaning Status");
        l3.setBounds(450, 160, 100, 20);
        add(l3);
```

```
        JLabel l4 = new JLabel("Price");
        l4.setBounds(670, 160, 100, 20);
        add(l4);
```

```
        JLabel l5 = new JLabel("Bed Type");
```

```

15.setBounds(870, 160, 100, 20);
add(15);

table = new JTable();
table.setBounds(0, 200, 1000, 300);
add(table);

try
{
    conn c = new conn();
    ResultSet rs = c.s.executeQuery("select * from room");
    table.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (Exception e)
{
    e.printStackTrace();
}

submit = new JButton("Submit");
submit.setBounds(300, 520, 120, 30);
submit.setBackground(Color.BLACK);
submit.setForeground(Color.WHITE);

submit.addActionListener(this);

add(submit);

back = new JButton("Back");
back.setBounds(500, 520, 120, 30);
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);

back.addActionListener(this);

add(back);

setBounds(300, 200, 1000, 600);
setVisible(true);
}

public void actionPerformed(ActionEvent ae)
{
    if (ae.getSource() == submit)
    {
        try
        {
            String query1 = "select * from room where bed_type =
"+bedType.getSelectedItemAt(0)+"";
            String query2 = "select * from room where availability = 'Available' AND bed_type =
"+bedType.getSelectedItemAt(0)+"";

            conn conn = new conn();
            ResultSet rs;

```

```
        if (available.isSelected())
        {
            rs = conn.s.executeQuery(query2);
        }
        else
        {
            rs = conn.s.executeQuery(query1);
        }

        table.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
else
{
    setVisible(false);
    new Reception();
}
}

public static void main(String[] args) {
    new SearchRoom();
}
}
```



Search For Room

Room Bed Type:

☒ Only display Available

Room Number	Availability	Clean Status	Price	Bed Type
106	Available	Cleaned	12000	Double Bed
109	Available	Cleaned	12000	Double Bed
110	Available	Cleaned	12000	Double Bed
111	Available	Dirty	26000	Double Bed
115	Available	Cleaned	9000	Double Bed
116	Available	Cleaned	9600	Double Bed
120	Available	Cleaned	12000	Double Bed

Search

Add Employee

```
package hotel.management.system;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class AddEmployee extends JFrame implements ActionListener {

    JTextField tfname, tfemail, tfphone, tfage, tfsalary, tfIDnum;
    JRadioButton rbmale, rbfemale;
    JButton submit;
    JComboBox cbjob;

    AddEmployee()
    {
        setLayout(null);

        JLabel lblname = new JLabel("NAME");
        lblname.setBounds(60, 30, 120, 30);
```

```
lblname.setFont(new Font("Tahoma", Font.PLAIN, 17));
add(lblname);
```

```
tfname = new JTextField();
tfname.setBounds(200, 30, 150, 30);
add(tfname);
```

```
JLabel lblage = new JLabel("AGE");
lblage.setBounds(60, 80, 120, 30);
lblage.setFont(new Font("Tahoma", Font.PLAIN, 17));
add(lblage);
```

```
tfage = new JTextField();
tfage.setBounds(200, 80, 150, 30);
add(tfage);
```

```
JLabel lblgender = new JLabel("GENDER");
lblgender.setBounds(60, 130, 120, 30);
lblgender.setFont(new Font("Tahoma", Font.PLAIN, 17));
add(lblgender);
```

```
rbmale = new JRadioButton("Male");
rbmale.setBounds(200, 130, 70, 30);
rbmale.setFont(new Font("Tahoma", Font.PLAIN, 14));
rbmale.setBackground(Color.WHITE);
add(rbmale);
```

```
rbfemale = new JRadioButton("Female");
rbfemale.setBounds(280, 130, 70, 30);
rbfemale.setFont(new Font("Tahoma", Font.PLAIN, 14));
rbfemale.setBackground(Color.WHITE);
add(rbfemale);
```

```
ButtonGroup gr = new ButtonGroup();
gr.add(rbmale);
gr.add(rbfemale);
```

```
JLabel lbljob = new JLabel("JOB");
lbljob.setBounds(60, 180, 120, 30);
lbljob.setFont(new Font("Tahoma", Font.PLAIN, 17));
add(lbljob);
```

```
String str[] = { "Front Desk Clerks", "Porters", "HouseKeeping", "Kitchen Staff", "Room Service",
"Chefs",
                "Waiter/Waitress", "Manager", "Accountant" };
```

```
cbjob = new JComboBox(str);
cbjob.setBounds(200, 180, 150, 30);
cbjob.setBackground(Color.WHITE);
add(cbjob);
```

```
JLabel lblsalary = new JLabel("SALARY");
lblsalary.setBounds(60, 230, 120, 30);
```

```
lblsalary.setFont(new Font("Tahoma", Font.PLAIN, 17));
add(lblsalary);

tfsalary = new JTextField();
tfsalary.setBounds(200, 230, 150, 30);
add(tfsalary);

JLabel lblphone = new JLabel("PHONE");
lblphone.setBounds(60, 280, 120, 30);
lblphone.setFont(new Font("Tahoma", Font.PLAIN, 17));
add(lblphone);

tfphone = new JTextField();
tfphone.setBounds(200, 280, 150, 30);
add(tfphone);

JLabel lblemail = new JLabel("EMAIL");
lblemail.setBounds(60, 330, 120, 30);
lblemail.setFont(new Font("Tahoma", Font.PLAIN, 17));
add(lblemail);

tfemail = new JTextField();
tfemail.setBounds(200, 330, 150, 30);
add(tfemail);

JLabel lblIDnum = new JLabel("ID NUMBER");
lblIDnum.setBounds(60, 380, 120, 30);
lblIDnum.setFont(new Font("Tahoma", Font.PLAIN, 17));
add(lblIDnum);

tfIDnum = new JTextField();
tfIDnum.setBounds(200, 380, 150, 30);
add(tfIDnum);

submit = new JButton("SUBMIT");
submit.setBackground(Color.BLACK);
submit.setForeground(Color.WHITE);
submit.setBounds(200, 430, 150, 30);

submit.addActionListener(this);

add(submit);

ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\tenth.jpg");
Image i2 = i1.getImage().getScaledInstance(450, 450, Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel image = new JLabel(i3);
image.setBounds(380, 60, 450, 370);
add(image);

getContentPane().setBackground(Color.WHITE);
```

```
        setBounds(350, 200, 850, 540);
        setVisible(true);

    }

    public void actionPerformed(ActionEvent ae)
    {
        String name = tfname.getText();
        String age = tfage.getText();
        String salary = tfsalary.getText();
        String phone = tfphone.getText();
        String email = tfemail.getText();
        String idNum = tfIDnum.getText();

        String gender = null;
        //email.equals("") && contains("@") && contains(".com)
        if (name.equals(""))
        {
            JOptionPane.showMessageDialog(null, "Name should not be empty");
            return;
        }

        if (rbmale.isSelected())
        {
            gender = "Male";
        }

        else if (rbfemale.isSelected())
        {
            gender = "Female";
        }

        String job = (String)cjob.getSelectedItemAt();

        try
        {
            conn conn = new conn();
            String query = "insert into employee(name, age, gender, job, salary, phone, email, idNum) values('"+name+"',
            '"+age+"', '"+gender+"', '"+job+"', '"+salary+"', '"+phone+"', '"+email+"', '"+idNum+"')";
            conn.s.executeUpdate(query);

            JOptionPane.showMessageDialog(null, "Employee added successfully");

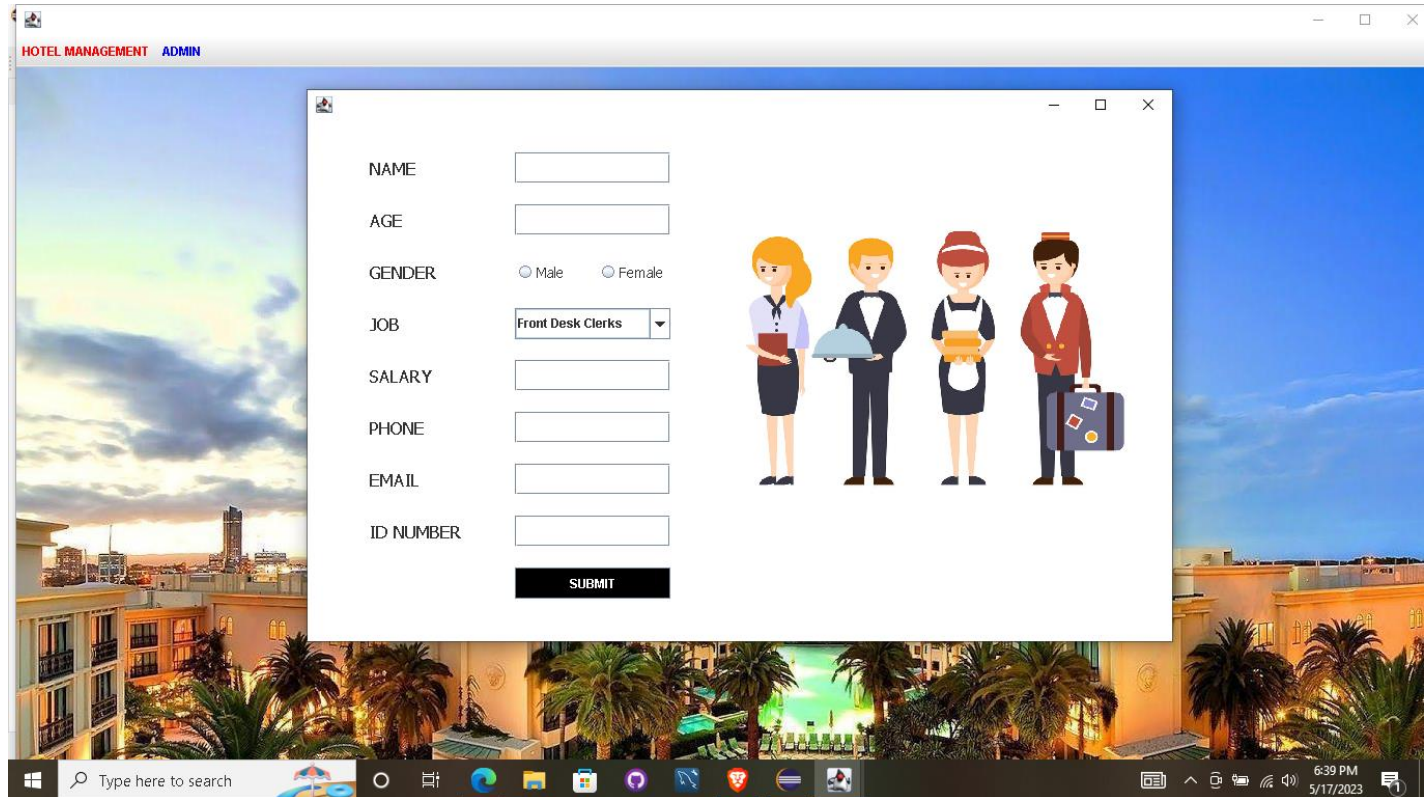
            setVisible(false);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }

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

```

        new AddEmployee();
    }
}

```



Add Rooms

```
package hotel.management.system;
```

```

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;

```

```

public class AddRoom extends JFrame implements ActionListener
{

```

```
JButton add, cancel;
JTextField tfroom, tfprice;
JComboBox typecombo, availablecombo, cleancombo;

AddRoom()
{
    getContentPane().setBackground(Color.WHITE);
    setLayout(null);

    JLabel heading = new JLabel("Add Rooms");
    heading.setFont(new Font("Tahoma", Font.BOLD, 15));
    heading.setBounds(150, 20, 200, 20);
    add(heading);

    JLabel lblroomno = new JLabel("Room Number");
    lblroomno.setFont(new Font("Tahoma", Font.BOLD, 15));
    lblroomno.setBounds(60, 80, 120, 30);
    add(lblroomno);

    tfroom = new JTextField();
    tfroom.setBounds(200, 80, 150, 30);
    add(tfroom);

    JLabel lblavailable = new JLabel("Available");
    lblavailable.setFont(new Font("Tahoma", Font.BOLD, 15));
    lblavailable.setBounds(60, 130, 120, 30);
    add(lblavailable);

    String availableOptions[] = {"Available", "Occupied"};

    availablecombo = new JComboBox(availableOptions);
    availablecombo.setBounds(200, 130, 150, 30);
    availablecombo.setBackground(Color.WHITE);
    add(availablecombo);

    JLabel lblclean = new JLabel("Cleaning Status");
    lblclean.setFont(new Font("Tahoma", Font.BOLD, 15));
    lblclean.setBounds(60, 180, 120, 30);
    add(lblclean);

    String cleanOptions[] = {"Cleaned", "Dirty"};

    cleancombo = new JComboBox(cleanOptions);
    cleancombo.setBounds(200, 180, 150, 30);
    cleancombo.setBackground(Color.WHITE);
    add(cleancombo);

    JLabel lblprice = new JLabel("Price");
    lblprice.setFont(new Font("Tahoma", Font.BOLD, 15));
    lblprice.setBounds(60, 230, 120, 30);
    add(lblprice);

    tfprice = new JTextField();
```

```
tfprice.setBounds(200, 230, 150, 30);
add(tfprice);

JLabel lbltype = new JLabel("Bed Type");
lbltype.setFont(new Font("Tahoma", Font.BOLD, 15));
lbltype.setBounds(60, 280, 120, 30);
add(lbltype);

String typeOptions[] = {"Single Bed", "Double Bed"};

typecombo = new JComboBox(typeOptions);
typecombo.setBounds(200, 280, 150, 30);
typecombo.setBackground(Color.WHITE);
add(typecombo);

add = new JButton("Add Room");
add.setForeground(Color.WHITE);
add.setBackground(Color.BLACK);
add.setBounds(60, 350, 130, 30);

add.addActionListener(this);

add(add);

cancel = new JButton("Cancel");
cancel.setForeground(Color.WHITE);
cancel.setBackground(Color.BLACK);
cancel.setBounds(220, 350, 130, 30);

cancel.addActionListener(this);

add(cancel);

ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\twelve.jpg");
JLabel image = new JLabel(i1);
image.setBounds(400, 30, 500, 300);
add(image);

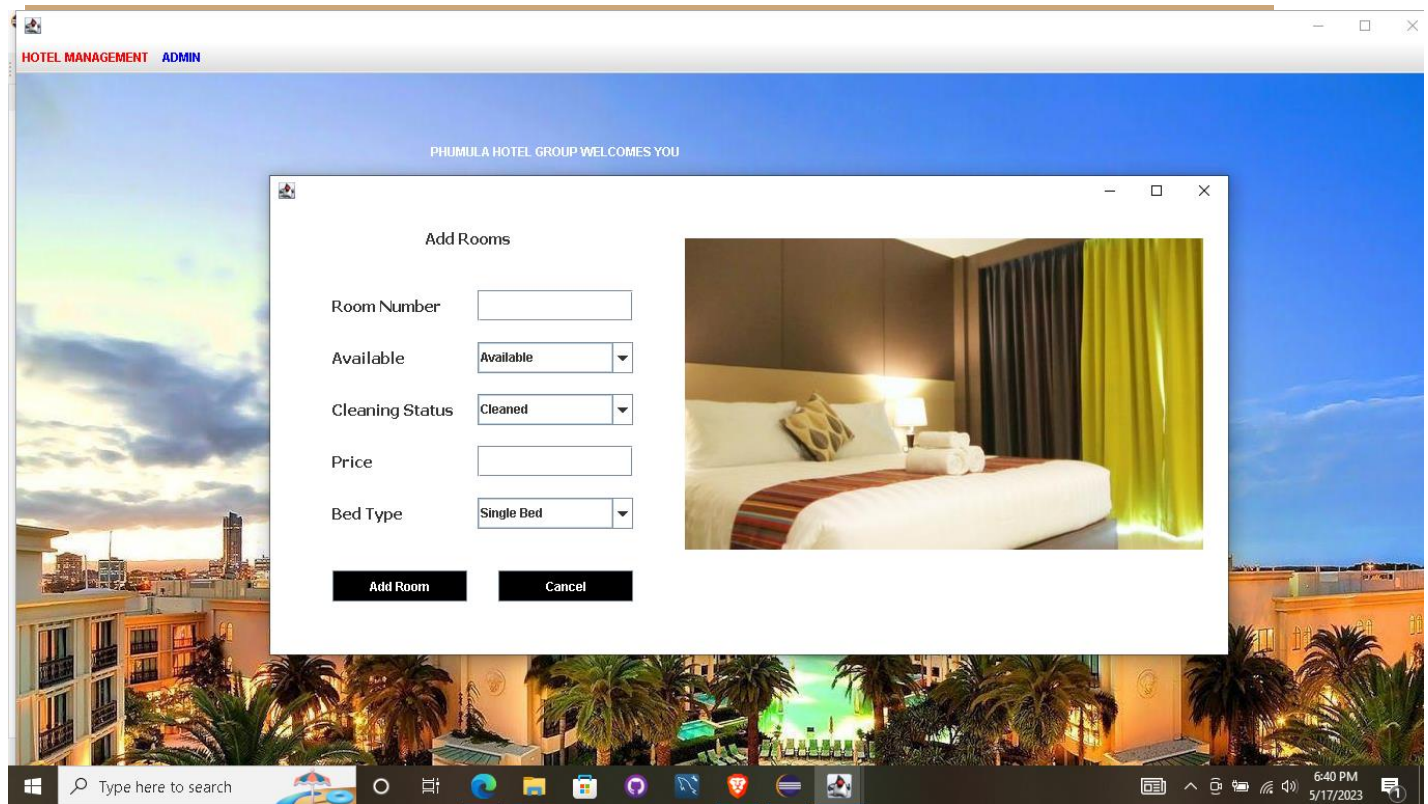
setBounds(330, 280, 940, 470);
setVisible(true);
}

public void actionPerformed(ActionEvent ae)
{
    if (ae.getSource() == add)
    {
        String roomnumber = tfroom.getText();
        String availability = (String) availablecombo.getSelectedItem();
        String status = (String) cleancombo.getSelectedItem();
        String price = tfprice.getText();
        String type = (String) typecombo.getSelectedItem();
```

```
        try
        {
            conn conn = new conn();
            String str = "insert into room(roomnumber, availability, cleaning_status, price, bed_type)
values('" + roomnumber + "', '" + availability + "', '" + status + "', '" + price + "', '" + type + "')";
            conn.s.executeUpdate(str);

            JOptionPane.showMessageDialog(null, "New Room Added Successfully");
            setVisible(false);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    else
    {
        setVisible(false);
    }
}

public static void main(String[] args)
{
    new AddRoom();
}
}
```



Add Drivers

```
package hotel.management.system;
```

```
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;
```

```
public class AddDrivers extends JFrame implements ActionListener
{
    JButton add, cancel;
    JTextField tfname, tfcompany, tfmodel, tfage, tflocation;
    JComboBox typecombo, availablecombo, agecombo, gendercombo;
```

```
    AddDrivers()
    {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
```

```
JLabel heading = new JLabel("Add Drivers");
heading.setFont(new Font("Tahoma", Font.BOLD, 15));
heading.setBounds(150, 10, 200, 20);
add(heading);
```

```
JLabel lblname = new JLabel("Name");
lblname.setFont(new Font("Tahoma", Font.BOLD, 15));
lblname.setBounds(60, 60, 120, 30);
add(lblname);
```

```
tfname = new JTextField();
tfname.setBounds(200, 70, 150, 30);
add(tfname);
```

```
JLabel lblage = new JLabel("Age");
lblage.setFont(new Font("Tahoma", Font.BOLD, 15));
lblage.setBounds(60, 110, 120, 30);
add(lblage);
```

```
tfage = new JTextField();
tfage.setBounds(200, 110, 150, 30);
add(tfage);
```

```
JLabel lblclean = new JLabel("Gender");
lblclean.setFont(new Font("Tahoma", Font.BOLD, 15));
lblclean.setBounds(60, 150, 120, 30);
add(lblclean);
```

```
String genderOptions[] = {"Male", "Female"};
gendercombo = new JComboBox(genderOptions);
gendercombo.setBounds(200, 150, 150, 30);
gendercombo.setBackground(Color.WHITE);
add(gendercombo);
```

```
JLabel lblprice = new JLabel("Car Company");
lblprice.setFont(new Font("Tahoma", Font.BOLD, 15));
lblprice.setBounds(60, 190, 120, 30);
add(lblprice);
```

```
tfcompany = new JTextField();
tfcompany.setBounds(200, 190, 150, 30);
add(tfcompany);
```

```
JLabel lbltype = new JLabel("Car Model");
lbltype.setFont(new Font("Tahoma", Font.BOLD, 15));
lbltype.setBounds(60, 230, 120, 30);
add(lbltype);
```

```
tfmodel = new JTextField();
tfmodel.setBounds(200, 230, 150, 30);
add(tfmodel);
```

```
JLabel lblavailable = new JLabel("Available");
```

```
lblavailable.setFont(new Font("Tahoma", Font.BOLD, 15));
lblavailable.setBounds(60,270, 120, 30);
add(lblavailable);

String driverOptions[] = {"Available", "Busy"};
availablecombo = new JComboBox(driverOptions);
availablecombo.setBounds(200, 270, 150, 30);
availablecombo.setBackground(Color.WHITE);
add(availablecombo);

JLabel lbllocation = new JLabel("Location");
lbllocation.setFont(new Font("Tahoma", Font.BOLD, 15));
lbllocation.setBounds(60, 310, 120, 30);
add(lbllocation);

tflocation = new JTextField();
tflocation.setBounds(200, 310, 150, 30);
add(tflocation);

add = new JButton("Add Driver");
add.setForeground(Color.WHITE);
add.setBackground(Color.BLACK);
add.setBounds(60, 370, 130, 30);

add.addActionListener(this);

add(add);

cancel = new JButton("Cancel");
cancel.setForeground(Color.WHITE);
cancel.setBackground(Color.BLACK);
cancel.setBounds(220, 370, 130, 30);

cancel.addActionListener(this);

add(cancel);

ImageIcon i1 = new ImageIcon("C:\\Users\\USER\\Documents\\eclipse-workspace\\Hotel Management
System\\src\\icons\\eleven.jpg");
Image i2 = i1.getImage().getScaledInstance(500, 300, Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel image = new JLabel(i3);
image.setBounds(400, 30, 500, 300);
add(image);

setBounds(300, 280, 980, 470);
setVisible(true);
}

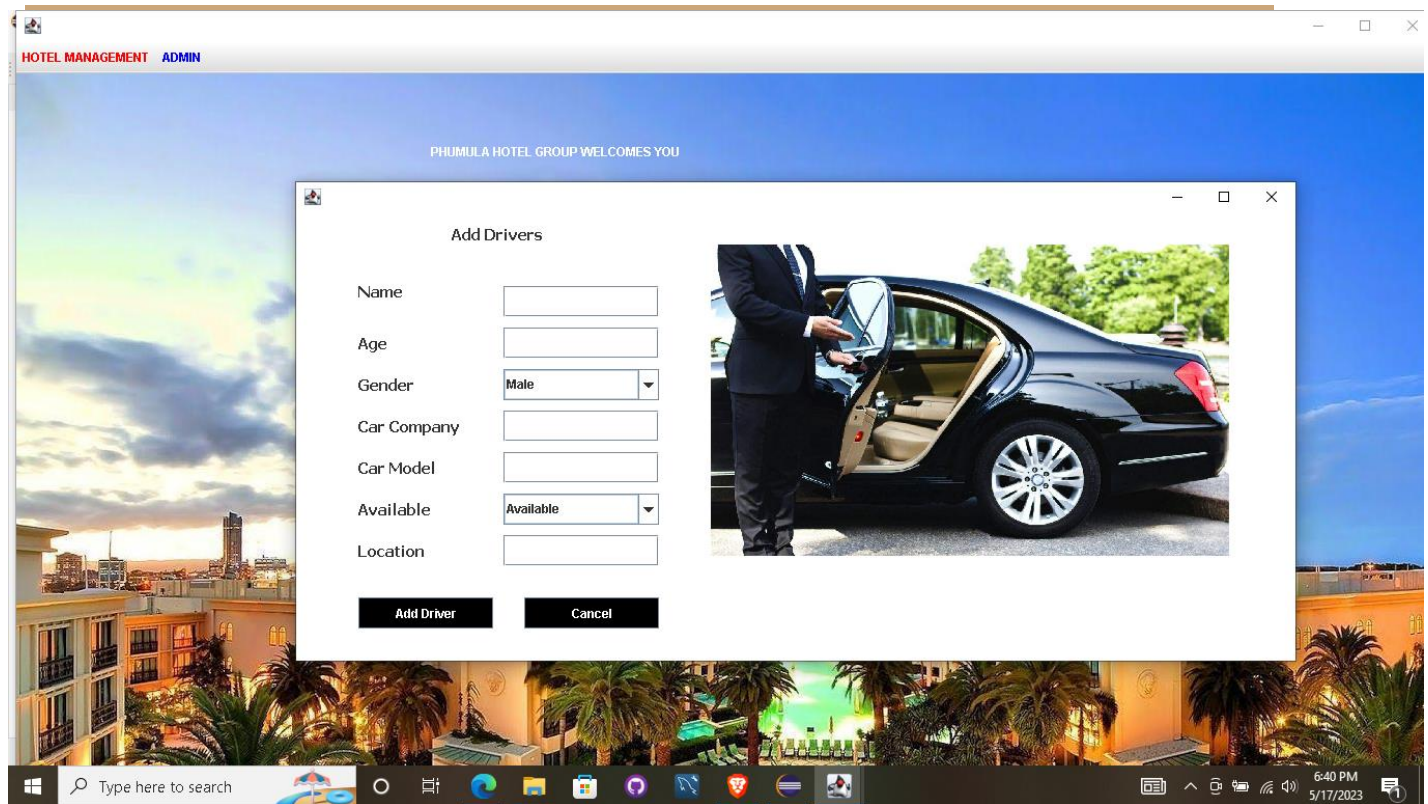
public void actionPerformed(ActionEvent ae)
{
    if (ae.getSource() == add)
    {
```

```
String name = tfname.getText();
String age = tfage.getText();
String gender = (String) gendercombo.getSelectedItem();
String company = tfcompany.getText();
String brand = tfmodel.getText();
String available = (String) availablecombo.getSelectedItem();
String location = tflocation.getText();

try
{
    conn conn = new conn();
    String str = "insert into driver(name, age, gender, company, brand, available, location) values '"+name+"',
"+age+"', '"+gender+"', '"+company+"', '"+brand+"', '"+available+"', '"+location+"'";
    conn.s.executeUpdate(str);

    JOptionPane.showMessageDialog(null, "New Driver Added Successfully");
    setVisible(false);
}
catch (Exception e)
{
    e.printStackTrace();
}
}
else
{
    setVisible(false);
}
}

public static void main(String[] args)
{
    new AddDrivers();
}
}
```



DATABASE

```
create database hotelmanagementsystem;
```

```
use hotelmanagementsystem;
```

Login Database

```
create table login(  
    username varchar(40),  
    password varchar(40)  
);
```

```
insert into login values('admin','12345');
```

Customer Database

```
create table customer(  
    id varchar(30),  
    number varchar(30),  
    name varchar(30),
```

```
gender varchar(30),  
country varchar(30),  
room_number varchar(30),  
status varchar(30),  
deposit varchar(30)  
);
```

Room Database

```
create table room(  
room_number varchar(20),  
availability varchar(20),  
clean_status varchar(20),
```

```
price varchar(20),  
bed_type varchar(30)  
);
```

Employee Database

```
create table employee(  
name varchar(30),  
age varchar(10),  
gender varchar(30),  
job varchar(30),  
salary varchar(30),  
phone varchar(30),  
aadhar varchar(30),  
email varchar(40)  
);
```

Driver Database

```
create table driver(  
name varchar(30),  
age varchar(10),  
gender varchar(20),  
company varchar(30),  
brand varchar(30),  
available varchar(10),  
location varchar(50)  
);
```

Department Database

```
create table department(  
department varchar(30),  
budget varchar(30)  
);
```

Testing

Unit Testing

Unit testing is a testing technique in which modules are tested individually. Small individual units of source code are tested to determine whether it is fit to use or not. Different modules of games are put to test while the modules are being developed. Here modules refer to individual levels, players, scenes

Integration Testing

Integration testing is the technique in which individual components or modules are grouped together and tested. It occurs after testing. The input for the integrated testing, are the modules that have already been unit tested.

System Testing

System testing is conducted on the entire system as a whole to check whether the system meets its requirements or not. 'Valar Morghulis' was installed on different systems and any errors or bugs that occurred were fixed.

Acceptance Testing

User Acceptance is defined as a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This testing happens in the final phase of testing before moving the software application to the Market or Production environment.

FUTURE SCOPE

The world is changing rapidly and so is the meaning of the Hotel Management System. Today hotel management is not only confined to hotels but has gone deep into tourism, catering, clubs, etc making it a very paying and an exciting career option.

With the rapid growth of the hotel industry pushed forward by foreign and domestic tourism and business travels, the demand for well trained and quality personnel too has gone up high. India is one of the preferred tourist and travel destinations. Approx 4.4 million tourists visit our country every year. The growth of 20% has been recorded in the tourist and hospitality industry over a few years and more growth is expected in coming years. At present, there are about 200 millions of jobs available in the industry, out of which 20% of the job opportunities are in India.

The Hotel Management System has a lot of enhancement options. In future more features may be added category-wise. It may try to analyze the user behaviour and preferences and accordingly suggest. All concepts can be applied to make the Hotel Management System more efficient.

REFERENCES

www.youtube.com

www.google
