

# Hotel Management System

S.E. mini-project report submitted in partial fulfillment

of the requirements of the degree of

Bachelor of Engineering

By

*Swarali More*  
*Ghanshyam Patel*  
*Hrutvik Rane*  
*Maitrey Phatak*

Supervisor:

Prof. Suchetadevi Gaikwad.



Department of Information Technology

Atharva College of Engineering

Year: 2020-21



# **ATHARVA COLLEGE OF ENGINEERING**

**MALAD (W), MUMBAI 400 095**

**YEAR: 2020-2021**

## **CERTIFICATE**

*This is to certify that*

**Swarali More**

*have submitted the Mini Project report for the requirements of the Bachelor of  
Engineering in Information Technology satisfactorily*

*on*

**“Hotel Management System”**

*As prescribed by the **University of Mumbai** Under the guidance of*

**PROJECT GUIDE**  
**Prof. Suchetadevi Gaikwad**

**H.O.D.**  
**Prof. Deepali Maste**

**PRINCIPAL**  
**Dr. Srikant Kalurkar**

**INTERNAL EXAMINER**

**COLLEGE SEAL**

**EXTERNAL EXAMINER**

# Approval for S. E. Mini Project Report

This project report entitled (*Hotel Management System*) by (*Swarali More*) is approved for the degree of *Bachelor of Engineering in Information Technology*

## Examiners:

(Internal Examiner Name & Sign) • -----

(External Examiner Name & Sign) • -----

Date: 08/12/2020

Place: Mumbai

# **Contents**

**Abstract**

## **1. Introduction**

**1.1 Introduction**

**1.2 Problem Statement**

**1.3 Objectives**

## **2. Literature Survey**

**2.1 Survey of Existing system**

**2.2 Limitation Existing system or research gap**

**2.3 Mini Project contribution**

## **3. Proposed System**

**3.1 Architecture/Framework**

**3.2 Software Requirement Specification**

**3.3 Implementation Code(write classes as heading)**

**3.4 Output Screenshots(with title of each output)**

## **4. Future work**

## **5. Conclusion**

**References**

# Abstract

- This application is specially developed to help hotel staff. The project Hotel Management manages and maintains the records of customers and rooms in the hotel. The rooms have different categories including single bed and multiple beds. 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 the entries of customers. • As a security we have provided Admin & Employee level authentication for different modules such as set-up employee etc.
- For the convenience of Administration, we have provided facility to generate report of room details, employee details, customer details and department details.

# Introduction

- Everyday humans achieving are a new level of programming zenith.
- With the increase in technological advances, it has now become more than important to update pre-existing systems to be able to stay safe and for smooth user experience.

## Problem statement

- Hotels are complex businesses, filled with tedious requirements and huge data filings. The traditional system of maintaining records in respective registers is now getting outdated with each increasing second. For optimum usage it is now become crucial to attain java database-connectivity in hotel management system.

# Objectives

- Enable the user to look up details of employee.
- Enable the user to look up the details of room status and department info.
- Allow the manager of the hotel to change the hotel information from time to time.

## Literature Survey

Existing Problem Faced	Survey Of Existing System
The existing system is manual system	To overcome the problems of manual system, online hotel management system is proposed
Data was present in different files which led to redundancy and inadequate data handling.	The Central objective of Online Hotel Management is smooth handling of data present in the system.
The hotel management has to keep records of rooms manually	This software can manage and keep records of hotel.

ATHARVA COLLEGE OF ENGINEERING 7

Sr No	Author	Paper	Content
-------	--------	-------	---------

1)	Muhammad Abid Nazir	Study of Development of Java Applications in Eclipse Environment and Development of Java Based Calendar Application with Email Notifications	Focuses on GUI in Java with the help of Java
2)	a) S G Ganesh b) Tushar Sharma	Building Database Applications with JDBC	Focuses on java database connectivity between Eclipse and MySQL.

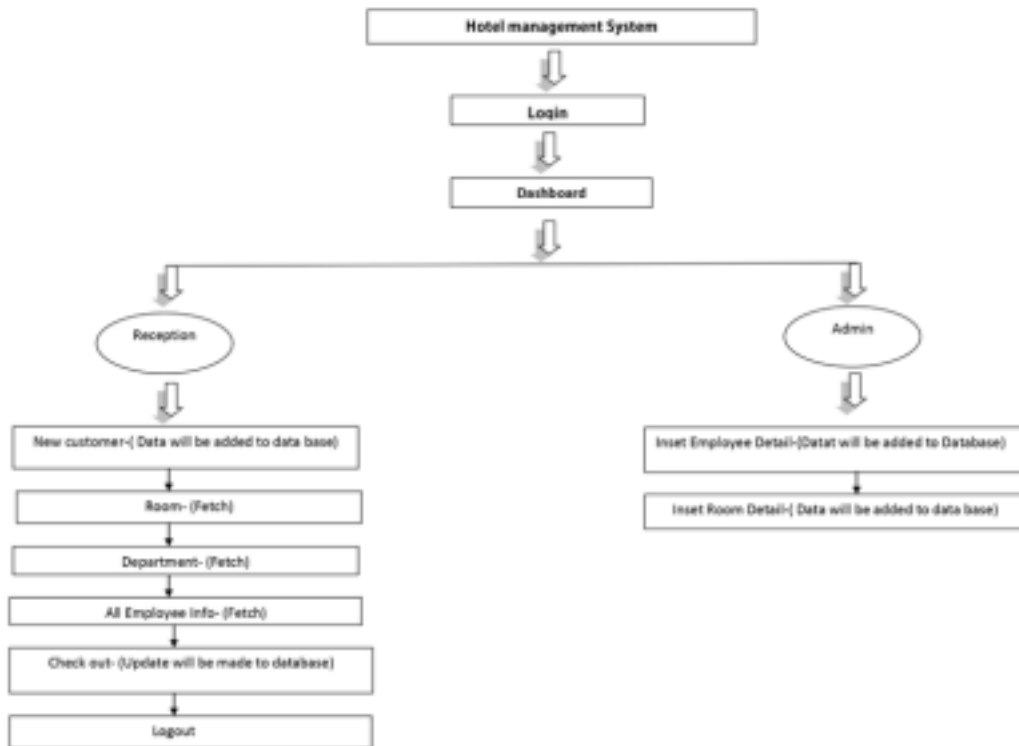
## Limitation Existing system

- It is a time consuming process.
- There is no surety of availability of rooms.
- Paper work results in need of lot of space to keep the data.
- Lack of security.
- Chances of human errors.



# Proposed System:

## Architecture/Framework



## Software Requirement Specification

- Language using: Java, MySQL
- Software using: Eclipse
- Database using: MySQL

# CODE:

## SPLASH:

```
package hotelmanagementsystem;

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

public class Splash extends JFrame
{

    Splash()
    {

    }

    public static void main(String[] args)
    {
        Splash2 f1 = new Splash2();
        f1.getContentPane().setBounds(0,0,0,0);

        ImageIcon i1 = new ImageIcon(("hotell/hotel.png.png"));
        Image i3 =
i1.getImage().getScaledInstance(1156,646,Image.SCALE_DEFAULT);
        ImageIcon i2 = new ImageIcon(i3);
        JLabel l1 = new JLabel(i2);

        JLabel j2 = new JLabel("Hotel Management
        System"); j2.setBounds(20,10,650,70);
        j2.setForeground(Color.black);
```

```

j2.setFont(new Font ("Bradley Hand ITC",Font.BOLD,50));
j2.setBorder(BorderFactory.createLineBorder(Color.yellow, 3));
l1.add(j2);

JButton b1 = new JButton("Enter");
b1.setBackground(Color.DARK_GRAY);
b1.setForeground(Color.orange);
b1.setBounds(900,530,140,40);
b1.addActionListener(new ActionListener()
{ public void actionPerformed(ActionEvent e)
    {
        try{
            login bd = new login();
            bd.setVisible(true);
            f1.setVisible(false);
        }
        catch (Exception e1){
            e1.printStackTrace();
        }
    }
});

l1.add(b1);

```

```

f1.getContentPane().add(l1);
f1.setVisible(true);
int i;
int x=1;
for(i=2; i<=780; i+=2.5, x+=1)

```

```

{

f1.setLocation(680 - ((i+x)/2), 400 - (i/2));
f1.setSize(i+x,i);
try{
Thread.sleep(10);
}catch(Exception e){}
}
while(true)
    {
        j2.setVisible(false);
        try
        {
            Thread.sleep(300);
        }
        catch(Exception e)
        {

        }
        j2.setVisible(true);
        try
        {
            Thread.sleep(300);
        }
        catch(Exception e)
        {

        }
    }
}
}
class Splash2 extends JFrame implements
Runnable{ Thread t1;

```

```

Splash2()
{

super("");
setLayout(new FlowLayout());
t1 = new Thread(this);
t1.start();
}

public void run(){
try{
Thread.sleep(5000);
//this.setVisible(false);

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

}

}

```

## Dashboard:

```

package hotelmanagementsystem;
import java.awt.*;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

public class Dashboard extends JFrame implements ActionListener
{
    JMenuBar mb;

```

```
JMenu m1,m2;
```

```
JMenuItem i1,i2,i3;
```

```
Dashboard()
```

```
{
```

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

```
    add(mb);
```

```
    m1 = new JMenu("Hotel");
```

```
    m1.setForeground(Color.black);
```

```
    m1.setFont(new Font("Berlin Sans FB", Font.PLAIN,  
    15)); mb.add(m1);
```

```
    m2 = new JMenu("Admin");
```

```
    m2.setForeground(Color.black);
```

```
    m2.setFont(new Font("Berlin Sans FB", Font.PLAIN,  
    15)); mb.add(m2);
```

```
    i1 = new JMenuItem("Reception");
```

```
    i1.setFont(new Font("Berlin Sans FB", Font.PLAIN,  
    15)); i1.addActionListener(this);
```

```
    m1.add(i1);
```

```
    i2 = new JMenuItem("Insert Employee details");
```

```
    i2.setFont(new Font("Berlin Sans FB", Font.PLAIN,  
    15)); i2.addActionListener(this);
```

```
    m2.add(i2);
```

```
    i3 = new JMenuItem("Insert room details");
```

```
    i3.setFont(new Font("Berlin Sans FB", Font.PLAIN,  
    15)); i3.addActionListener(this);
```

```
m2.add(i3);
```

```
mb.setBounds(0,0,1280,30);
```

```
ImageIcon il = new ImageIcon("hotell/paris.png");
```

```
JLabel l1 = new JLabel(il);
```

```
l1.setBounds(0,0,1280,870);
```

```
add(l1);
```

```
JLabel l2 = new JLabel("Four Seasons");
```

```
l2.setFont(new Font("Bradley Hand ITC", Font.PLAIN,  
68)); l2.setForeground(Color.WHITE);
```

```
l2.setBounds(480,60,400,200);
```

```
l1.add(l2);
```

```
setLayout(null);
```

```
setBounds(0,0,1280,900);
```

```
setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent ae)
```

```
{
```

```
    if(ae.getActionCommand().equals("Reception"))
```

```
    {
```

```
        new Reception().setVisible(true);
```

```
    }
```

```

        if(ae.getActionCommand().equals("Insert Employee
        details")) {
            new InsertEmployee().setVisible(true);
        }

        if(ae.getActionCommand().equals("Insert room
        details")) {
            new InsertRoom().setVisible(true);
        }

    }

    public static void main (String [] args)
    {
        new Dashboard().setVisible(true);
    }
}

```

## Reception:

```

package hotelmanagementsystem;

import javax.swing.*;

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

public class Reception extends JFrame {

    private JPanel contentPane;

```



```

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

    public Reception()
    {

setBounds(230, 100, 850, 570);

        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);

        ImageIcon i1 = new ImageIcon(("hotell/reception.jpeg"));
        Image i3 = i1.getImage().getScaledInstance(848, 698, Image.SCALE_SMOOTH);
        ImageIcon i2 = new ImageIcon(i3);
        JLabel l1 = new JLabel(i2);
        l1.setBounds(0, 0, 850, 570);
        add(l1);

        JButton btnNewCustomerForm = new JButton("New Customer Form");
        btnNewCustomerForm.setFont(new Font("Berlin Sans FB", Font.BOLD, 15));
        btnNewCustomerForm.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try{
                    NewCustomer custom = new NewCustomer();
                    custom.setVisible(true);

setVisible(false);

                } catch (Exception e1) {
                    e1.printStackTrace();
                }
            }
        });
        btnNewCustomerForm.setBounds(10, 30, 201, 30);
    }

```

```
btnNewCustomerForm.setBackground(Color.BLACK);
btnNewCustomerForm.setContentAreaFilled(false);
btnNewCustomerForm.setOpaque(false);
btnNewCustomerForm.setForeground(Color.WHITE);
l1.add(btnNewCustomerForm);
```

```
        JButton btnNewButton = new JButton("Room");
        btnNewButton.setFont(new Font("Berlin Sans FB", Font.BOLD,
        15)); btnNewButton.setContentAreaFilled(false);
btnNewButton.setOpaque(false);
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                try{
                    Room room = new Room();
                    room.setVisible(true);
setVisible(false);

                }
                catch(Exception e){
                    e.printStackTrace();
                }
            }
        });
        btnNewButton.setBounds(10, 70, 200, 30);
btnNewButton.setBackground(Color.BLACK);
btnNewButton.setForeground(Color.WHITE);

        l1.add(btnNewButton);

        JButton btnNewButton_1 = new JButton("Department");
        btnNewButton_1.setFont(new Font("Berlin Sans FB", Font.BOLD,
        15)); btnNewButton_1.setContentAreaFilled(false);
        btnNewButton_1.setOpaque(false);
```

```

btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try{
            Department dept = new Department();
            dept.setVisible(true);

            setVisible(false);

        }
        catch (Exception e1){
            e1.printStackTrace();
        }
    }
});

```

```

btnNewButton_1.setBounds(10, 110, 200, 30);
btnNewButton_1.setBackground(Color.BLACK);
btnNewButton_1.setForeground(Color.WHITE);

```

```

l1.add(btnNewButton_1);

```

```

JButton btnNewButton_2 = new JButton("All Employee Info");
btnNewButton_2.setFont(new Font("Berlin Sans FB", Font.BOLD,
15)); btnNewButton_2.setContentAreaFilled(false);
btnNewButton_2.setOpaque(false);
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try{

            Employee em = new Employee();
            em.setVisible(true);
            setVisible(false);

```

```

        }
        catch (Exception e1){
            e1.printStackTrace();
        }

    }

});

btnNewButton_2.setBounds(10, 150, 200, 30);
btnNewButton_2.setBackground(Color.BLACK);
btnNewButton_2.setForeground(Color.WHITE);

l1.add(btnNewButton_2);

JButton btnNewButton_3 = new JButton("Customer Info");
btnNewButton_3.setFont(new Font("Berlin Sans FB", Font.BOLD,
15)); btnNewButton_3.setContentAreaFilled(false);
btnNewButton_3.setOpaque(false);
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try{
            //error -----
            CustomerInfo customer = new CustomerInfo();
            customer.setVisible(true);
            setVisible(false);
        }
        catch (Exception e1){
            e1.printStackTrace();
        }
    }
});

btnNewButton_3.setBounds(10, 190, 200, 30);

```

```
btnNewButton_3.setBackground(Color.BLACK);  
btnNewButton_3.setForeground(Color.WHITE);
```

```
l1.add(btnNewButton_3);
```

```
  
JButton btnNewButton_4 = new JButton("Check Out");  
btnNewButton_4.setFont(new Font("Berlin Sans FB", Font.BOLD,  
15)); btnNewButton_4.setContentAreaFilled(false);  
btnNewButton_4.setOpaque(false);  
btnNewButton_4.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e)  
    {  
        Checkout check;  
        try {  
            check = new Checkout();  
            check.setVisible(true);  
setVisible(false);  
        } catch (Exception e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        }  
    }  
});  
btnNewButton_4.setBounds(10, 230, 200, 30);  
btnNewButton_4.setBackground(Color.BLACK);  
btnNewButton_4.setForeground(Color.WHITE);  
  
l1.add(btnNewButton_4);
```

```

        JButton btnNewButton_5 = new JButton("Log Out");
        btnNewButton_5.setFont(new Font("Berlin Sans FB", Font.BOLD,
        15)); btnNewButton_5.setContentAreaFilled(false);
        btnNewButton_5.setOpaque(false);
        btnNewButton_5.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                try {
new login().setVisible(true);

setVisible(false);

                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
        btnNewButton_5.setBounds(10, 270, 200, 30);
        btnNewButton_5.setBackground(Color.BLACK);
        btnNewButton_5.setForeground(Color.WHITE);

        l1.add(btnNewButton_5);
        getContentPane().setBackground(Color.white);
        setVisible(true);
    }
}

```

### **Rest of the code:**

<https://drive.google.com/drive/folders/1N3QoAGHVW3zMwev2syYSgMANvqolHrgb?usp=sharing>

**Video Link of Code:**

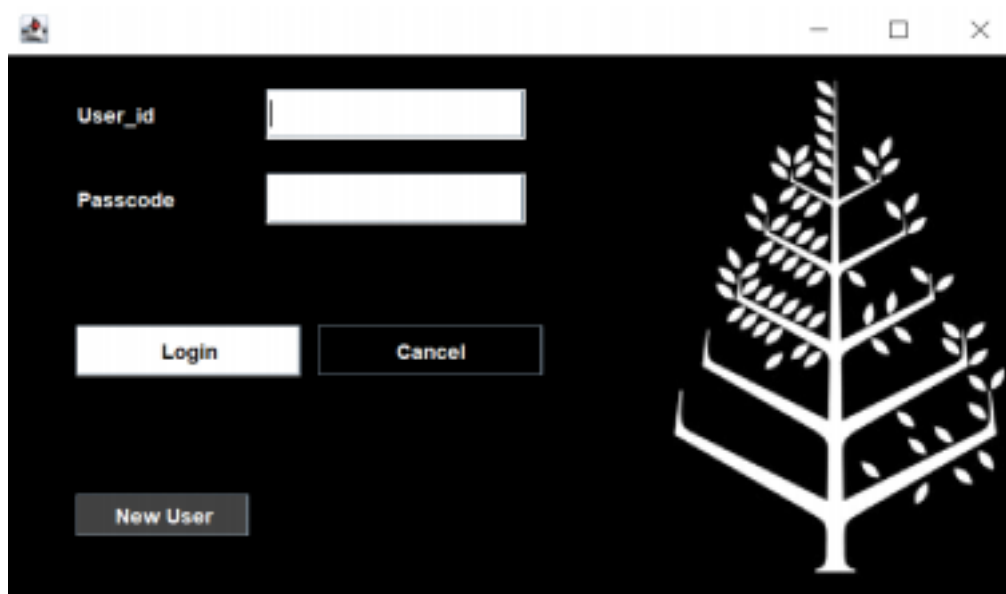
<https://drive.google.com/file/d/1tgvvUit7nOOSIbDnikHL1QA7Hk4cuZWsview?usp=sharing>

## Output Screenshots:

1) Main Screen

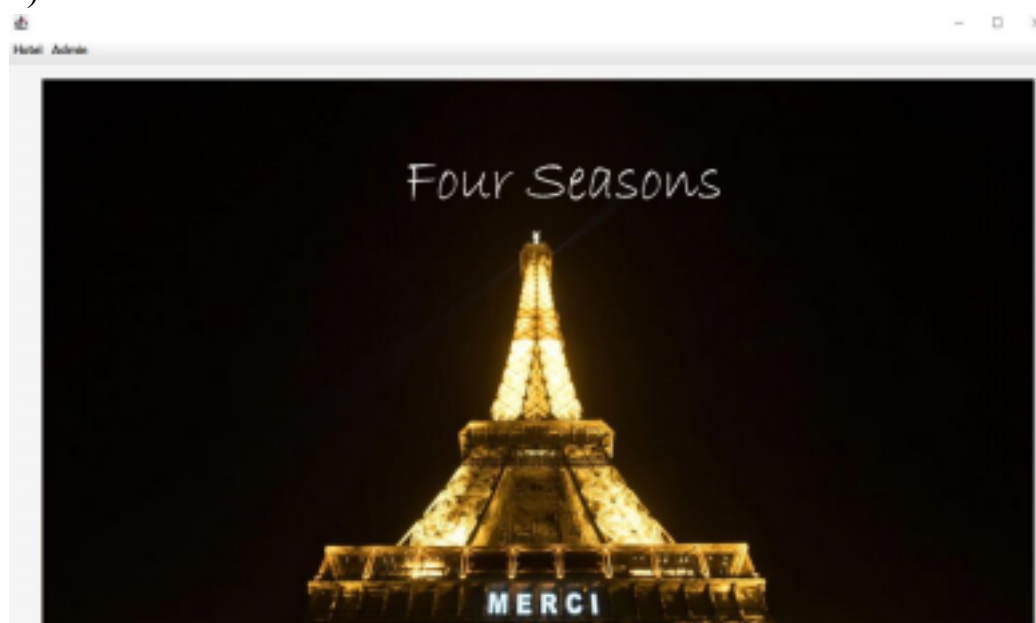


2) Login Page



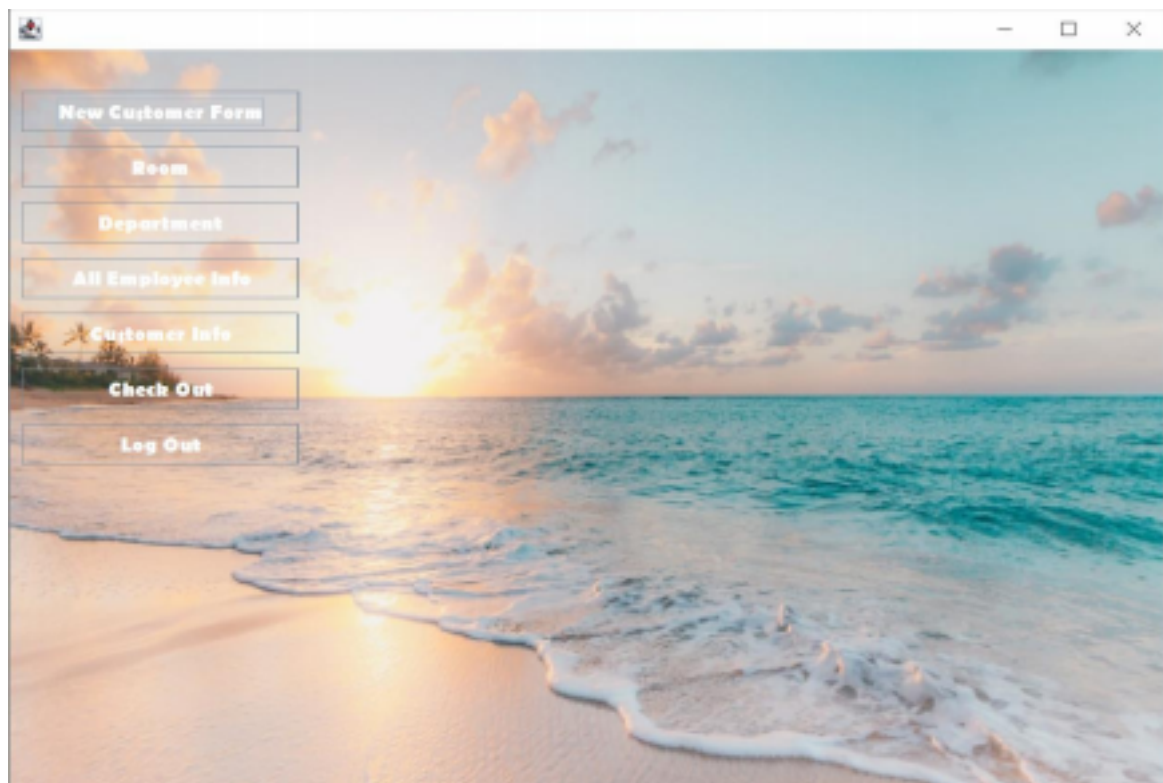
A login window with a black background. On the left, there are two white input fields labeled "User\_id" and "Passcode". Below these are two buttons: "Login" and "Cancel". At the bottom left is a "New User" button. On the right side is a large white logo of a stylized tree with many leaves. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

### 3) Dashboard



### 4) Reception





5) New User

A screenshot of a web application interface for a "New User" registration form. The form is set against a black background. It contains the following fields and controls:

- Name**: A text input field.
- Age**: A text input field.
- Gender**: Two radio buttons labeled "Male" and "Female".
- Job**: A dropdown menu.
- Salary**: A text input field.
- Mobile**: A text input field.
- Aadhar**: A text input field.
- Email**: A text input field.
- Submit**: A yellow button at the bottom.

The window has standard OS controls (minimize, maximize, close) in the top right corner.

6) New Room

**Add Rooms**

Room Number:

Availability: Available ▼

Cleaned:  ▼

Price:

Bed Type: Single Bed ▼

## 7) Employee Info

Name	Age	Gender	Job	Salary	Phone	Aadhar	Gmail
Maitrey	20	Male	Chef	800000	9874352776	567887431296	maitrey@gmail.com
Hrutvik	20	Male	Manager	800000	8907985553	234567895401	hrutvik@gmail.com
Svarali	19	Female	Manager	800000	8095032113	998774214711	svarali12@gmail.com
Gharshyam	21	Male	Accountant	800000	7898638522	78893461170	gharshyam@123
Robert	21	Male	Housekeeping	15,000	8794412311	457892462986	robert@gmail.com
Liam	22	Male	Front Desk Clerks	25,000	8765942123	876544365372	liam123@gmail.com
Tanmay	19	Male	Housekeeping	30000	9999999999	998293248	tanmay1st@gmail...
Rome	25	Male	Kitchen Staff	30,000	98478962	89439946289	rome@123
Damon	23	Male	Manager	1,00,000	84378932	8437983849813	damon@gmail.com

## 8) Department Info

Department	Budget
Front Office	1,00,000
Housekeeping	1,50,000
Kitchen and Food production	5,50,000
Maintenance	6,50,000
Accounts	3,50,000
Security	6,50,000
HR	4,50,000
IT	5,50,000

## 9) Customer Info

ID	Numb...	Name	Gender	Count...	Room	Check-in Status	Deposit
Passport	78976249	Derek	Male	Canada	101	Yes	7000
Passport	768953272	Meredith	Female	Switzerland	105	Yes	10,000
Aadhar Card	958472461461	Elena	Female	Australia	200	Yes	6000
Passport	97536235	Joseph	Male	Brazil	201	No	2000
Aadhar Card	123456789089	arya	Male	India	201	yes	4500
Passport	87403274091	Gaurangi	Female	Canada	203	yes	8500

## Future Work

- Laundry System
- Restaurant System
- Tourist package and guide System
- Meeting and conference facility

## Conclusion

- Thus, we have created a program which will lead to optimum usage of data recording and processing in the system.
- It will help to organize records in a more efficient way.

## References

- [www.wikipedia.com](http://www.wikipedia.com)
- [www.btech.com](http://www.btech.com)
- [www.w3school.com](http://www.w3school.com)
- [www.dotnetheaven.com](http://www.dotnetheaven.com)

