GLOBSYN FINISHING SCHOOL

Hotel Management System

Java 11 with Swing

Hotel Management System

Group members

- NIKHIL SINGH, BENGAL COLLEGE OF ENGINEERING AND TECHNOLOGY,12500117055
- ISHA RANI, BENGAL COLLEGE OF ENGINEERING AND TECHNOLOGY,172048191
- ASHUTOSH KUMAR, BENGAL COLLEGE OF ENGINEERING AND TECHNOLOGY, 172027106
- KUMARI SHRUTI, BENGAL COLLEGE OF ENGINEERING AND TECHNOLOGY,172054212

Table of Contents

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty Prof. Chandan Mukherjee for his exemplary guidance, monitoring and constant encouragement throughout the course of this project.

The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Nikhil Singh

Isha Rani

Ashutosh Kumar

Kumari Shruti

Project Objective

The aim of this Hotel Management System project is to build a system that will able to automate many operations in a hotel. Modern day hotels aim to create a user friendly atmosphere with the availability of concierges who remember frequent visitors and making it possible to call and make reservations. While such hotels are extremely expensive, such a service can also be provided in a cost – effective manner with the use of computers.

The main objective of the java project on hotel management system is to manage the detail of guest, details of rooms, guest booking, searching, bill calculating and others.

The project is based on administrative end however user has to sign up and search the room of his choice if available they are prompted to book the room thereby. The purpose of the project is to build an application to reduce the manual work for managing the rooms, Customers, Payments and booking jobs. It tracks all the details about the booking and other hotel services.

The primary goal consists of:-

- Maintain an effective and efficient system of room booking.
- Operate a record management system for user to view available rooms.
- Maintain a record of guest who are logging in and also of those who are booking the room.
- The admin has the power to add, delete, and clear the status of the rooms in the hotel.
- The records of the added rooms with their availability are stored in the records of the admin.

Project Scope

The Scope of the project involves a User and Admin.

The User has to signup first only then he is allowed to search and book room. After Sign up user has to login with his account and then he is prompted to search the room. If the room of his choice is available then a table with the available room is displayed.

User can now book the required room with his respective check in and checkout date and book the room. After booking the bill is display on the screen.

Admin has more power and control over the software. He can add the rooms of his desire. After adding a new room its status is by default false means available and after the checkout the status becomes true.

After the checkout the room is available to the new customer. Admin

Can search the respective record of the customer details and also can

View the availability of rooms, Customer Login Details, Customer Booking details and has power to delete the records of old rooms.

Moreover the every text fields are validated so that the occurrence of error is reduced.

System Specifications

Hardware specifications

CPU- Dual core 64-bit 2.8 GHz 8.00 GT/s CPUs

Ram- 2 GB (minimum) or more.

Free Disk Space- 2 GB or more for the installation of the software's

Like Eclipse idea and Java development Toolkit and workspace of eclipse.

Software Specifications

Operating system- Any OS because Java is Platform independent.

Runtime Environment - JRE 1.7 or more

Applications- Eclipse IDE and Java development Toolkit.

Functional Requirements

Module 1: User module comprises of the search room modules and the room booking modules. User can search the room of his choice if available he/she can book that room if not available then he/she has to search another room for booking.

Module 2: login modules are a frame of the Home. Here from one window both user and admin can login to their respective modules and can do their respective work.

Module 3: The main module of the whole program. Here the admin can add the rooms, display the available room, delete the rooms , view the guest login as well as the guest booking details and also can the search the booking details of the user from the available record details here user id.

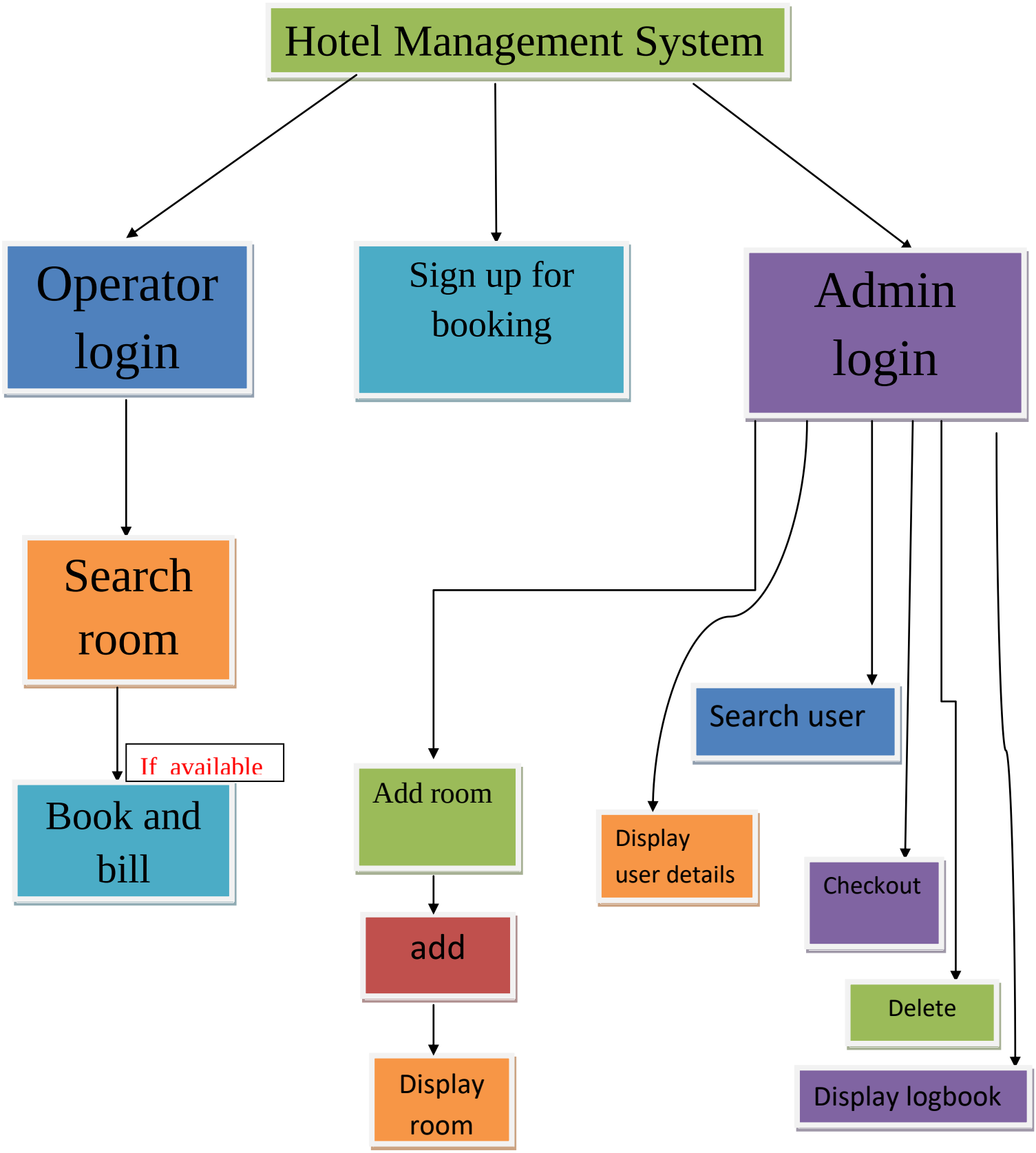
Moreover there is a module available to reduce the human error and misinterpretations these are:-

1. All the fields in signup , add room, search room, booking etc are validated to avoid any false and empty values.

2. Avoiding errors in data.

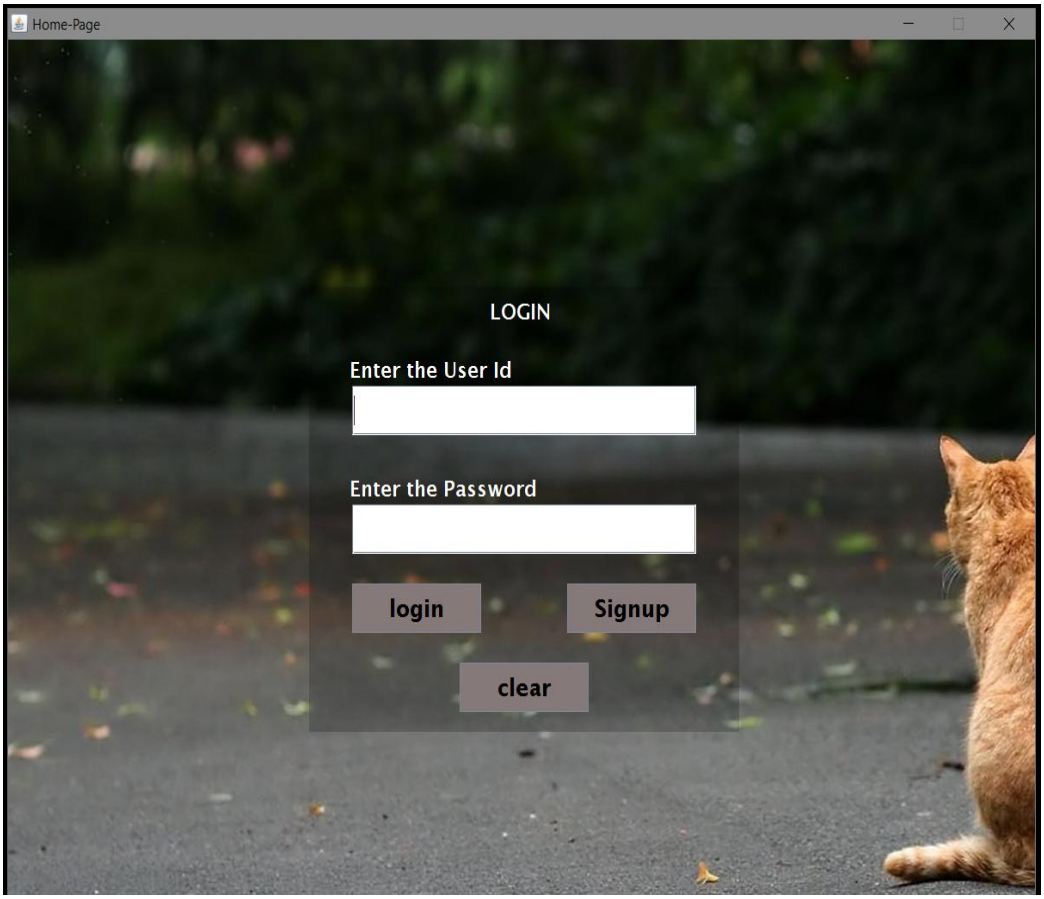
3. Controlling the amount of input in some fields which require specific characters.

Application Work Flow

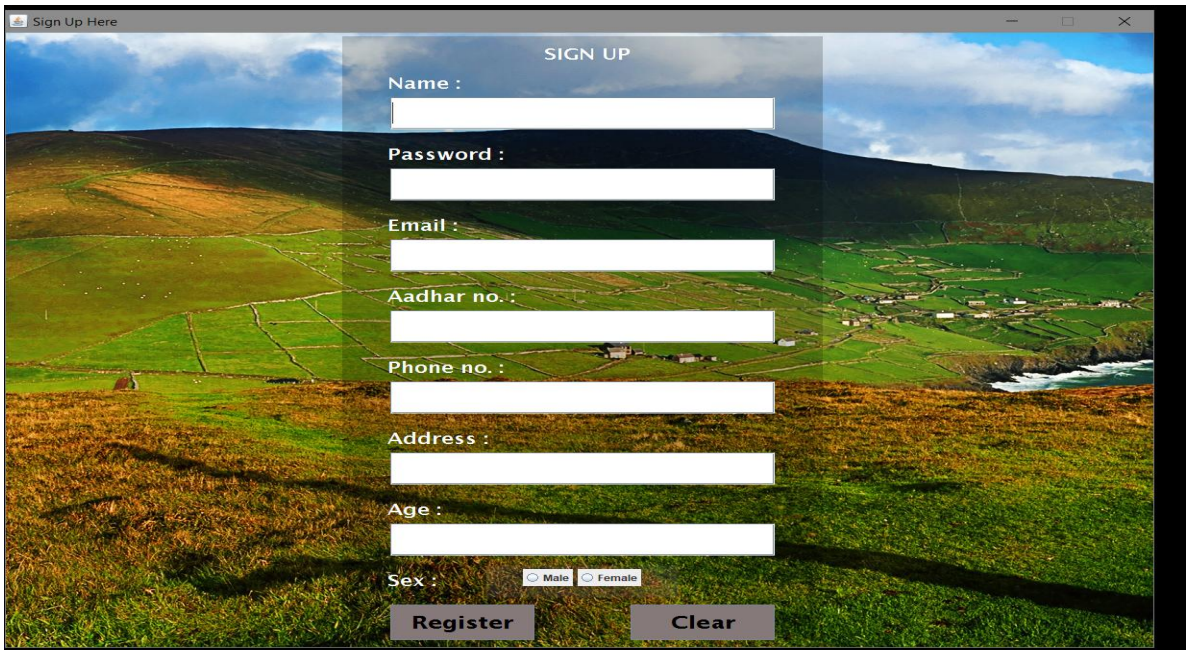


Screenshots

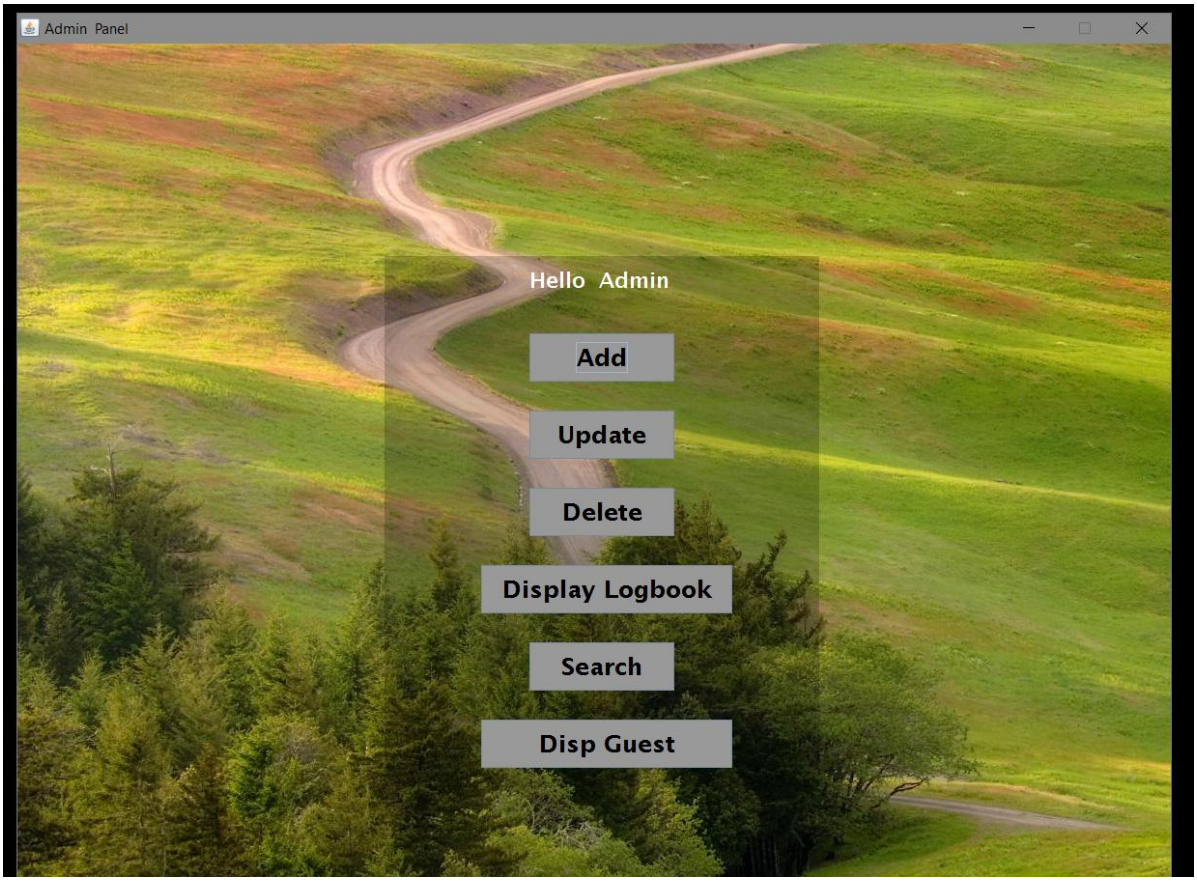
Home



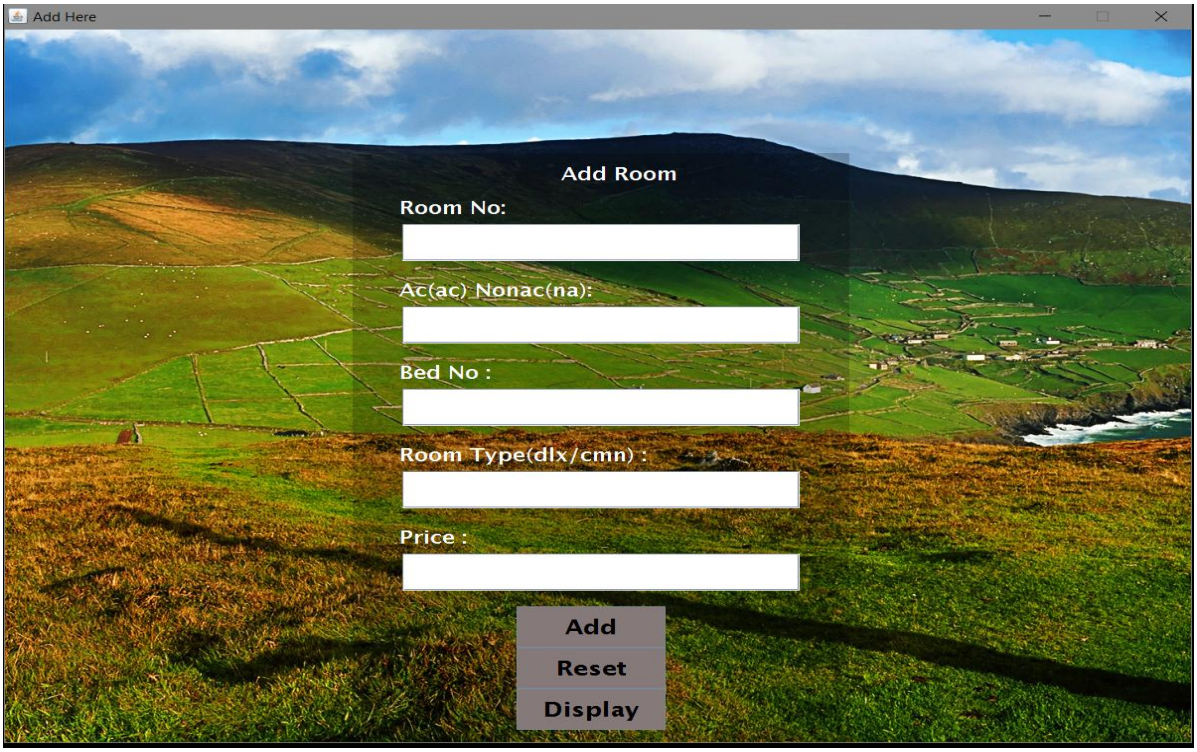
Signup



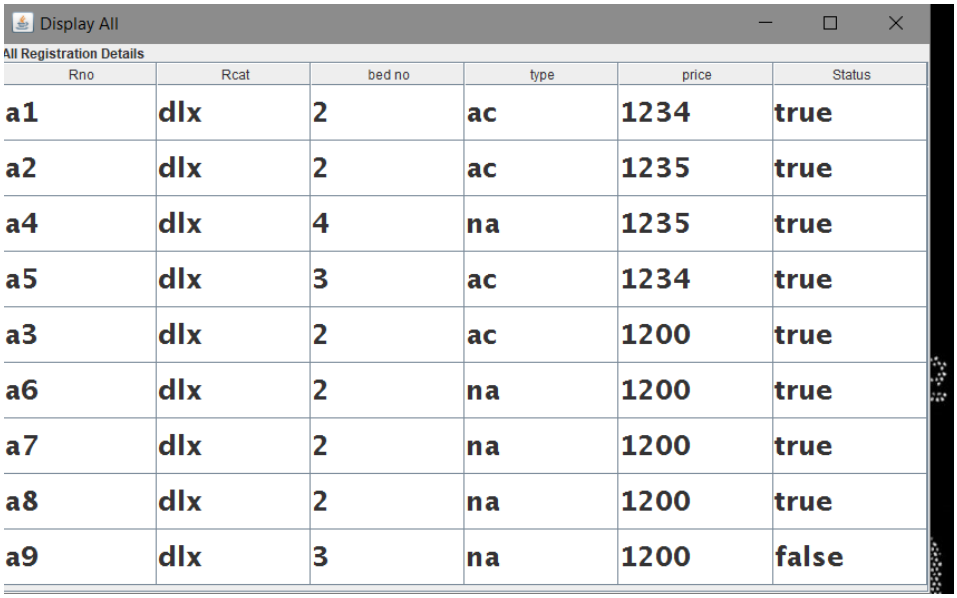
Admin login



Add room

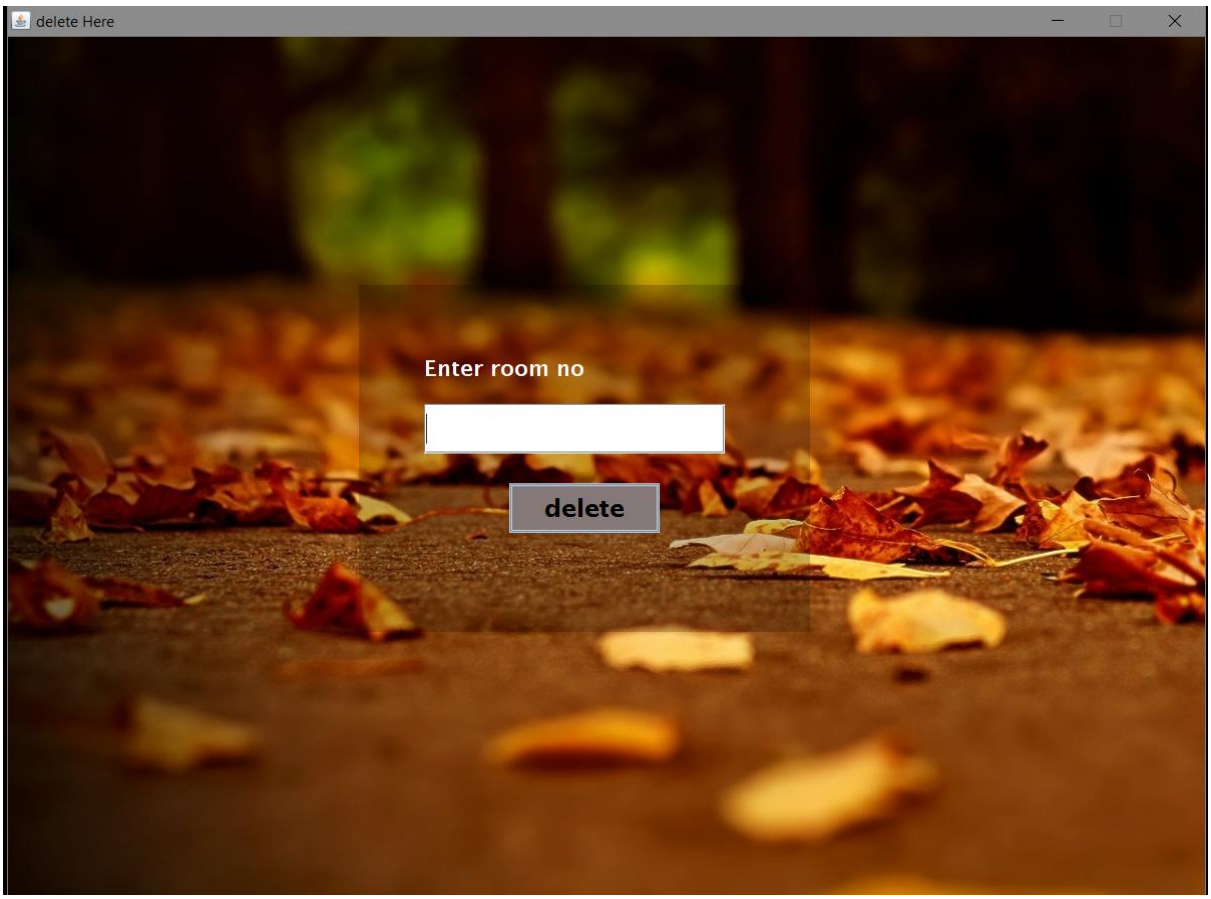


Display added room

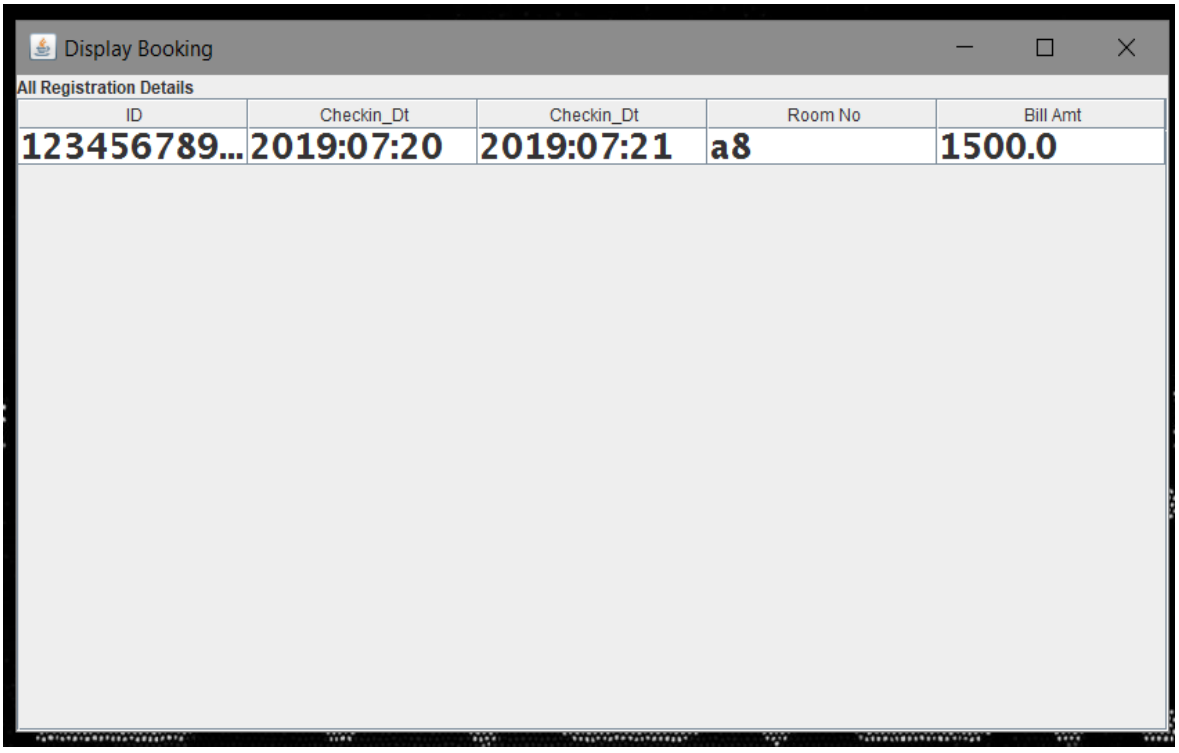


Rno	Rcat	bed no	type	price	Status
a1	dlx	2	ac	1234	true
a2	dlx	2	ac	1235	true
a4	dlx	4	na	1235	true
a5	dlx	3	ac	1234	true
a3	dlx	2	ac	1200	true
a6	dlx	2	na	1200	true
a7	dlx	2	na	1200	true
a8	dlx	2	na	1200	true
a9	dlx	3	na	1200	false

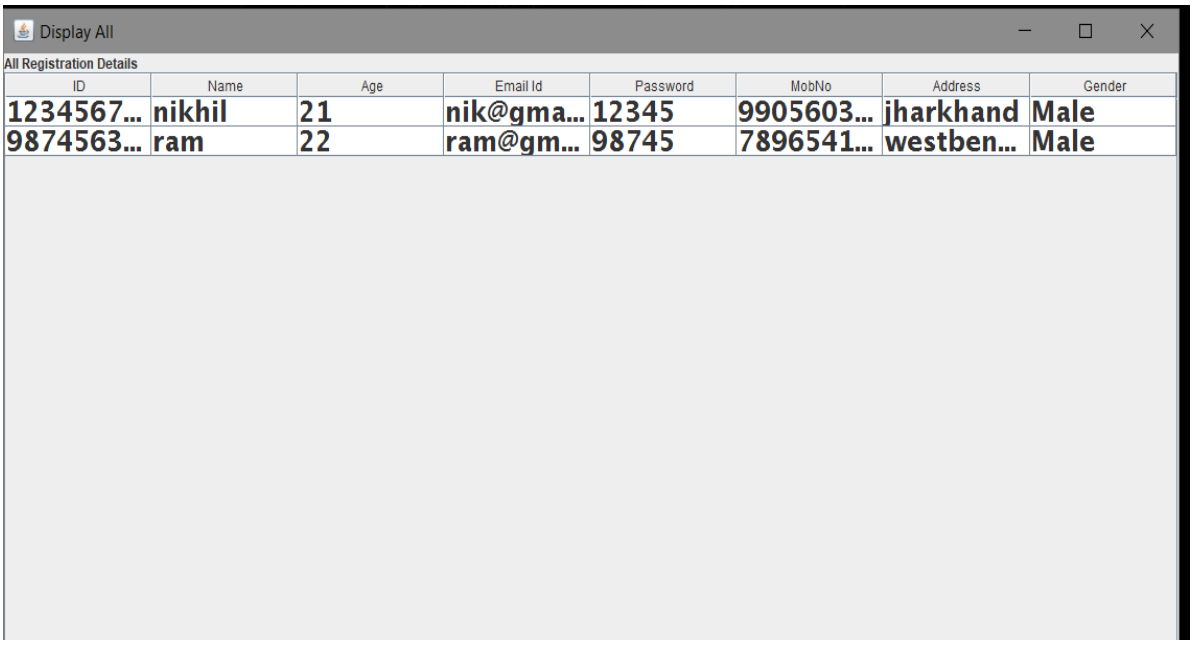
Delete room



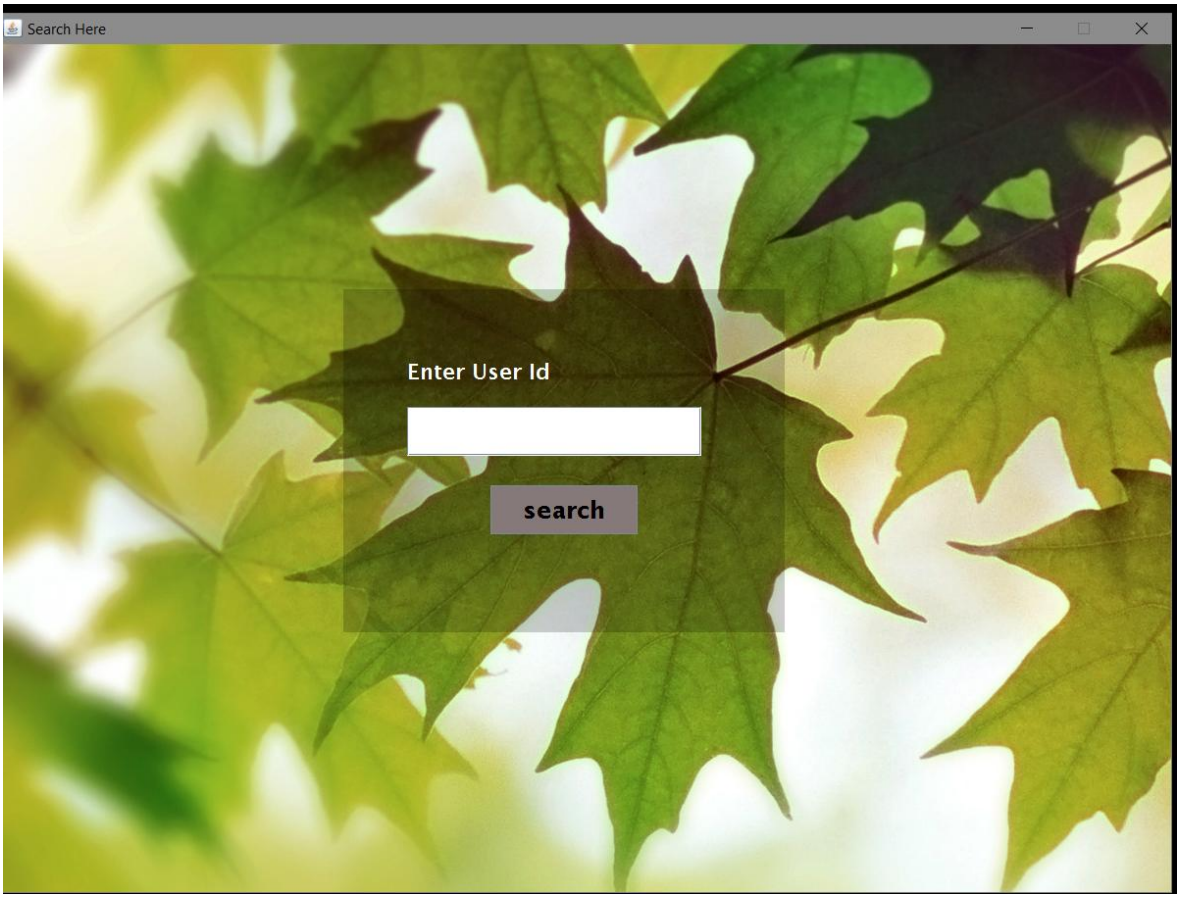
Display Booking



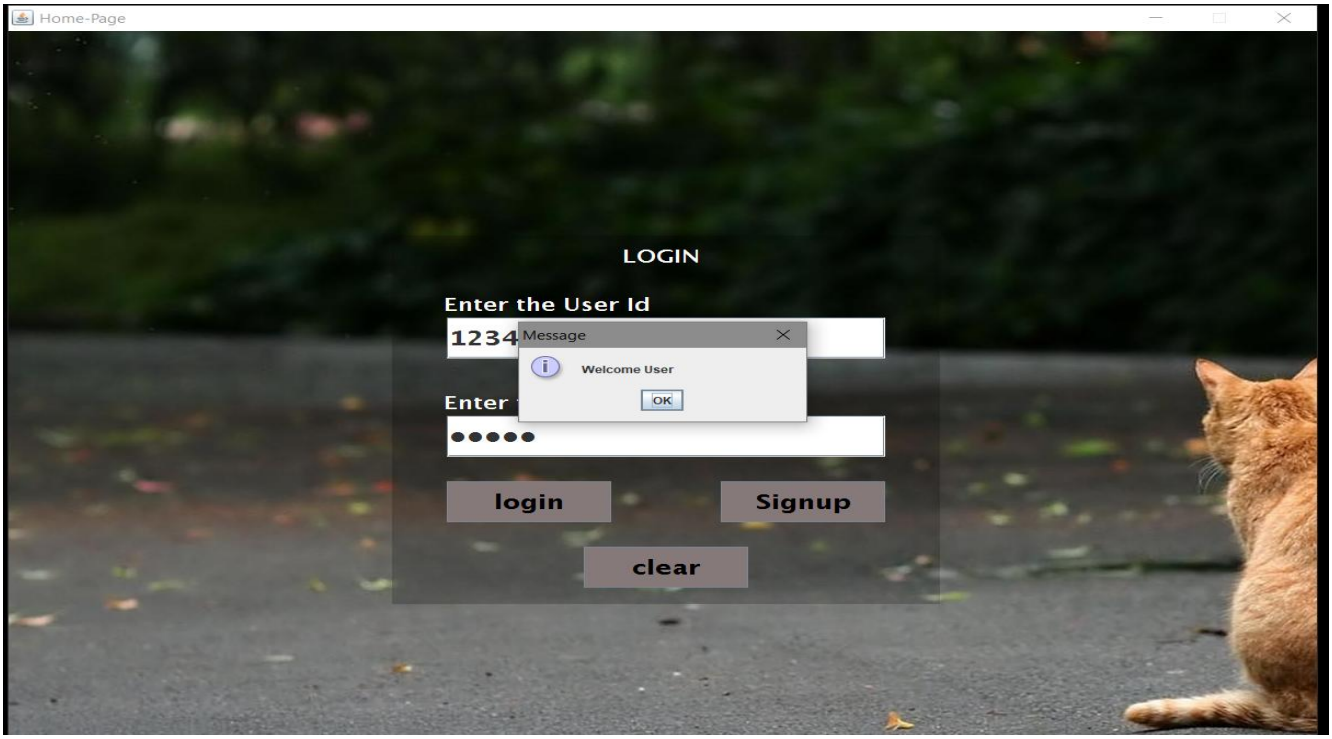
Display Guest Info



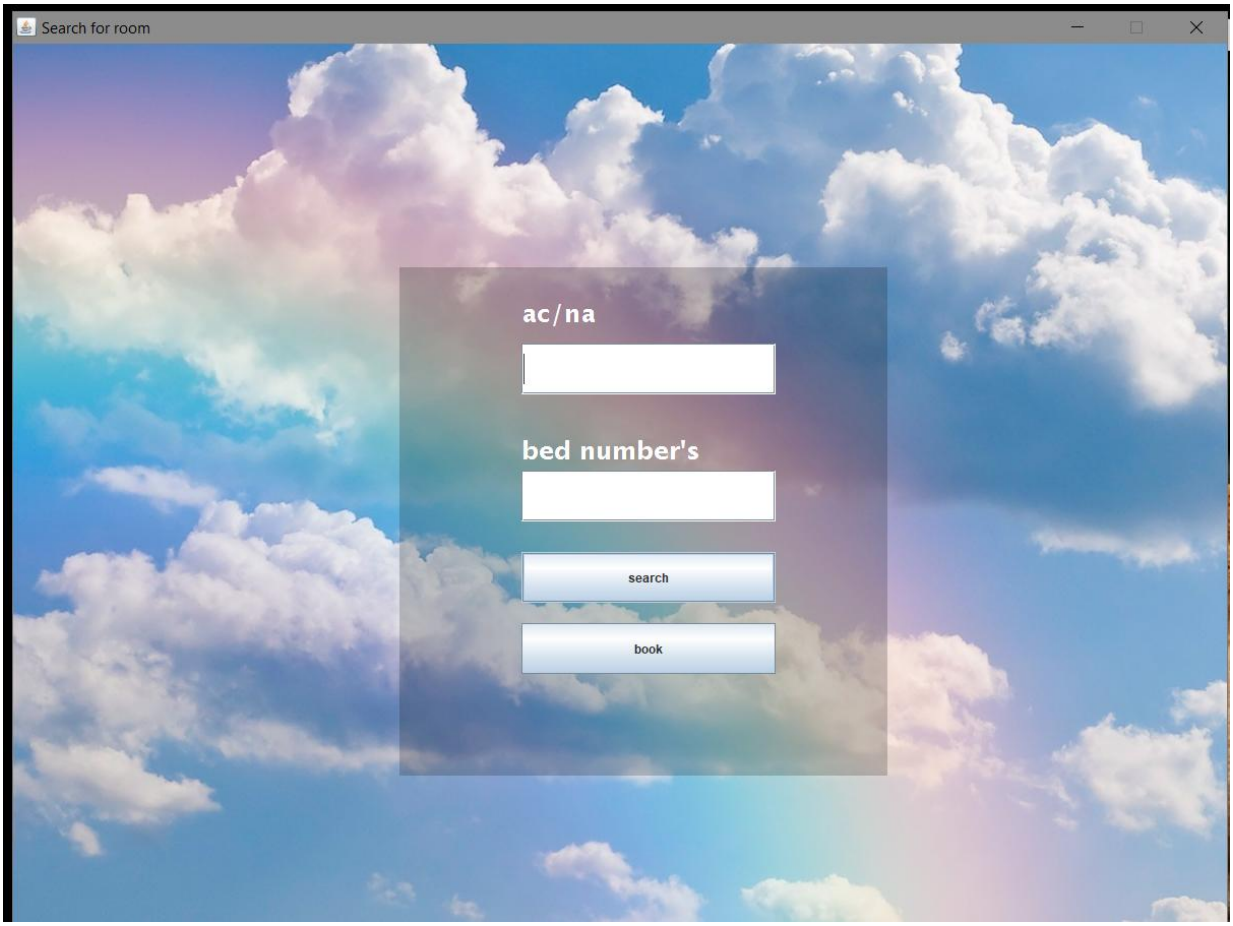
Search User



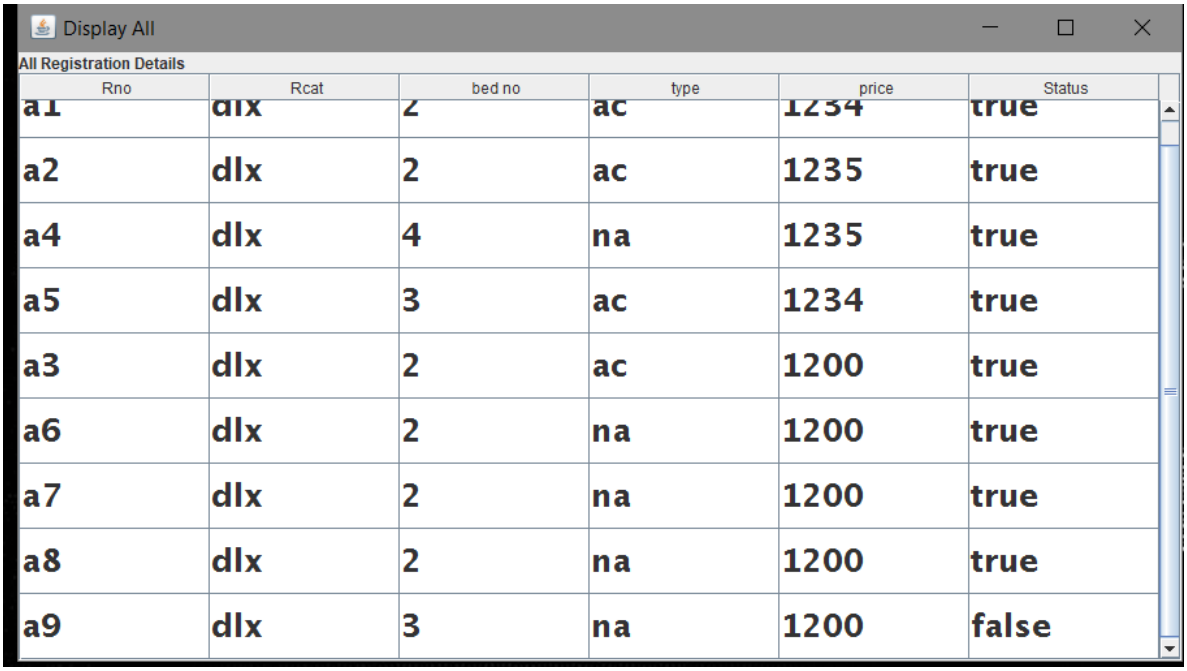
User Sign In



Searching User room

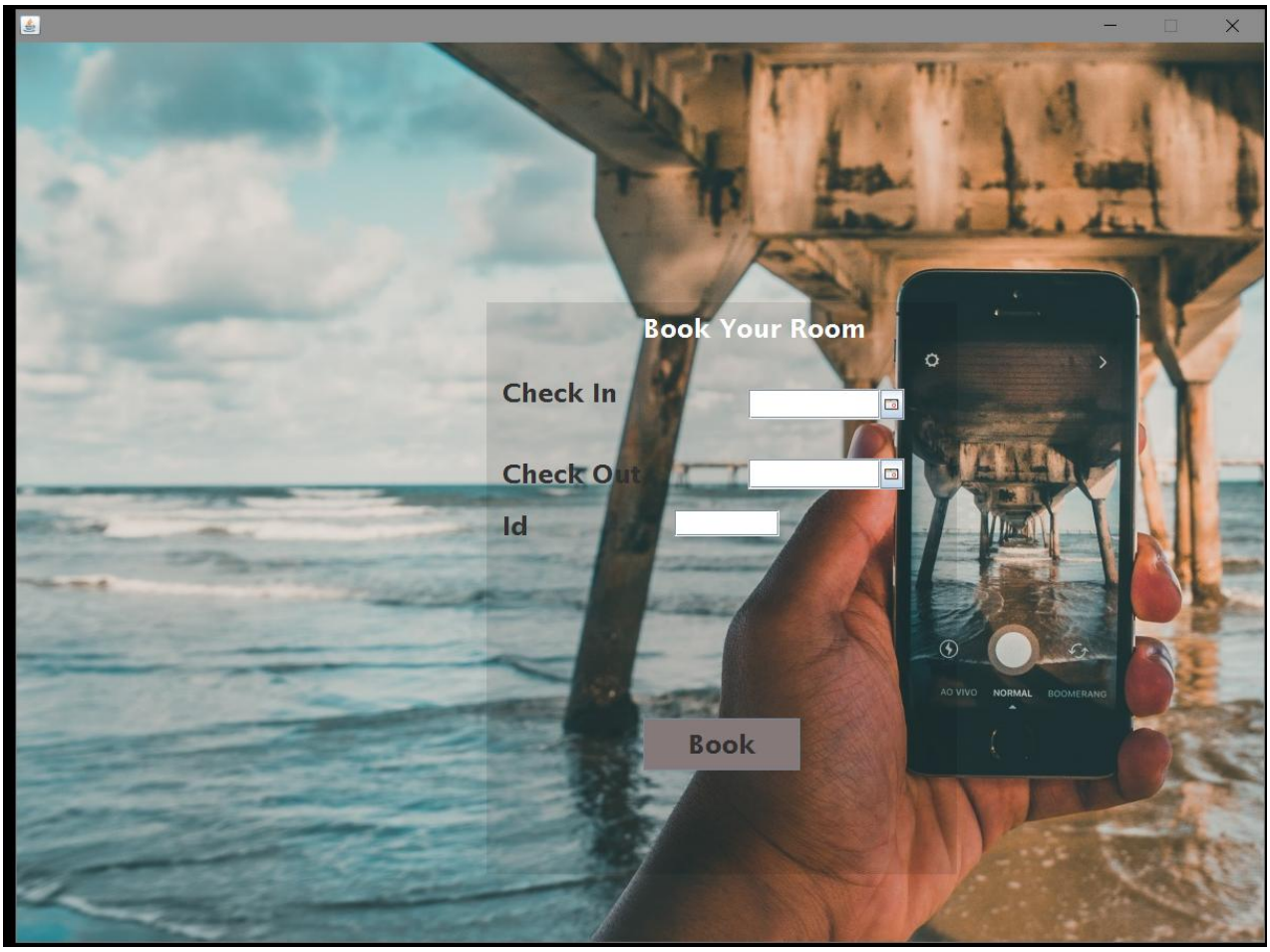


Found room

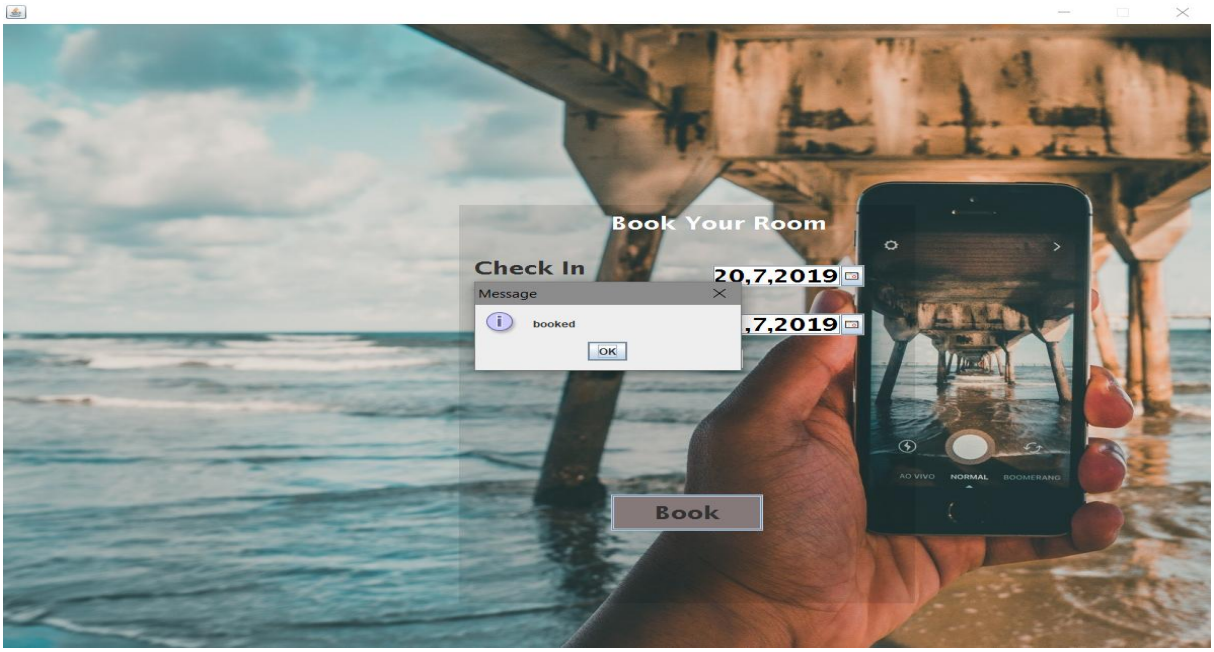


Rno	Rcat	bed no	type	price	Status
a1	dlx	2	ac	1234	true
a2	dlx	2	ac	1235	true
a4	dlx	4	na	1235	true
a5	dlx	3	ac	1234	true
a3	dlx	2	ac	1200	true
a6	dlx	2	na	1200	true
a7	dlx	2	na	1200	true
a8	dlx	2	na	1200	true
a9	dlx	3	na	1200	false

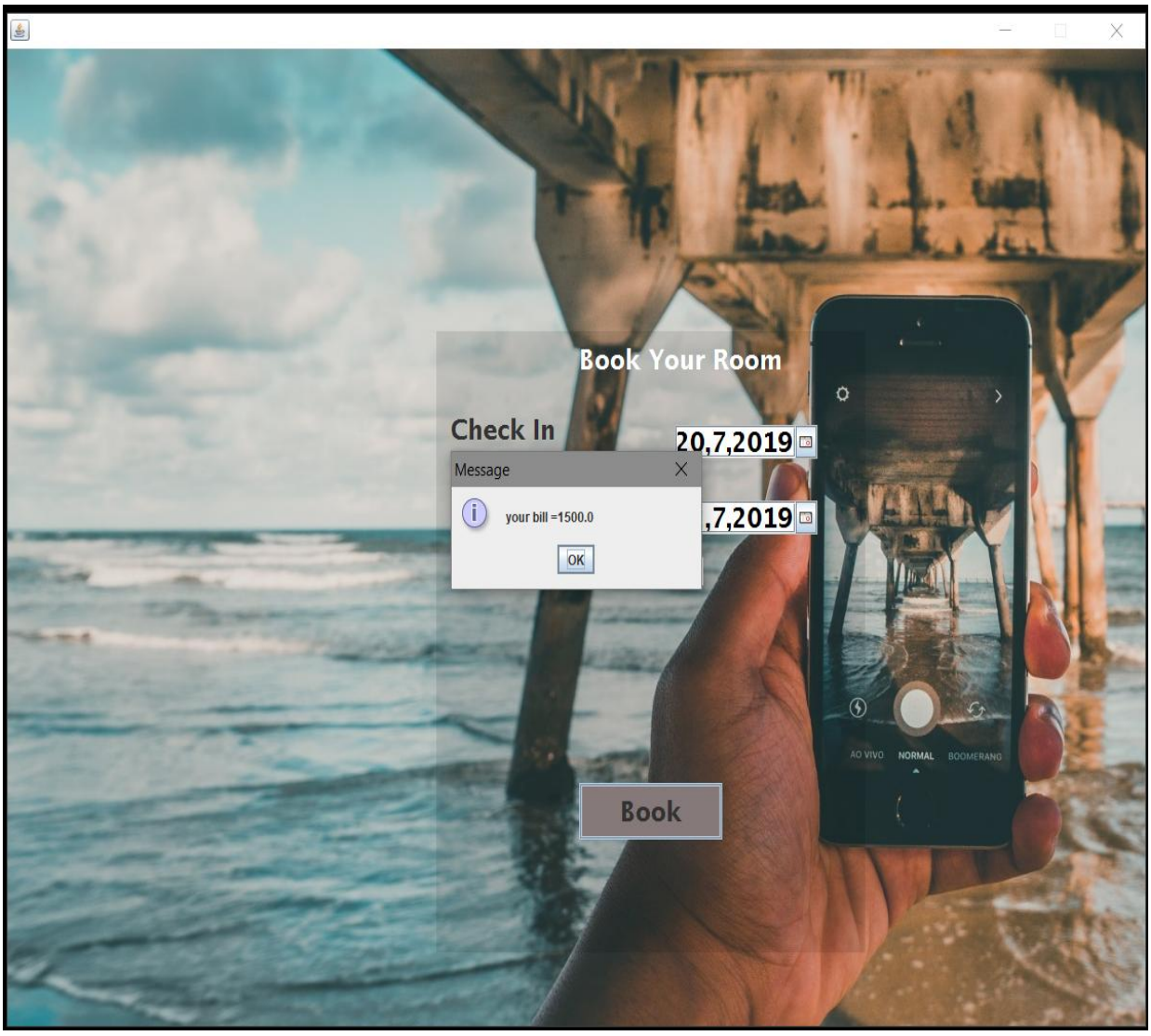
Room Booking



Book Room



Billing Room



Future Scope of Improvement

The proposed Online Hotel Management System is much more efficient in terms of human and monetary resources.

The number of staff to ensure successful implementation is reduced. Lead time of service is reduced and it is easy to predict room availabilities.

Electronic documentation of hotel operations helps management in keeping the hotel in check. A visitor can be sure of consistent service even across other hotels of the same chain.

The existing Online Hotel Management System depends on many highly trained individuals to take care of guests. Such a system is prone to error and delays. Visitors may not have a consistent experience and have to be bothered about remembering mundane things such as payments and restaurants.

A lot of paperwork is generated and it is difficult for a supervisor to go through all these documents. Allocation of rooms based on expected vacancies is also difficult as it requires extra work on the part of the employees. The costs of running such a hotel is also great.

Jdbc and online model will be much more efficient and faster than the one developed.

Code

Home Page

```
package Pack1;

import java.awt.Color;

import java.awt.Font;

import java.util.ArrayList;

import java.util.Scanner;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

class login extends JFrame

{

        private static final long serialVersionUID = 1L;

        private JButton loginb,signupb,clearb;

        private JPanel login;

        private JTextField t1;

        private JLabel l1,l2,l3;

        private JPasswordField p1;
```

```
   ArrayList<Guest_Info> emplist =
UserInfoReadWriteFromFile.readObjectFromFile();
        Guest_Info obj;
        @SuppressWarnings("deprecation")
        login()
        {super("Home-Page");
                Font f=new Font("Lucida Sans",Font.BOLD,25);
                Font f2=new Font("Lucida Sans",Font.BOLD,22);
                login = new JPanel();
                login.setLayout(null);
                setSize(450, 350);
                login.setBackground(new Color(0, 0, 0, 50));
                login.setBounds(350, 250, 500, 450);
            l3=new JLabel("LOGIN");
            l3.setBounds(210,0,250,50);
            l3.setFont(f2);
            l3.setForeground(Color.WHITE);
      l1=new JLabel("Enter the User Id");
      t1=new JTextField();
      l1.setFont(f2);
      l1.setForeground(Color.white);
      l1.setBounds(47, 60,400, 48);
      t1.setFont(f);
      t1.setBounds(50, 100, 400, 50);
```

```
l2=new JLabel("Enter the Password");
l2.setFont(f2);
l2.setForeground(Color.white);
l2.setBounds(47, 180,400, 48);
p1=new JPasswordField();
p1.setFont(f);
p1.setBounds(50, 220, 400, 50);
login.add(l3);
login.add(l1);
login.add(t1);
login.add(l2);
login.add(p1);
  loginb=new JButton("login");
loginb.setForeground(Color.black);
loginb.setBackground(new Color(134, 121, 121));
loginb.setFont(f);
  signupb=new JButton("Signup");
signupb.setForeground(Color.black);
signupb.setBackground(new Color(134, 121, 121));
signupb.setFont(f);
 clearb=new JButton("clear");
clearb.setForeground(Color.black);
clearb.setBackground(new Color(134, 121, 121));
```

```
clearb.setFont(f);
loginb.setBounds(50,300,150, 50);
login.add(loginb);
signupb.setBounds(300,300,150, 50);
login.add(signupb);
clearb.setBounds(175,380, 150, 50);
loginb.addActionListener((e) ->
{boolean isdatavalidate = dataValidation();                    if
(t1.getText().equalsIgnoreCase("admin") &&
p1.getText().equalsIgnoreCase("admin")){
JOptionPane.showMessageDialog(this, "welcome Admin");
new admin_login();
}
else if (getinfo()) {
if (isdatavalidate) {
JOptionPane.showMessageDialog(this, "Welcome User");
new searchroom();
}
} else
JOptionPane.showMessageDialog(this, "invalid id and password");});
signupb.addActionListener((e) ->
{
        new signup();
        });
```

```
        clearb.addActionListener((e) ->
    {
      this.dispose();
      new login();
    });
          login.add(clearb);
          JLabel bg_img=new JLabel();
    bg_img.setIcon(new
ImageIcon(Login_main.class.getResource("/img.jpg")));
    bg_img.setBounds(0, 0, 1200,900);
          add(bg_img);
    bg_img.add(login);
                setSize(1200,900);
                setLayout(null);
                setLocationRelativeTo(null);
                setResizable(false);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setVisible(true);
      }
      public boolean dataValidation()
 {  boolean namevalidation = nameCheck();
   boolean emailvalidation = emilcheck();
                  if (namevalidation && emailvalidation)
                 return true;
```

```
  return false;
 }
public boolean nameCheck()
{
   boolean namevalid = true;
   String namepattern = "^[a-zA-Z0-9_]*$";
   @SuppressWarnings("resource")
Scanner sc1 = new Scanner(t1.getText().trim());
   String match = sc1.findInLine(namepattern);
   if (match == null)
   {
           JOptionPane.showMessageDialog(this, "Invalid Username..Please Enter
Valid Username");
           t1.setText("");
     namevalid = false;
   }
   return namevalid;
}
public boolean emilcheck()
 {
         boolean emailvalid = true;
         String emailpattern = "^[a-zA-Z0-9]{5}$";
         @SuppressWarnings({ "deprecation", "resource" })
         Scanner sc2 = new Scanner(p1.getText().trim());
```

```
String match1 = sc2.findInLine(emailpattern);
        if (match1 == null)
        {
          JOptionPane.showMessageDialog(this, "Invalid Password..Please Enter
Valid Password");
          p1.setText("");
          emailvalid = false;
        }       return emailvalid;
 }
@SuppressWarnings("deprecation")
public boolean getinfo() {
        boolean found = false;
        try
        {
        for (int i = 0; i < emplist.size(); i++) {
                obj = emplist.get(i);
                if (t1.getText().equalsIgnoreCase(emplist.get(i).getId()) &&
p1.getText().equalsIgnoreCase(emplist.get(i).getPassword())) {
found = true;
break;}}
}catch(Exception e){
JOptionPane.showMessageDialog(this,"wrong");
}
if (found)return true; else
```

```
return false;
}}
public class Login_main {
	public static void main(String[] args) {
		new login();}
}
```

Admin Login

```
package Pack1;

import java.awt.Color;
import java.awt.Font;
import java.time.LocalDate;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
public class admin_login extends JFrame
{
	private static final long serialVersionUID = 1L;
	private JPanel login;
	private JLabel l3;
```

```
private JButton addb,updateb,deleteb,displayb,prevb,searchb;
private String chkout="";
private String sysdate;
public admin_login()
{
        super("Admin  Panel");
        Font f=new Font("Lucida Sans",Font.BOLD,25);
        Font f2=new Font("Lucida Sans",Font.BOLD,22);
        login = new JPanel();
        login.setLayout(null);
        setSize(450, 500);
        login.setBackground(new Color(0, 0, 0, 50));
        login.setBounds(380, 220, 450, 550);
      l3=new JLabel("Hello  Admin");
    l3.setBounds(150,0,250,50);
    l3.setFont(f2);
    l3.setForeground(Color.WHITE);
    addb = new JButton("Add");
        addb.setForeground(Color.black);
        addb.setBackground(new Color(153, 153, 153));
        addb.setFont(f);
        updateb = new JButton("Update");
        updateb.setForeground(Color.black);
```

```
        updateb.setBackground(new Color(153, 153, 153));
        updateb.setFont(f);
        displayb = new JButton("Display Logbook");
        displayb.setForeground(Color.black);
        displayb.setBackground(new Color(153, 153, 153));
        displayb.setFont(f);
        deleteb = new JButton("Delete");
        deleteb.setForeground(Color.black);
        deleteb.setBackground(new Color(153, 153, 153));
        deleteb.setFont(f);
        prevb = new JButton("Disp Guest");
        prevb.setForeground(Color.black);
        prevb.setBackground(new Color(153, 153, 153));
        prevb.setFont(f);
    searchb = new JButton("Search");
        searchb.setForeground(Color.black);
        searchb.setBackground(new Color(153, 153, 153));
        searchb.setFont(f);
        addb.addActionListener((e) ->
        {new Add();});
        searchb.addActionListener((e) ->
        {new Search();});
```

```
deleteb.addActionListener((e) ->{
        new delete();});
displayb.addActionListener((e) ->
{new disptable2();
                });            prevb.addActionListener((e) ->
                {                       new display();
                });            updateb.addActionListener((e) ->
                {
                        String a=" ";
                        LocalDate today = LocalDate.now();
                        int day = today.getDayOfMonth();
                        int month = today.getMonthValue();
                        int year = today.getYear();
                          sysdate =year+":"+month+":"+day;
                    //sysdate="2019:07:20";
                        String room;
                          ArrayList<new_info>blist;
                          blist=UserCheckInData.readObjectFromFile();
                          for(new_info re:blist)
                          {chkout=re.getChkout();
                          if(sysdate.equals(chkout))
                          { room=re.getIntr();
```

```
                ArrayList<room_info> rlist;
                rlist=Roomwrite.readObjectFromFile();
                for(room_info r:rlist)
                {       if(room.equals(r.getRno()))
               {
                a+=room+" ";
               r.setStatus(false);
               boolean chk=r.isStatus();
                System.out.println(chk);
                Roomwrite.writeObjecttoFile(rlist);
                break;
                }}}}
                 System.out.println(a);        });
    addb.setBounds(150, 80, 150, 50);
    updateb.setBounds(150, 160, 150, 50);
    deleteb.setBounds(150, 240, 150, 50);
    displayb.setBounds(100, 320, 260, 50);
    searchb.setBounds(150, 400, 150, 50);
prevb.setBounds(100,480, 260,50);
 login.add(l3);
 login.add(addb);
 login.add(updateb);
```

```
            login.add(deleteb);
            login.add(displayb);
            login.add(searchb);
            login.add(prevb);
               JLabel bg_img = new JLabel();
               bg_img.setIcon(new
ImageIcon(admin_login.class.getResource("/img2.jpg")));
               bg_img.setBounds(0, 0, 1200, 900);
               add(bg_img);
               bg_img.add(login);
               setSize(1200,900);
               setLayout(null);
               setLocationRelativeTo(null);
               setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
               setVisible(true);
                 }}
```

Search Room

```
package Pack1;
import java.awt.Color;
import java.awt.Font;
import java.util.ArrayList;
```

```
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
public class searchroom extends JFrame
{
	private static final long serialVersionUID = 1L;
	int flag = 0;
	String room="";
	String id="";
	private JLabel l1,l2;
	private JPanel login;
	private JTextField t1,t2;
	private JButton b;
	searchroom()
	{
		super("Search for room");
		login = new JPanel();
		login.setLayout(null);
```

```
        setSize(450, 500);
        login.setBackground(new Color(0, 0, 0, 50));
        login.setBounds(380, 220, 480, 500);
      Font f=new Font("Lucida Sans",Font.BOLD,25);
        l1=new JLabel("ac/na");
        l1.setFont(f);
        l1.setForeground(Color.WHITE);
    l1.setBounds(120,20, 250,50);
       t1=new JTextField();
    t1.setFont(f);
    t1.setBounds(120,75,250,50);
    l2=new JLabel("bed number's");
    l2.setFont(f);
    l2.setForeground(Color.WHITE);
    l2.setBounds(120,155,250,50);
     t2=new JTextField();
    t2.setFont(f);
    t2.setBounds(120,200,250,50);
       b=new JButton("search");
    b.setBounds(120,280,250,50);
    JButton  book=new JButton("book");
  book.setBounds(120,350,250,50);
```

```
b.addActionListener((e) ->
{try {
    ArrayList<room_info> rlist;
        rlist = Roomwrite.readObjectFromFile();
        String data[][];
        data = new String[rlist.size()][12];
        int r = 0;
            for (room_info re : rlist) {
                data[r][0] = re.getRno();
                data[r][1] = re.getRcat();
                data[r][2] = re.getBno();
                data[r][3] = re.getRtype();
                data[r][4] = re.getPrice();
                data[r][5] =
Boolean.toString(re.isStatus());
                int bn = Integer.parseInt(data[r][2]);
                int t2n = Integer.parseInt(t2.getText());
                if
(data[r][3].equalsIgnoreCase(t1.getText()) && bn == t2n
                        &&
data[r][5].equalsIgnoreCase("false")) {
                    flag = 1;
                    room = data[r][0];
                    new disproom();
```

```
                    break;
                    }
                    r++;
                    }
                                        if (flag == 0) {
                        JOptionPane.showMessageDialog(this,
"room not available");
                    }
                } catch (Exception a) {
                    JOptionPane.showMessageDialog(this, "please
fill the text fields");
                }
        });
        book.addActionListener((s) ->
        {
            if(flag==1)
            new user_login(room);
            else
JOptionPane.showMessageDialog(this,"search the room first ");
        });
      login.add(l1);
      login.add(t1);
      login.add(l2);
```

```
        login.add(t2);
        login.add(b);
        login.add(book);
        JLabel bg_img = new JLabel();
        bg_img.setIcon(new
ImageIcon(admin_login.class.getResource("/img3.jpg")));
        bg_img.setBounds(0, 0, 1200, 900);
        add(bg_img);
        bg_img.add(login);
        setSize(1200,900);
        setLayout(null);
        setLocationRelativeTo(null);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setVisible(true);}}
```

Room Booking

```
public class user_login extends JFrame {
private static final long serialVersionUID = 1L;
 private String yolo;
 private String pric;
private JPanel login;
private JLabel l3;
JDateChooser dateChooser_chkin = new JDateChooser();
JDateChooser dateChooser_chkout = new JDateChooser();

double cost,tcost;
long checkdiff;
```

```
        private String ids="";
                private ArrayList<new_info> blist=new
ArrayList<new_info>();
        new_info rt;
        private ArrayList<room_info> rlist=new
ArrayList<room_info>();
        room_info rx;
        public user_login(String roomno)
        {
                Font f=new Font("Lucida Sans",Font.BOLD,25);
                login = new JPanel();
                login.setLayout(null);
                setSize(450,550);
                login.setBackground(new Color(0,0,0,25));
                login.setBounds(450,250,450,550);
                 l3=new JLabel("Book Your Room ");
            l3.setBounds(150,0,250,50);
            l3.setFont(f);
            l3.setForeground(Color.WHITE);
            login.add(l3);

                JLabel lblCheckInDate = new JLabel("Check In");
                lblCheckInDate.setFont(f);
                lblCheckInDate.setBounds(15, 72, 137, 30);
                login.add(lblCheckInDate);

                JLabel lblCheckOutDate = new JLabel("Check Out");
                lblCheckOutDate.setFont(f);
                lblCheckOutDate.setBounds(15, 150, 162, 30);
                login.add(lblCheckOutDate);

                JLabel rno= new JLabel("Id");
                rno.setFont(f);
                rno.setBounds(15, 200, 202, 30);
                login.add(rno);
                JTextField rn1=new JTextField();
                rn1.setFont(f);
```

```
        rn1.setBounds(180,200,100, 25);
        login.add(rn1);
                JButton btnCheckRooms = new JButton("Book");
                btnCheckRooms.setBackground(new Color(134, 121,
121));
                btnCheckRooms.setIcon(null);
                btnCheckRooms.setFont(f);
                btnCheckRooms.setBounds(150,400,150,50);
                login.add(btnCheckRooms);
                                dateChooser_chkin.setBorder(new
LineBorder(new Color(0, 0, 0,50)));
                dateChooser_chkin.setDateFormatString("d,M,yyyy");
                dateChooser_chkin.setFont(f);
                dateChooser_chkin.setBounds(250, 83, 150,30);
                login.add(dateChooser_chkin);
                dateChooser_chkout.setDateFormatString("d,M,yyyy");
                dateChooser_chkout.setBorder(new LineBorder(new
Color(0, 0, 0,50)));
                dateChooser_chkout.setFont(f);
                dateChooser_chkout.setBounds(250, 150, 150, 30);
                login.add(dateChooser_chkout);
                btnCheckRooms.addActionListener(new ActionListener()
{
                        public void actionPerformed(ActionEvent arg0) {

try
{SimpleDateFormat dFormat=new
SimpleDateFormat("YYYY:MM:dd");
String d1=dFormat.format(dateChooser_chkin.getDate());
SimpleDateFormat dFormat1=new
SimpleDateFormat("YYYY:MM:dd");
String d2=dFormat1.format(dateChooser_chkout.getDate());
String t[]=new String[3];
t=d1.split(":");
final LocalDate chkindate,chkoutdate;
```

```
        chkindate=LocalDate.of(Integer.parseInt(t[0]),Integer.parseInt(t
[1]),Integer.parseInt(t[2]));
                        t=d2.split(":");
        chkoutdate=LocalDate.of(Integer.parseInt(t[0]),Integer.parseInt
(t[1]),Integer.parseInt(t[2]));
                        if(chkindate.compareTo(chkoutdate)<=0)
                        {
                        long
d=ChronoUnit.DAYS.between(chkindate,chkoutdate);
                        String g=String.valueOf(d);
                        rlist = Roomwrite.readObjectFromFile();
                    //data = new String[rlist.size()][12];
                        for(room_info rx : rlist)
                    {
                                yolo=rx.getRno();
                                pric=rx.getPrice();
                                if(yolo.equalsIgnoreCase(roomno))
                                {
                                        rx.setStatus(true);
                                        System.out.println(rx.getRno()+"
"+rx.getRcat()+" "+rx.getBno()+" "+rx.getRtype()+"
"+rx.getPrice()+" "+rx.isStatus());
                                        Roomwrite.writeObjecttoFile(rlist);
                                        break;
                                }
                    }
                    ids=rn1.getText();
                       double price = bill(pric,g);
                        String price2 = String.valueOf(price);
                          rlist=Roomwrite.readObjectFromFile();
                          rt=new new_info(d1,d2,price2,roomno,ids);
                          blist =
UserCheckInData.readObjectFromFile();
                         blist.add(rt);
                        JOptionPane.showMessageDialog(null,"booked");
JOptionPane.showMessageDialog(null,"your bill ="+ price2);
```

```
 UserCheckInData.writeObjecttoFile(blist);
                        }
                    else
                        {
                        JOptionPane.showMessageDialog(null,"Invalid
dates");
                        }
                    }catch(Exception e)
                    {
                       e.printStackTrace();
                        JOptionPane.showMessageDialog(null,"Empty
fields");
                    }
                    }              });
            JLabel bg_img = new JLabel();
            bg_img.setIcon(new
ImageIcon(user_login.class.getResource("/img9.jpg")));
            bg_img.setBounds(0, 0, 1200,900);
            add(bg_img);
            bg_img.add(login);
            setSize(1200, 900);
            setLayout(null);
            setLocationRelativeTo(null);
            setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            setVisible(true);
        }
        public double bill(String price,String day)
        {
            double  net = 0.0;
        int diff=Integer.parseInt(day);
        double pric=Double.parseDouble(price);
            if (diff >= 0)
            {
           net=(pric*0.25)+pric;
            }
            else{net=0.0;}return net;}}
```

Signup form

```
public class signup extends JFrame {
private static final long serialVersionUID = 1L;
private JButton signupb, clearb;
	private JPanel sign;
	private JTextField t1, t2, t3, t4, t5, t6, t7;
	private JLabel l0, l1, l2, l3, l4, l5, l6, l7, l8;
	private JRadioButton m, fe;
	private ArrayList<Guest_Info> emplist = new
ArrayList<Guest_Info>();
	Guest_Info re;
	Guest_Info r;
	signup() {
		super("Sign Up Here");
		Font f = new Font("Lucida Sans", Font.BOLD, 25);
		Font f2 = new Font("Lucida Sans", Font.BOLD, 22);
		sign = new JPanel();
		sign.setLayout(null);
		setSize(450, 350);
		sign.setBackground(new Color(0, 0, 0, 50));
		sign.setBounds(350, 5, 500, 850);
		l0 = new JLabel("SIGN UP");
		l0.setBounds(210, 0, 250, 50);
		l0.setFont(f2);
		l0.setForeground(Color.WHITE);
		l1 = new JLabel("Name :");
		t1 = new JTextField();
		l1.setFont(f2);
		l1.setForeground(Color.white);
		l1.setBounds(47, 45, 400, 42);
		t1.setFont(f);
		t1.setBounds(50, 86, 400, 45);
		l2 = new JLabel("Password :");
		t2 = new JTextField();
		l2.setFont(f2);
		l2.setForeground(Color.white);
		l2.setBounds(47, 145, 400, 42);
```

```
t2.setFont(f);
t2.setBounds(50, 186, 400, 45);
l3 = new JLabel("Email :");
t3 = new JTextField();
l3.setFont(f2);
l3.setForeground(Color.white);
l3.setBounds(47, 245, 400, 42);
t3.setFont(f);
t3.setBounds(50, 286, 400, 45);
l4 = new JLabel("Aadhar no. :");
t4 = new JTextField();
l4.setFont(f2);
l4.setForeground(Color.white);
l4.setBounds(47, 345, 400, 42);
t4.setFont(f);
t4.setBounds(50, 386, 400, 45);
l5 = new JLabel("Phone no. :");
t5 = new JTextField();
l5.setFont(f2);
l5.setForeground(Color.white);
l5.setBounds(47, 445, 400, 42);
t5.setFont(f);
t5.setBounds(50, 486, 400, 45);
l6 = new JLabel("Address :");
t6 = new JTextField();
l6.setFont(f2);
l6.setForeground(Color.white);
l6.setBounds(47, 545, 400, 42);
t6.setFont(f);
t6.setBounds(50, 586, 400, 45);
l7 = new JLabel("Age :");
t7 = new JTextField();
l7.setFont(f2);
l7.setForeground(Color.white);
l7.setBounds(47, 645, 400, 42);
t7.setFont(f);
t7.setBounds(50, 686, 400, 45);
```

```
l8 = new JLabel("Sex :");
l8.setFont(f2);
l8.setForeground(Color.white);
l8.setBounds(47, 745, 400, 42);
m = new JRadioButton("Male");
fe = new JRadioButton("Female");
ButtonGroup zz = new ButtonGroup();
zz.add(m);
zz.add(fe);
JPanel xx = new JPanel();
xx.add(m);
xx.add(fe);
xx.setBounds(150, 745, 200, 45);
xx.setBackground(new Color(134, 121, 121, 50));
sign.add(l0);
sign.add(l1);
sign.add(t1);
sign.add(l2);
sign.add(t2);
sign.add(l3);
sign.add(t3);
sign.add(l4);
sign.add(t4);sign.add(l5);
sign.add(t5);
sign.add(l6);
sign.add(t6);
sign.add(l7);
sign.add(t7);
sign.add(l8);
sign.add(xx);
signupb = new JButton("Register");
signupb.setForeground(Color.black);
signupb.setBackground(new Color(134, 121, 121));
signupb.setFont(f);
signupb.addActionListener((e) -> {
        if (dataValidation()) {
                        add();
```

```
                        new Login_main();
                    }
                else {
                    JOptionPane.showMessageDialog(null, "please
fill the form properly");
                }
            });
            clearb = new JButton("Clear");
            clearb.setForeground(Color.black);
            clearb.setBackground(new Color(134, 121, 121));
            clearb.setFont(f);
            clearb.addActionListener((e) -> {
                this.dispose();
                new signup();
            });
            signupb.setBounds(50, 800, 150, 50);
            sign.add(signupb);
            clearb.setBounds(300, 800, 150, 50);
            sign.add(clearb);
            JLabel bg_img = new JLabel();
            bg_img.setIcon(new
ImageIcon(signup.class.getResource("/img7.jpg")));
            bg_img.setBounds(0, 0, 1200, 900);
            add(bg_img);
            bg_img.add(sign);
            setSize(1200, 900);
            setLayout(null);
            setLocationRelativeTo(null);
            setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            setVisible(true);
        }
        public void add() {
            String id, pass, name, emailid, mobno, address, gender,
age;
            id = t4.getText().trim();
            pass = t2.getText().trim();
```

```
            name = t1.getText().trim();
            emailid = t3.getText().trim();
            mobno = t5.getText().trim();
            address = t6.getText().trim();
            age = t7.getText().trim();
            gender = "";
            if (m.isSelected()) {
                  gender = "Male";
            } else if (fe.isSelected()) {
                  gender = "Female";
            }
            re = new Guest_Info(id, pass, name, emailid, mobno,
address, gender, age);
            emplist =
UserInfoReadWriteFromFile.readObjectFromFile();
            emplist.add(re);
            UserInfoReadWriteFromFile.writeObjecttoFile(emplist);
            JOptionPane.showMessageDialog(this, "Registered,Now
you can book Rooms");
      }
      public boolean dataValidation() {
            boolean namevalidation = nameCheck();
            boolean emailvalidation = emilcheck();
            boolean mobnovalidation = mobnocheck();
            boolean passvalidation = passcheck();
            boolean aadharvalidation = aadharcheck();
            boolean agevalidation = agecheck();
            if (namevalidation && emailvalidation &&
mobnovalidation && passvalidation && aadharvalidation &&
agevalidation)
                  return true;
            return false;
      }
      public boolean nameCheck() {
            boolean namevalid = true;
            String namepattern = "^[a-zA-Z]{3,12}$";
            @SuppressWarnings("resource")
```

```
            Scanner sc1 = new Scanner(t1.getText().trim());
            String match = sc1.findInLine(namepattern);

            if (match == null) {
                  JOptionPane.showMessageDialog(this, "Invalid
Name..Please Enter Another Name");
                  t1.setText("");
                  // t1.setBackground(Color.RED);
                  namevalid = false;
            }
            return namevalid;
      }
      public boolean emilcheck() {
            boolean emailvalid = true;
            String emailpattern = "^(\\w)+@(\\w+\\.)(\\w+\\.)?[A-Za-
z]+$";
            @SuppressWarnings("resource")
            Scanner sc2 = new Scanner(t3.getText().trim());
            String match1 = sc2.findInLine(emailpattern);
            if (match1 == null) {
                  JOptionPane.showMessageDialog(this, "Invalid
EmailId..Please Enter Another Email");
                  t3.setText("");
                  // temailid.setBackground(Color.RED);
                  emailvalid = false;
            }
            return emailvalid;
      }
      public boolean mobnocheck() {
            boolean mobilenovalid = true;
            String mobnopattern = "^[0-9]{10}$";
            @SuppressWarnings("resource")
            Scanner sc3 = new Scanner(t5.getText().trim());
            String match2 = sc3.findInLine(mobnopattern);
            if (match2 == null) {
                  JOptionPane.showMessageDialog(this, "Invalid
Mobno..Please Enter Another MobNo");
```

```
                t5.setText("");                 mobilenovalid = false;
            }
            return mobilenovalid;
        }
        public boolean aadharcheck() {
            boolean aadharvalid = true;
            String aadharpattern = "^[0-9]{12}$";
            @SuppressWarnings("resource")
            Scanner sc3 = new Scanner(t4.getText().trim());
            String match2 = sc3.findInLine(aadharpattern);
            if (match2 == null) {
                JOptionPane.showMessageDialog(this, "Invalid
Aadhar number, It should be 12 digits");
                t4.setText("");                 aadharvalid = false;
            }
            return aadharvalid;
        }
        public boolean passcheck() {
            boolean passvalid = true;
            String passpattern = "^[a-zA-Z0-9]{5}$";
            @SuppressWarnings("resource")
            Scanner sc3 = new Scanner(t2.getText().trim());
            String match2 = sc3.findInLine(passpattern);
            if (match2 == null) {
                JOptionPane.showMessageDialog(this, "password
must be within 5 characters");
                t2.setText("");                 passvalid = false;
            }
            return passvalid;
        }
        public boolean agecheck() {
            boolean agevalid = true;
            String agepattern = "^[0-9]{2}$";
            @SuppressWarnings("resource")
            Scanner sc3 = new Scanner(t7.getText().trim());
            String match2 = sc3.findInLine(agepattern);
            if (match2 == null) {
```

```
                    JOptionPane.showMessageDialog(this, "password
must be within 5 characters");
                    t7.setText("");
                    agevalid = false;
                }
                return agevalid;
        }
        public boolean getinfo() {boolean found = false;
try {
                    for (int i = 0; i < emplist.size(); i++) {
                        r = emplist.get(i);
                        if (t4.getText().equals(emplist.get(i).getId())) {
                            found = true;
                            break;
                        }
                    }
                } catch (Exception e) {
                    JOptionPane.showMessageDialog(this, "Okay..!!");}
                if (found)
                    return true;
                else
                    return false;}}
```

Add Room

```
public class Add extends JFrame {
private static final long serialVersionUID = 1L;
                private JButton signupb,clearb,dispb;
                private JPanel sign;
                private JTextField t1,t2,t3,t4,t5;
                private JLabel l0,l1,l2,l3,l4,l5;
                private ArrayList<room_info> rlist=new
ArrayList<room_info>();
                room_info rx;
                Add()
                {
                super("Add Here");
```

```
Font f=new Font("Lucida Sans",Font.BOLD,25);Font f2=new
Font("Lucida Sans",Font.BOLD,22);
                    sign = new JPanel();
                    sign.setLayout(null);
                    setSize(450, 350);
                    sign.setBackground(new Color(0, 0, 0, 50));
                    sign.setBounds(350,150, 500, 750);
                  l0=new JLabel("Add Room");
                  l0.setBounds(210,0,250,50);
                  l0.setFont(f2);
                  l0.setForeground(Color.WHITE);
                    l1=new JLabel("Room No:");
                t1=new JTextField();
                l1.setFont(f2);
                l1.setForeground(Color.white);
                l1.setBounds(47, 45,400, 42);
                t1.setFont(f);
                t1.setBounds(50, 86, 400, 45);
                l2=new JLabel("Ac(ac) Nonac(na):");
                t2=new JTextField();
                l2.setFont(f2);
                l2.setForeground(Color.white);
                l2.setBounds(47, 145,400, 42);
                t2.setFont(f);
                t2.setBounds(50, 186, 400, 45);
               l3=new JLabel("Bed No :");
                t3=new JTextField();
                l3.setFont(f2);
                l3.setForeground(Color.white);
                l3.setBounds(47, 245,400, 42);
                t3.setFont(f);
                t3.setBounds(50, 286, 400, 45);
                l4=new JLabel("Room Type(dlx/cmn) :");
                t4=new JTextField();
                l4.setFont(f2);
                l4.setForeground(Color.white);
                l4.setBounds(47, 345,400, 42);
```

```
t4.setFont(f);
t4.setBounds(50, 386, 400, 45);
l5=new JLabel("Price :");
t5=new JTextField();
l5.setFont(f2);
l5.setForeground(Color.white);
l5.setBounds(47, 445,400, 42);
t5.setFont(f);
t5.setBounds(50, 486, 400, 45);
sign.add(l0);
sign.add(l1);
sign.add(t1);
sign.add(l2);
sign.add(t2);
sign.add(l3);
sign.add(t3);
sign.add(l4);
sign.add(t4);
sign.add(l5);
sign.add(t5);
signupb=new JButton("Add");
signupb.setForeground(Color.black);
signupb.setBackground(new Color(134, 121, 121));
signupb.setFont(f);
 clearb=new JButton("Reset");
clearb.setForeground(Color.black);
clearb.setBackground(new Color(134, 121, 121));
clearb.setFont(f);
  dispb=new JButton("Display");
dispb.setForeground(Color.black);
dispb.setBackground(new Color(134, 121, 121));
dispb.setFont(f);
signupb.setBounds(165,550,150, 50);
sign.add(signupb);
clearb.setBounds(165,600,150, 50);
sign.add(clearb);
dispb.setBounds(165,650,150, 50);
```

```
            sign.add(dispb);
                    signupb.addActionListener((e)->
            {
                            add();
                           });
                    clearb.addActionListener((e)->
            {
             this.dispose();
               new Add();
            });
            dispb.addActionListener((e)->
            {
             new disproom();
                 });
             JLabel bg_img=new JLabel();
           bg_img.setIcon(new
ImageIcon(signup.class.getResource("/img7.jpg")));
            bg_img.setBounds(0, 0, 1200,900);
            add(bg_img);
            bg_img.add(sign);
                            setSize(1200,900);
                    setLayout(null);
                    setLocationRelativeTo(null);
                    setResizable(false);
       setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                    setVisible(true);
             }
                            public void add()
             {
                                    String Rno,Rtype,Bno,Rcat,price;
             boolean status=false;
                            Rno=t1.getText();
             Rtype=t2.getText();
             Bno=t3.getText();
          Rcat=t4.getText();
          price=t5.getText();
```

```
                    rx=new
room_info(Rno,Rtype,Bno,Rcat,price,status);
                    rlist = Roomwrite.readObjectFromFile();
                    if(checkfields())
                    {
                rlist.add(rx);
                Roomwrite.writeObjecttoFile(rlist);
                JOptionPane.showMessageDialog(this, "Added");
                   }
                    else
                          JOptionPane.showMessageDialog(null,
"empty fields");
            }
                        public boolean checkfields()
            {

if(t1.getText().equals(null)&&t2.getText().equals(null)&&t3.getText
().equals(null)&& t4.getText().equals(null)
            &&t5.getText().equals(null))
{
      return true;
}
else
      return false;
            }
```

Display Logbook

```
public class disptable2 extends JFrame
{
            private static final long serialVersionUID = 1L;

            public disptable2()
            {

                  super("Display Booking");
```

```
                JTable datatable=new JTable();
                ArrayList<new_info> blist;
                String
heading[]={"ID","Checkin_Dt","Checkin_Dt","Room
No","Bill Amt"};
                String data[][];
                Font f=new Font("Lucida
Sans",Font.BOLD,25);
                DefaultTableModel model=new
DefaultTableModel();                try
                {
                blist=UserCheckInData.readObjectFromFile()
                    data = new String[blist.size()][12];
                    int r=0;
                    for(new_info rq : blist)
                    {
                        data[r][0]=rq.getRt();
                        data[r][1]=rq.getChkin();
                        data[r][2]=rq.getChkout();
                        data[r][3]=rq.getIntr();
                        data[r][4]=rq.getNet();
                        r++;
                    }
                    model.setDataVector(data, heading);
                    datatable.setFont(f);
                    datatable.setRowHeight(25);
                    datatable.setModel(model);
                    Container con = getContentPane();
                    con.setLayout(new BorderLayout());
                    datatable.setBounds(0, 0, 500, 850);
                    JScrollPane jsp = new
JScrollPane(datatable);con.add(new JLabel("All Registration
Details"), BorderLayout.NORTH);
```

```
con.add(jsp, BorderLayout.CENTER);
setSize(850, 500);
setLocation(200, 200);
setVisible(true);
}catch(Exception e)
{
e.printStackTrace();
}}}
```

Display User Info

```
public class display extends JFrame
{
	private static final long serialVersionUID = 1L;

	public display()
	{
	super("Display All");
		JTable datatable=new JTable();
		ArrayList<Guest_Info> blist;
String heading[]={"ID","Name","Age","Email Id","Password","MobNo","Address","Gender"};
		String data[][];
		Font f=new Font("Lucida Sans",Font.BOLD,25);
		DefaultTableModel model=new DefaultTableModel();

		try
		{
			blist = UserInfoReadWriteFromFile.readObjectFromFile();

			data = new String[blist.size()][12];


			int r=0;
			for(Guest_Info re : blist)
```

```
                {
                        data[r][0]=re.getId();
                        data[r][1]=re.getName();
                        data[r][2]=re.getAge();
                        data[r][3]=re.getEmailid();
                        data[r][4]=re.getPassword();
                        data[r][5]=re.getMobno();
                        data[r][6]=re.getAddress();
                        data[r][7]=re.getGender();
                        r++;
                }
                model.setDataVector(data, heading);
                datatable.setFont(f);
                datatable.setRowHeight(25);
                datatable.setModel(model);
                Container con = getContentPane();
                con.setLayout(new BorderLayout());
                datatable.setBounds(0, 0, 500, 850);
                JScrollPane jsp = new JScrollPane(datatable);
                con.add(new JLabel("All Registration Details"),
BorderLayout.NORTH);
                con.add(jsp, BorderLayout.CENTER);
                setSize(850, 500);
                setLocation(200, 200);
                setVisible(true);
            }catch(Exception e)
            {
                e.printStackTrace();
            }}}
```

Delete Room

```
public class delete extends JFrame {
    private static final long serialVersionUID = 1L;
    private ArrayList<room_info> rlist;
    private JPanel sign;
    private JLabel l1;
```

```
        private JTextField t1;
        private JButton searchb;
        private int found = -1;
        public delete() {
                super("delete Here");
                Font f = new Font("Lucida Sans", Font.BOLD, 25);
                Font f2 = new Font("Lucida Sans", Font.BOLD, 22);
                sign = new JPanel();
                sign.setLayout(null);
                setSize(450, 350);
                sign.setBackground(new Color(0, 0, 0, 50));
                sign.setBounds(350, 250, 450, 350);
                l1 = new JLabel("Enter room no");
                t1 = new JTextField();
                l1.setFont(f2);
                l1.setForeground(Color.white);
                l1.setBounds(65, 60, 400, 48);
                t1.setFont(f);
                t1.setBounds(65, 120, 300, 50);
                searchb = new JButton("delete");
                searchb.setForeground(Color.black);
                searchb.setBackground(new Color(134, 121, 121));
searchb.setFont(f);
                searchb.setBounds(150, 200, 150, 50);
                searchb.addActionListener((e) -> {
            rlist=Roomwrite.readObjectFromFile();
                        for (int i = 0; i <rlist.size(); i++) {
                                if
(t1.getText().trim().equals(rlist.get(i).getRno()))
                                {
                                        found = i;
                                        break;
                                }
                        }
                        if (found == -1) {
                        JOptionPane.showMessageDialog(this, "Room
deleted");
```

```
                    new admin_login();
                } else {
                    deleteInfoData(found);
                JOptionPane.showMessageDialog(this, "Data
Deleted");
                    if (found == -1) {
                JOptionPane.showMessageDialog(this, "Id is not
available");
                            new admin_login();
            }}});
            sign.add(l1);
            sign.add(t1);
            sign.add(searchb);
            JLabel bg_img = new JLabel();
            bg_img.setIcon(new
ImageIcon(delete.class.getResource("/img4.jpg")));
            bg_img.setBounds(0, 0, 1200, 900);
            add(bg_img);
            bg_img.add(sign);
            setSize(1200, 900);
            setLayout(null);
            setLocationRelativeTo(null);
            setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            setVisible(true);}
        public void deleteInfoData(int index) {
            ArrayList<room_info> rlist;
            rlist=Roomwrite.readObjectFromFile();
            if (index > 0) {
                rlist.remove(index);
              Roomwrite.writeObjecttoFile(rlist);
            } else
                JOptionPane.showMessageDialog(this, "Id is not
available");
            Roomwrite.writeObjecttoFile(rlist);}}
```

Display room

```
public class disproom extends JFrame
{
	private static final long serialVersionUID = 1L;
		disproom()
	{
	super("Display All");
	Font f=new Font("Lucida Sans",Font.BOLD,25);
	String heading[]={"Rno","Rcat","bed
no","type","price","Status"};
	String data[][];
	ArrayList<room_info> list;
	try
	{
		list = Roomwrite.readObjectFromFile();
				data = new String[list.size()][12];
int r=0;
		for(room_info ra : list)
		{
			data[r][0]=ra.getRno();
			data[r][1]=ra.getRcat();
			data[r][2]=ra.getBno();
			data[r][3]=ra.getRtype();
			data[r][4]=ra.getPrice();
			data[r][5]=Boolean.toString(ra.isStatus());
		}
		Container con=getContentPane();
		con.setLayout(new BorderLayout());
		JTable datatable=new JTable(data, heading);
		datatable.setRowHeight(50);
		datatable.setFont(f);
		JScrollPane jsp=new JScrollPane(datatable);
		con.add(new JLabel("All Registration
Details"),BorderLayout.NORTH);
		con.add(jsp,BorderLayout.CENTER);
		setSize(850, 300);
		setLocation(200, 200);setVisible(true);
```

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

Search Room

```
public class SearchDisp extends JFrame
{
        private static final long serialVersionUID = 1L;

        public SearchDisp(int index)
        {
                super("Search Window");
                String heading[]={"ID","Name","Age","Email
Id","MobNo","Address","Gender"
                                ,"Cindate","Coutdate"};
                String data[][] = new String[12][12];
                ArrayList<Guest_Info> emplist;
                ArrayList<new_info> blist;
                Font f=new Font("Lucida Sans",Font.BOLD,25);
                JTable datatable=new JTable();
                DefaultTableModel model=new DefaultTableModel();
                try
                {
                        emplist=
UserInfoReadWriteFromFile.readObjectFromFile();
                        blist=UserCheckInData.readObjectFromFile();

                        Guest_Info re = emplist.get(index);
                        new_info rx=blist.get(index);

                        int r = 0;

                    data[r][0]=re.getId();
                        data[r][1]=re.getName();
                        data[r][2]=re.getAge();
                        data[r][3]=re.getEmailid();
                        data[r][4]=re.getMobno();
```

```
                    data[r][5]=re.getAddress();
                    data[r][6]=re.getGender();
                    data[r][7]=rx.getChkin();
                    data[r][8]=rx.getChkout();



                    model.setDataVector(data, heading);
                    datatable.setFont(f);
                    datatable.setRowHeight(25);
                    datatable.setModel(model);

                    Container con = getContentPane();
                    con.setLayout(new BorderLayout());
                    datatable.setBounds(0, 0, 500, 850);
                    JScrollPane jsp = new JScrollPane(datatable);

                    con.add(new JLabel("All Registration Details"),
BorderLayout.NORTH);
                    con.add(jsp, BorderLayout.CENTER);

                    setSize(850, 300);
                    setLocation(200, 200);

        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                    setVisible(true);

                }catch(Exception e)
                {
                    JOptionPane.showMessageDialog(this,"No Booking
from this id");
                }

        }
}
```

Guest info model class

```
public class Guest_Info implements Serializable {
	private static final long serialVersionUID = 1L;
	private String id;
	private String password;
	private String name;
	private String emailid;
	private String mobno;
	private String address;
	private String gender;
	private String age;
	public Guest_Info(String id, String password, String name,
String emailid, String mobno, String address,
			String gender, String age) {
		super();
		this.id = id;
		this.password = password;
		this.name = name;
		this.emailid = emailid;
		this.mobno = mobno;
		this.address = address;
		this.gender = gender;
		this.age = age;
	}
	public void setAge(String age) {
		this.age = age;
	}
	public String getId() {
		return id;
	}
	public void setId(String id) {
		this.id = id;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
```

```
        this.password = password;
}
public String getName() {
        return name;
}
public void setName(String name) {
        this.name = name;
}
public String getEmailid() {
        return emailid;
}
public void setEmailid(String emailid) {
        this.emailid = emailid;
}
public String getMobno() {
        return mobno;
}
public void setMobno(String mobno) {
        this.mobno = mobno;
}
public String getAddress() {
        return address;
}
public void setAddress(String address) {
        this.address = address;}
        public String getGender() {
        return gender;}
public void setGender(String gender) {
        this.gender = gender;
}
public String getAge() {
        return age;
}
public void setDob(String age) {
        this.age = age;
}}
```

Room Info Model Class

```
public class room_info implements Serializable{
  private static final long serialVersionUID = 1L;
        private String Rno;
        private String Rtype;
        private String Bno;
        private String Rcat;
        private String Price;
        private boolean status;
        public  room_info(String Rno,String Rtype,String Bno,String
Rcat,String Price,boolean status)
        {
                super();
                this.Rno = Rno;
       this.Rtype=Rtype;
                this.Bno=Bno;
                this.Rcat=Rcat;
                this.Price=Price;
                this.status=status;
        }
                public String getRno() {
                return Rno;
        }
        public void setRno(String rno) {
                Rno = rno;
        }
        public String getRtype() {
                return Rtype;
        }
        public void setRtype(String rtype) {
                Rtype = rtype;
        }
        public String getBno() {
                return Bno;
        }
        public void setBno(String bno) {
                Bno = bno;
```

```
}
public String getRcat() {
        return Rcat;
}
public void setRcat(String rcat) {
        Rcat = rcat;
}
public String getPrice() {
        return Price;
}
public void setPrice(String price) {
        Price = price;
}
public boolean isStatus() {
        return status;
}
public void setStatus(boolean status) {
        this.status = status;
}}
```

New info Model Class

```
public class new_info implements Serializable {
        private static final long serialVersionUID = 1L;
        private String password;
        private String chkin;
        private String chkout;
        private String net;
        private String intr;
        public String rt;
public  new_info(String chkin,String chkout,String
net,String intr,String rt )
        {
                super();
                        this.intr=intr;
                this.chkin=chkin;
                this.chkout=chkout;
                this.net=net;
```

```
        this.rt=rt;

    }
    public String getChkin() {
        return chkin;
    }
    public void setChkin(String chkin) {
        this.chkin = chkin;
    }
    public String getChkout() {
        return chkout;
    }
    public void setChkout(String chkout) {
        this.chkout = chkout;
    }
    public String getNet() {
        return net;
    }
    public void setNet(String net) {
        this.net = net;
    }
     public String getIntr() {
            return intr;
        }
        public void setIntr(String intr) {
            this.intr =intr;
        }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getRt() {
        return rt;
    }
```

```
        public void setRt(String rt) {
            this.rt = rt;
        }
    }
```

Room write file class

```
package Pack1;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
public class Roomwrite
{
    @SuppressWarnings("unchecked")
    public static ArrayList<room_info> readObjectFromFile()
    {
        ArrayList<room_info> userlistfromfile;
        try
        {
          FileInputStream fin=new FileInputStream("room.txt");
          ObjectInputStream oin=new ObjectInputStream(fin);
      userlistfromfile=(ArrayList<room_info>)oin.readObject();
          oin.close();
          fin.close();
        }catch(Exception e)
         {
            userlistfromfile=new ArrayList<room_info>();
         }
        return userlistfromfile;
    }
    public static void writeObjecttoFile(ArrayList<room_info>
ulist)
    {
        try
         {
```

```
                    FileOutputStream fout=new
FileOutputStream("room.txt");
                    ObjectOutputStream oout=new
ObjectOutputStream(fout);
                    oout.writeObject(ulist);
                    oout.close();
                }catch(Exception e){System.out.println(e);}
        }
}
```

Guest_write file class

```
public class UserInfoReadWriteFromFile
{
        @SuppressWarnings("unchecked")
        public static ArrayList<Guest_Info> readObjectFromFile()
        {
                ArrayList<Guest_Info> userlistfromfile;
                try
                {
                  FileInputStream fin=new
FileInputStream("database.txt");
                  ObjectInputStream oin=new ObjectInputStream(fin);

userlistfromfile=(ArrayList<Guest_Info>)oin.readObject();
        oin.close();
                  fin.close();
                }catch(Exception e)
                 {
                        userlistfromfile=new ArrayList<Guest_Info>();
                 }
                        return userlistfromfile;
        }
                public static void
writeObjecttoFile(ArrayList<Guest_Info> ulist)
        {
                try
                 {
    FileOutputStream fout=new FileOutputStream("database.txt");
```

```
        ObjectOutputStream oout=new ObjectOutputStream(fout);
                oout.writeObject(ulist);
                        oout.close();
                }catch(Exception e){System.out.println(e);}
        }
}
```

New Info write file class

```
public class UserCheckInData {
                @SuppressWarnings("unchecked")
                public static ArrayList<new_info> readObjectFromFile()
                {
                        ArrayList<new_info> listfromfile;
                        try
                        {
FileInputStream fin=new FileInputStream("checkin.txt");
  ObjectInputStream oin=new ObjectInputStream(fin);
                  listfromfile=(ArrayList<new_info>)oin.readObject();
                          oin.close();
                          fin.close();
                        }catch(Exception e)
                         {
                                listfromfile=new ArrayList<new_info>();
                         }
                        return listfromfile;
                }
                public static void
writeObjecttoFile(ArrayList<new_info> ulist)
                {
                        try
                         {
  FileOutputStream fout=new FileOutputStream("checkin.txt");
ObjectOutputStream oout=new ObjectOutputStream(fout);
                              oout.writeObject(ulist);
                              oout.close();
                        }catch(Exception e){System.out.println(e);}
                }}
```

Display Table

```
public class disproom extends JFrame
{
	private static final long serialVersionUID = 1L;

	disproom ()
	{
	Super ("Display All");

	Font f=new Font ("Lucida Sans",Font.BOLD,25);
	String heading[]={"Rno","Rcat","bed
no","type","price","Status"};
	String data [][];
	ArrayList<room_info> list;
	try
	{
		list = Roomwrite.readObjectFromFile();

		data = new String[list.size()][12];

		int r=0;
		for(room_info ra : list)
		{
			data[r][0]=ra.getRno();
			data[r][1]=ra.getRcat();
			data[r][2]=ra.getBno();
			data[r][3]=ra.getRtype();
			data[r][4]=ra.getPrice();
			data[r][5]=Boolean.toString(ra.isStatus());
			r++;
		}
```

```
            Container con=getContentPane ();
            con.setLayout (new BorderLayout ());
            JTable datatable=new JTable (data, heading);
            datatable.setRowHeight (50);

datatable.setFont (f);
JScrollPane jsp=new JScrollPane (datatable);
con.add (new JLabel ("All Registration Details"),
BorderLayout.NORTH);
            con.add (jsp,BorderLayout.CENTER);

            setSize (850, 300);
            setLocation (200, 200);
            setVisible (true);
      }catch(Exception e)
      {
            e.printStackTrace();
      }
      }
}
```

Certificate

This is to certify that **Mr. Nikhil Singh** of **BENGAL COLLEGE OF ENGINEERING AND TECHNOLOGY, DURGAPUR**, registration number: **171250110064**, , has successfully completed a project on **HOTEL MANAGEMENT SYSTEM** using Java 11 with Swing under the guidance of **MR. CHANDAN MUKHERJEE.**

Chandan Mukherjee

Globsyn Finishing School

(A division of Globsyn Skills)

Certificate

This is to certify that **Mr. Ashutosh Kumar** of **BENGAL COLLEGE OF ENGINEERING AND TECHNOLOGY, DURGAPUR**, registration number: **171250110026**, , has successfully completed a project on **HOTEL MANAGEMENT SYSTEM** using Java 11 with Swing under the guidance of **MR. CHANDAN MUKHERJEE.**

Chandan Mukherjee

Globsyn Finishing School

(A division of Globsyn Skills)

Certificate

This is to certify that **Ms. Isha Rani** of **BENGAL COLLEGE OF ENGINEERING AND TECHNOLOGY, DURGAPUR**, registration number: **171250110047**, , has successfully completed a project on **HOTEL MANAGEMENT SYSTEM** using Java 11 with Swing under the guidance of **MR. CHANDAN MUKHERJEE.**

Chandan Mukherjee

Globsyn Finishing School

(A division of Globsyn Skills)

Certificate

This is to certify that **Ms. Kumari Shruti** of **BENGAL COLLEGE OF ENGINEERING AND TECHNOLOGY, DURGAPUR**, registration number: **171250110052**, , has successfully completed a project on **HOTEL MANAGEMENT SYSTEM** using Java 11 with Swing under the guidance of **MR. CHANDAN MUKHERJEE.**

Chandan Mukherjee

Globsyn Finishing School

(A division of Globsyn Skills)