A

Project Report

On

HOTEL MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHMOLOGY

In

INFORMATION TECHNOLOGY

UNDER THE GUIDANCE OF

Er. NEELAM

(Project Manager)

DEPARTMENT OF INFORMATION TECHNOLOGY

Submitted By:

ADITYA KASAUDHAN (100190502)

Submitted To :-

SIR CHHOTU RAM INSTITUTE OF ENGINEERING & TECHNOLOGY

CHAUDHARY CHARAN SINGH UNIVERSITY, MEERUT

(2022-23)

ACKNOWLEDGEMENT

I would like to express my deep and sincere gratitude to my supervisor **Er. Suraj Singh** , Project Manager (**Meca-Tredz Technology ptv Ltd. Lucknow**), who gave me his full support and encouraged me to work in an innovated and challenging project for Educational field . His wide knowledge and logical thinking gave me right direction all the time .

I am deeply greatful to my project co-ordinator for his help and support provided at every step of the project . last but not the least, I thank to all employees of **Meca-Tredz Technology (P.)Ltd. Lucknow**. For their support and co-operation .

ADITYA KASAUDHAN

100190502

SIR CHHOTU RAM INSTITUTE OF ENGINEERING & TECHNOLOGY
CHAUDHARY CHARAN SINGH UNIVERSITY MEERUT
Approved by A.I.C.T.E., New Delhi

Student Declaration/Certificate

I Aditya Kasaudhan hereby declare that the work, which is being presented in the project entitled " HOTEL MANAGEMENT SYSTEM "in partial fulfillment of the requirement for the award of BACHELOF TECHNOOGY (B-TECH) degree, is an authentic record of my own work carried out under the supervision of ER. SURAJ SINGH The matter embodied in this project has not been submitted by us for the award of any other degree.

This is to certify that the above statements made by the candidate are correct to the best of my knowledge.

ADITYA KASAUDHAN
(100190502)

COMPLETION CERTIFICATE

This is to certify that **ADITYA KASAUDHAN** from **SIR CHHOTU RAM INSTITUTE OF ENGINEERING AND TECHNOLOGY** was working on the project entitled "**HOTEL MANAGEMENT SYSTEM**" in **MECA- TREDZ TECHNOLOGY .** He was engaged with us during **20 JULY** to **3 SEPTEMBER** for a period of 45 DAYS .

He has done an excellent job during his engagement with the Software Development & Testing Division of the company. He has completed his project during the training tenure. His performance has been good and satisfactory.

I would like to take this opportunity to express my appreciation to Mr. **Aditya Kasaudhan** for her work and wish him all the very best for his future endeavours .

Regards:
Er. SURAJ SINGH
Project Manager
MECA-TREDZ TECHNOLOGY
LUCKNOW (U.P.)

TABLE OF CONTENTS

ACKNOWLEDGEMENT
CERTIFICATE OF ACCEPTANCE
TABLE OF CONTENTS
LIST OF FIGURES
LIST OF TABLES
Chapter 1
INTRODUCTION
Chapter 2
OBJECTIVE
Chapter 3
MODULS
Chapter 4
SOFTWARE ENGINEERING PARADIGMS
Chapter 5
ROLE IN THE PROJECT
Chapter 6
WORKING METHODOLOGY
Chapter 7
TESTING & RESULT
Chapter 8
SCREENSHOTS
Chapter 9
CONCLUSION
Chapter 10
FUTURE SCOPE

LIST OF TABLES

- Employee
- Department
- Room
- Driver
- New customer
- Manager Info
- Customer Info
- All Employee
- Check out
- Update Status
- Update Room Status
- Search Room
- Update Room
- Logout

Chapter 1

INTRODUCT

Hotel Management System is a software system where the management of entire hotel is computerized . The hotel management system is designed using Core Java for front end and MYSQL Server as the secured backend database. In this project the details are maintained like customer details, reservation details ,Booking details and billing details The reservation process of reserving rooms for the customers , canceling the reserved rooms ,booking the rooms , vacating the rooms , billing process ,etc .all are computerized and the management is done with out any difficulty. The reports can be viewed completely and the head of the management daily or weekly or monthly can review it . For company auditing it will be more useful . This Proposed System will be interactive , faster and user-friendly for the end users . Using the hotel management system the following activities can be performed .

Chapter 2

OBJECTIVE

The main objective of to facilitate easy management and administration of a hotel with capabilities to do Booking or reservations of the rooms ,Cancellation of the rooms, Cash billing , Room service , Restaurant service, Restaurant billing , Total billing , Travels arrangement etc . using the automated hotel management software . One can keep detailed records or info on an unlimited amount of customers . The system lets the user know which all rooms are available for occupancy at any point of time . This makes the booking considerably faster . And thus helps the hotel in better management and reduce a lot of paper work as well as manpower.

Chapter 3

MODULES

- Room Enquiry and Reservation
- Check In/Check-out Module
- Bills and Payments Module
- Room Details
- Guests details
- Report

MODULE DESCRIPTION

Room Enquiry and Reservation Module

This module deals with reservation enquiry and reservation .During reservation , the details of the customers , type of room required and number of room required are fed in to the system .Once the se in formations are entered , the system search es for the unoccupied rooms and displays the result .In reservation

Enquiry , customer can get the information such as rent of rooms and details of rooms available .

Check In/Check-out Module

This module deals with the reservation check in (i.e .,allocating room for the customer who has already reserved)Direct check in(i.e., checking in with out reservation),handling check out of guest and Bill payment

Bills and Payments Module

This module deals with the generation and tracking of bills and payments made by the guest .The bills are classified in to lodging bills and Restaurant Bill .The lodging bills calculated using the check in and checkout details of the person .The Restaurant bill is generated based on the food items consumed by the guest during his stay in the hotel.

Report

This module deals with the generation of the reports for the various modules .The customer list can be generated .Room status list can be retrieved for reference .The check in and checkout registers can also be retrieved for any future queries.

Chapter 4

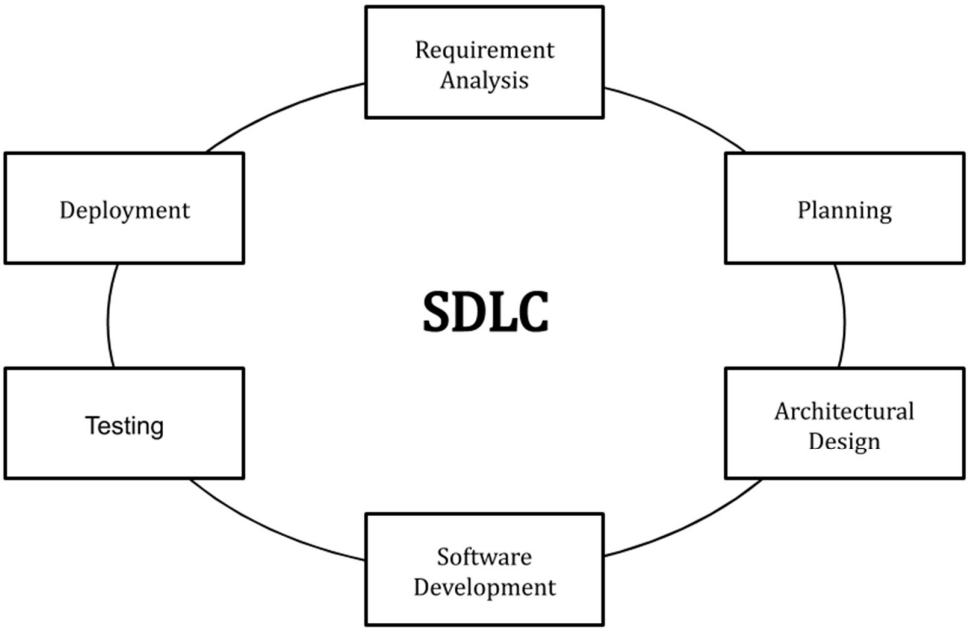
SOFTWARE ENGINEERING PARADIGMS

Software paradigm refers to the methods and stages used in the production of software. Software programming paradigm is a subset of software design paradigm, which is a subset of software development paradigm in the future. A collection of executable programming code, accompanying libraries, and documentation is referred to as software. All engineering ideas relevant to software development are referred to as software engineering in the software development paradigm. It is divided into sections like as requirements gathering, software design, programming, and so on. Software development includes the software design paradigm. It entails design, upkeep, and programming.

SOFTWARE DEVELOPMENT LIFE CYCLE

The Software Development Life Cycle (SDLC) is an approach for developing high-quality software that includes well defined processes. The Software Development Life Cycle, or SDLC, is a method for producing high-quality, low-cost software in the least amount of time. SDLC is a well-structured flow of stages that enables a company to quickly develop high-quality software that has been thoroughly tested and is ready for production. The SDLC technique focuses on the phases of software development as follows:

- Requirement Analysis
- Planning
- Architectural Design
- Software Development
- Testing
- Deployment


Requirement Analysis
Planning
Architectural Design
Software Development
Testing
Deployment
SDLC

E-R Diagram :-

ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.

In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

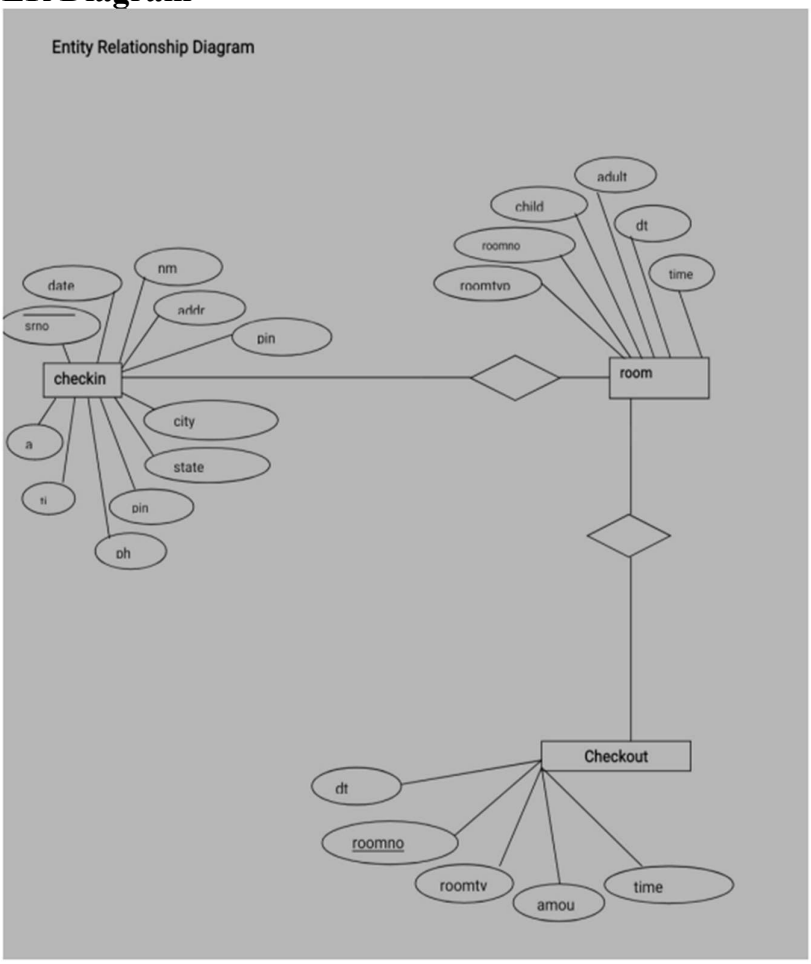
Component of ER Diagram:

1. Entity: An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

2. Attribute: The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

3. Relationship: A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

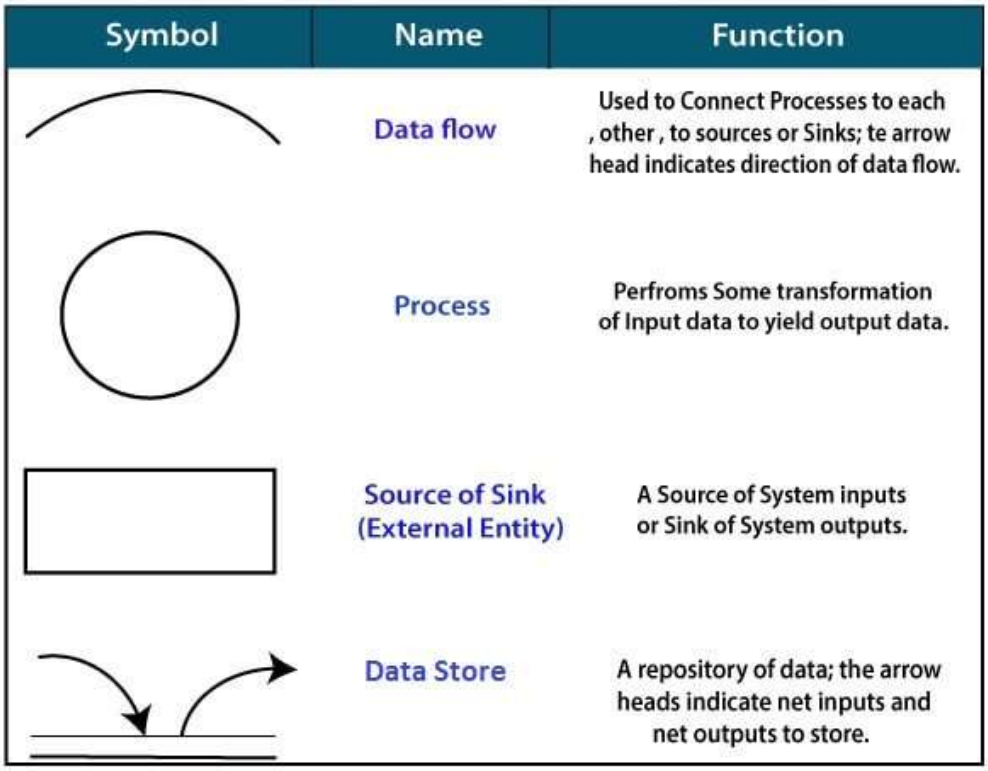
ER Diagram



DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement

graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored.



Symbols for Data Flow Diagrams

Circle: A circle (bubble) shows a process that transforms data inputs into data outputs.

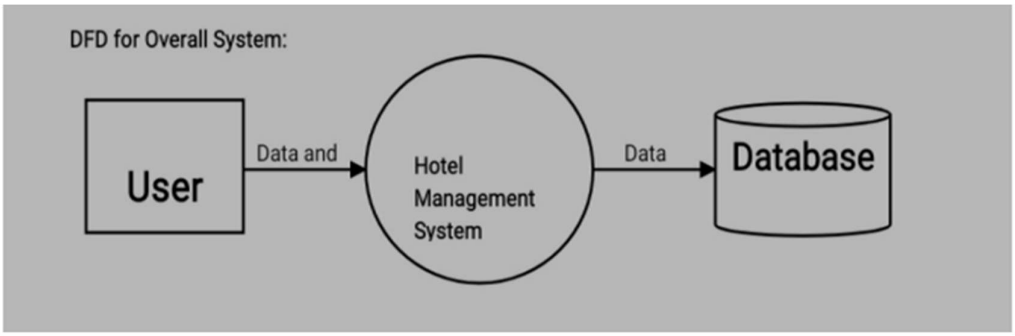
Data Flow: A curved line shows the flow of data into or out of a process or data store.

Data Store: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.

Source or Sink: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

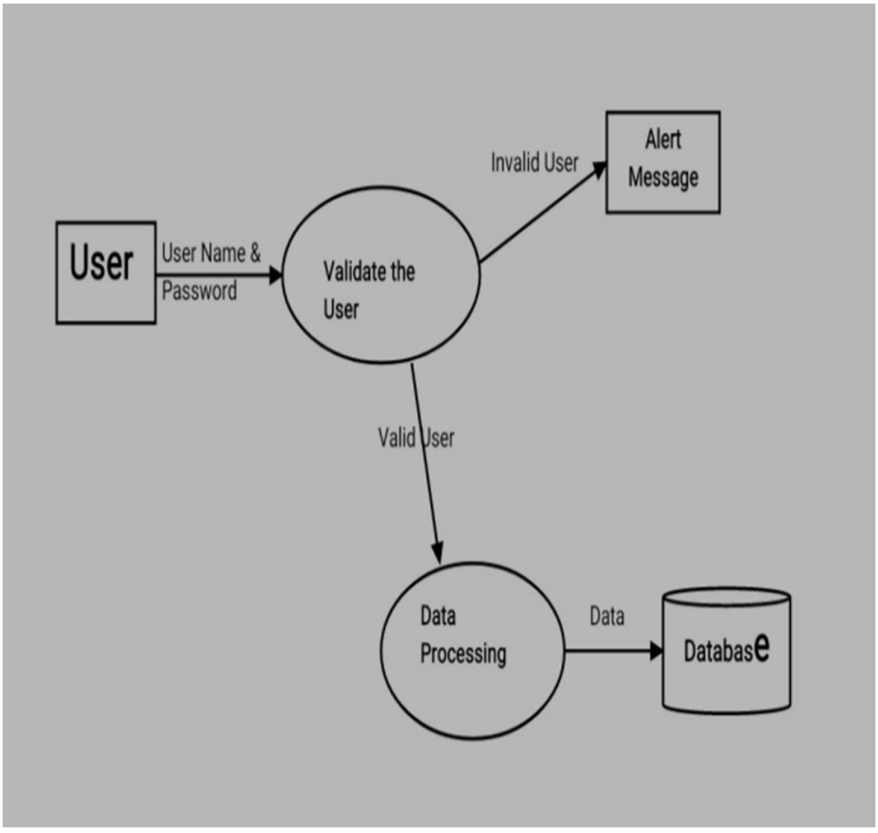
Levels in Data Flow Diagrams (DFD):

1. **0-level DFD:** It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles.

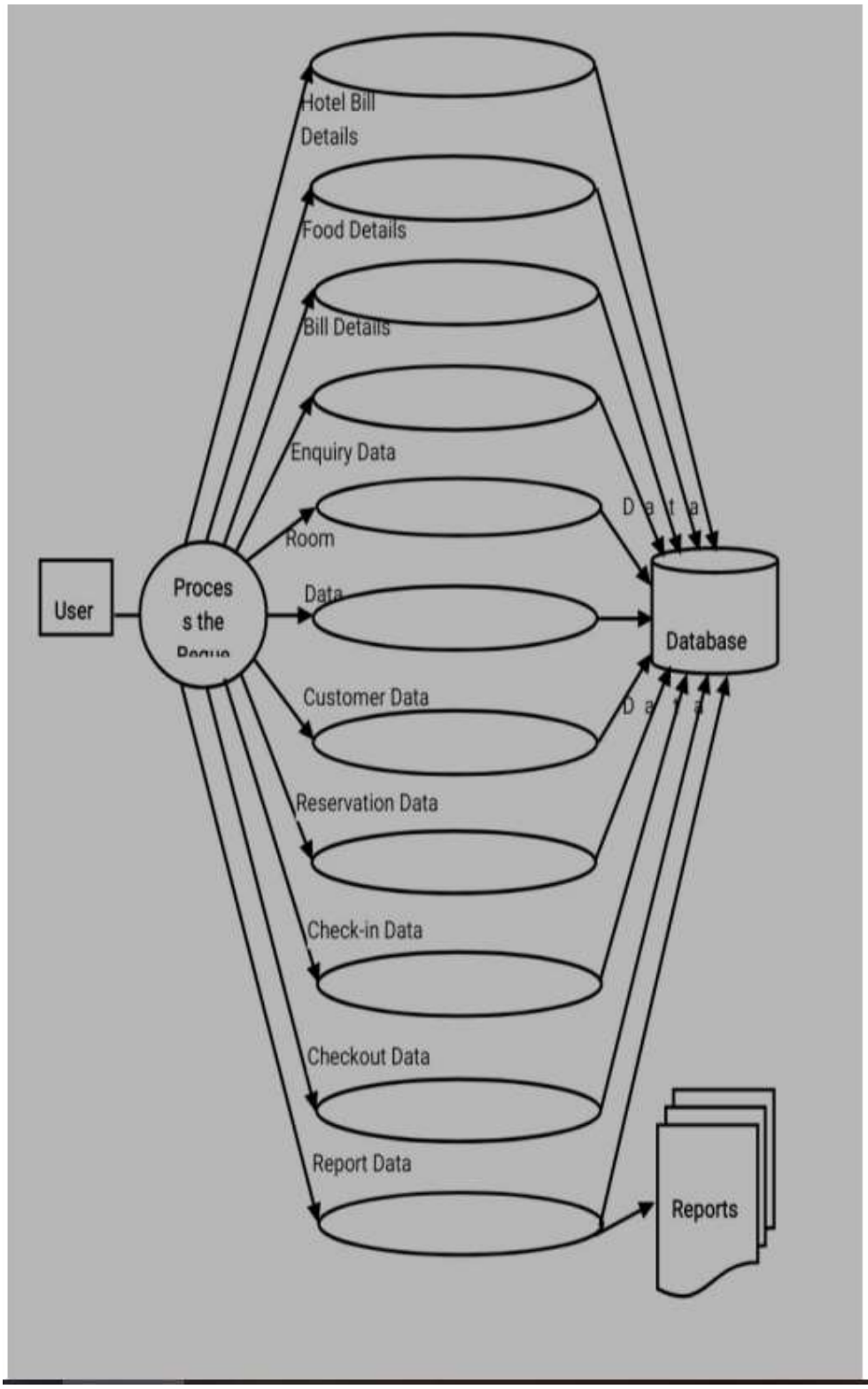


2. **1-level DFD:** In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into subprocesses.

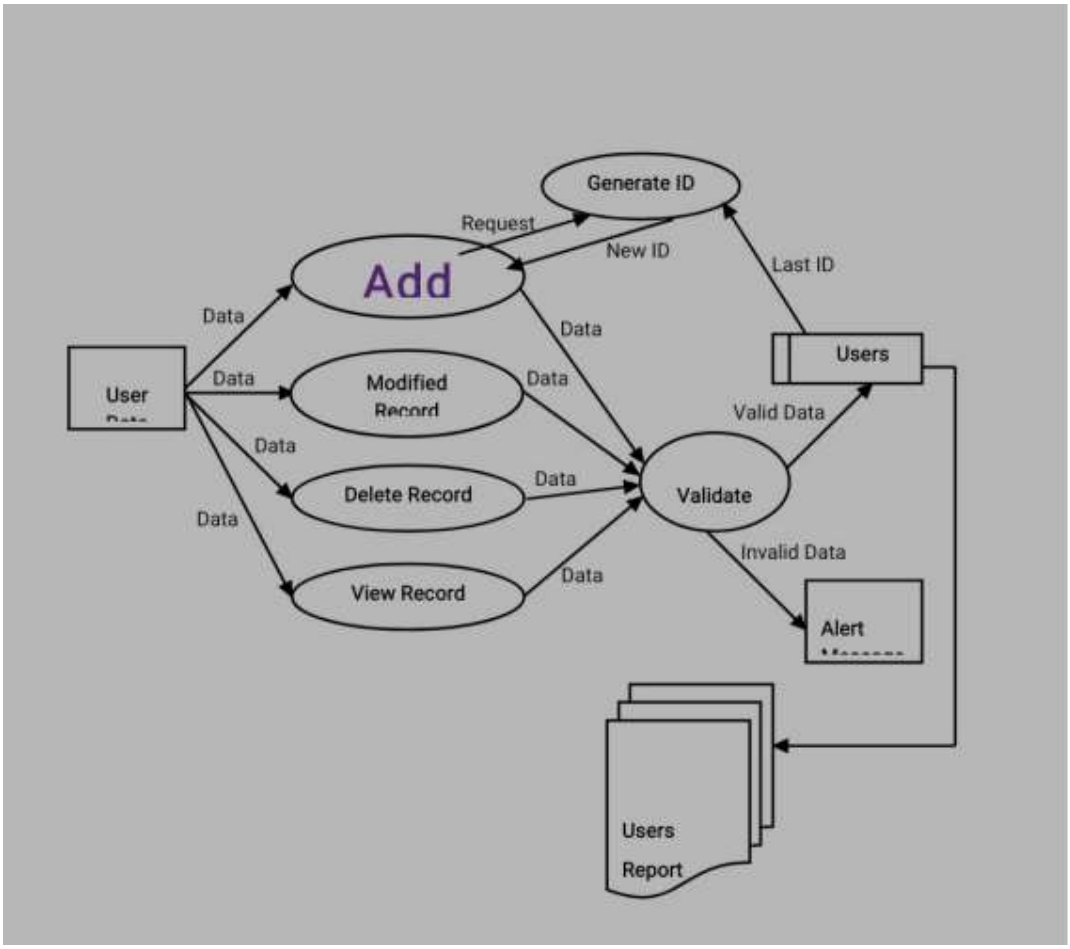
DFD For Authentication



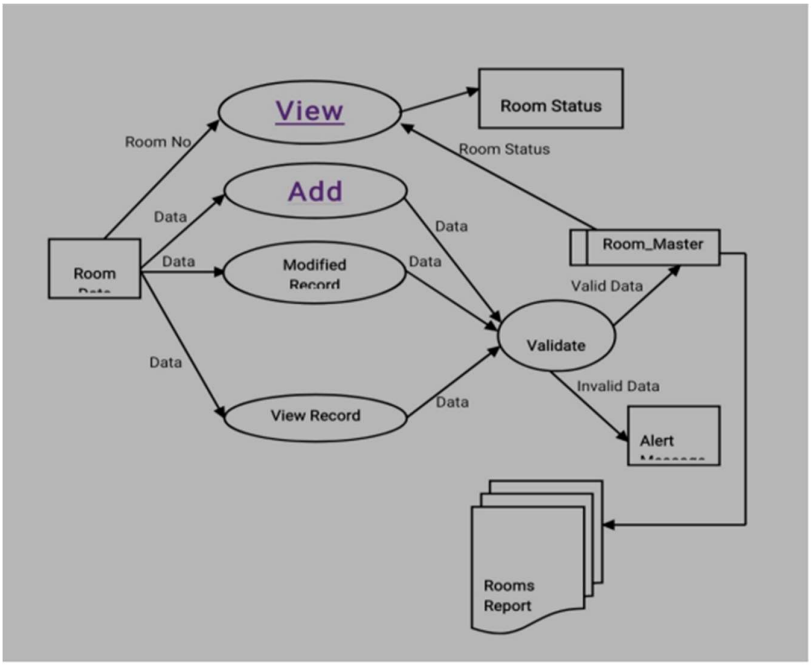
DFD for modules in the hotel management system



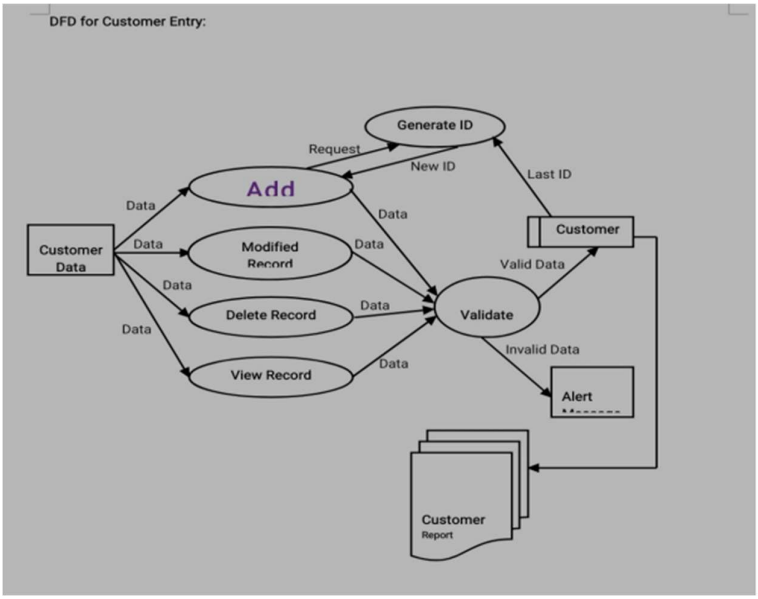
DFD for user entry



DFD For Room entry



DFD For customer entry



Chapter 5

ROLE IN THE PROJECT

My role in this project is on the frontend side. I have done all the designing on the frontend. I am also helping a little bit with database design. I am using HTML, CSS and Java script on the frontend to design the web application.

- Frontend Designing
- Helping in Database Design
- Implement all the logic on the frontend

Chapter 6

WORKING METHODOLOGY

Software Requirements:

- Operating System: Windows (7/8/8.1/10) macOS / Linux/ Android/ IOS.
- Browser: Google Chrome / Mozilla Firefox / Safari / Edge.
- Client: Web Browser
- Tool: Microsoft VS Code
- HTML 5, CSS 3, JAVASCRIPT, PHP
- MYSQL Database

Hardware Requirements:

- Depends on overall incoming traffic and load
- Processor: Intel i3
- RAM: 8 GB DDR4
- Internet Connectivity will be required*

Chapter 7

TESTING & RESULT

Functions	Description	%TC's Passed	Remarks
booking	Check booking is created	100%	
Complaint	Check complaint is created	100%	
Customer	Check New customer is booking room	100%	
Staff	Check staff is edited	100%	
Shift	Check shift is added	100%	
Emp _ history	Check emp history is Done	100%	

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also. Psychology of Testing : The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work. Testing is the process of executing a program with the intent of finding errors.

Testing Objectives: The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say

¬ Testing is a process of executing a program with the intent of finding an error.

¬ A successful test is one that uncovers an as yet undiscovered error.

¬ A good test case is one that has a high probability of finding error, if it exists.

¬ The tests are inadequate to detect possibly present errors.

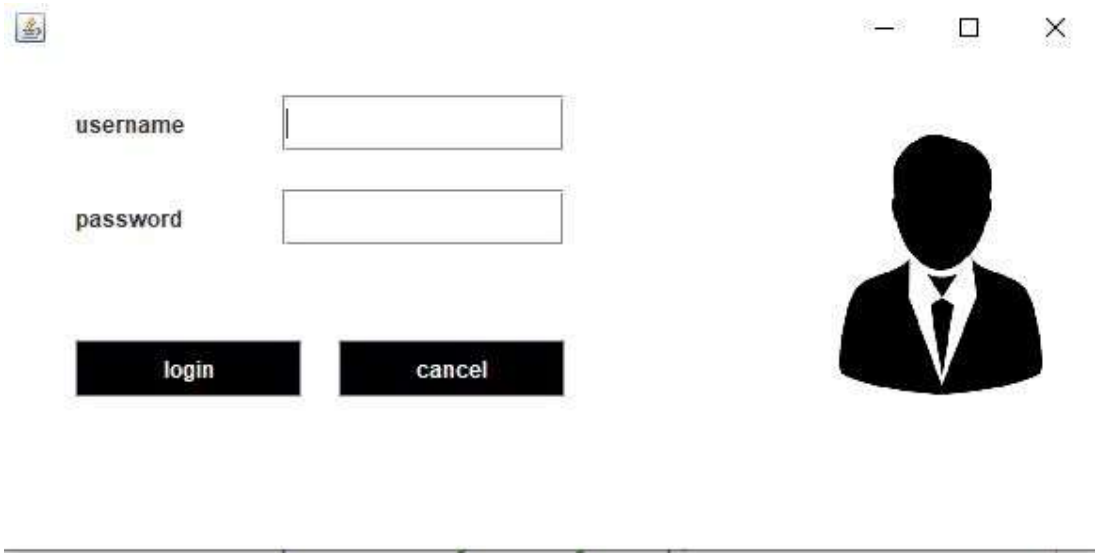
¬ The software more or less confirms to the quality and reliable standards.

Chapter 8

SCREENSHOTS

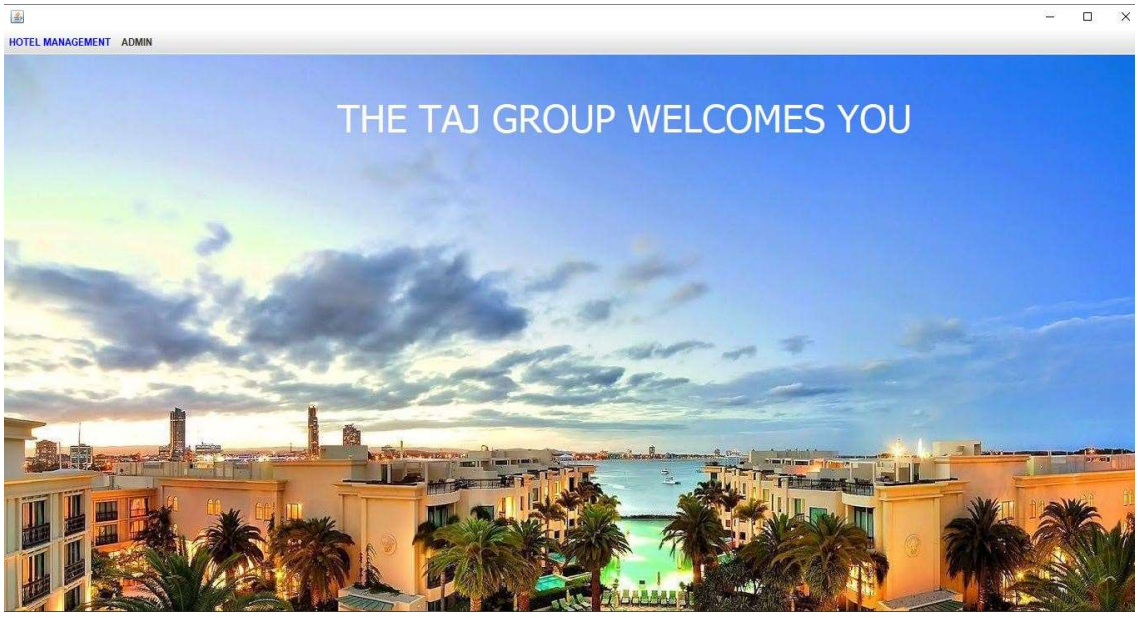
Index

Login

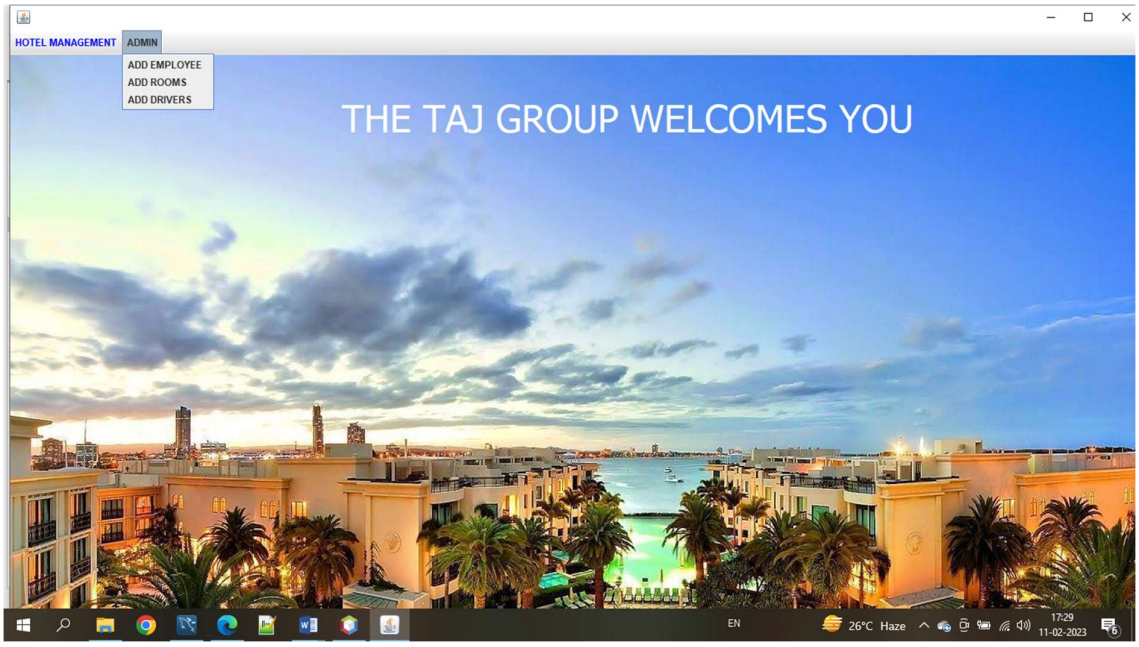


Dashboard

min

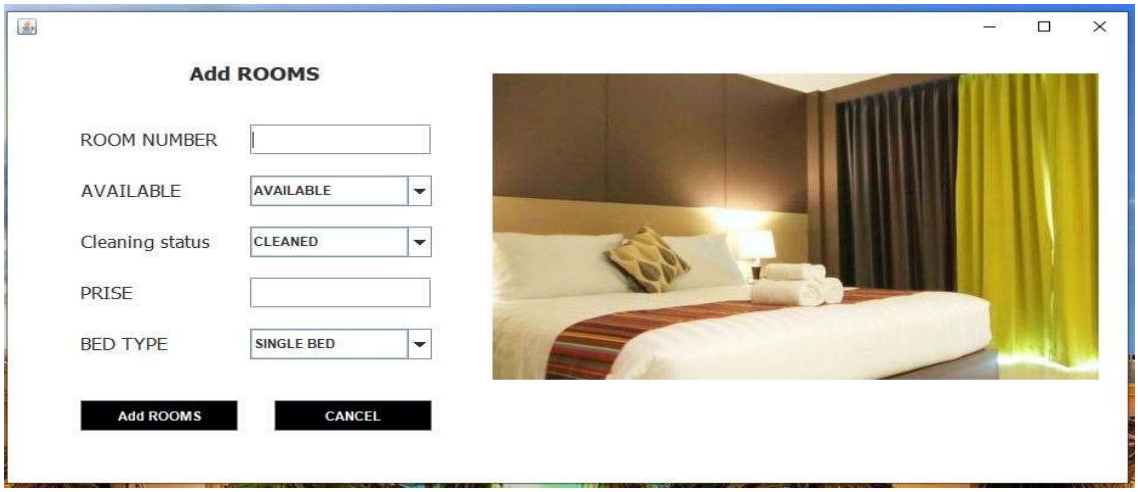


Admin Panel

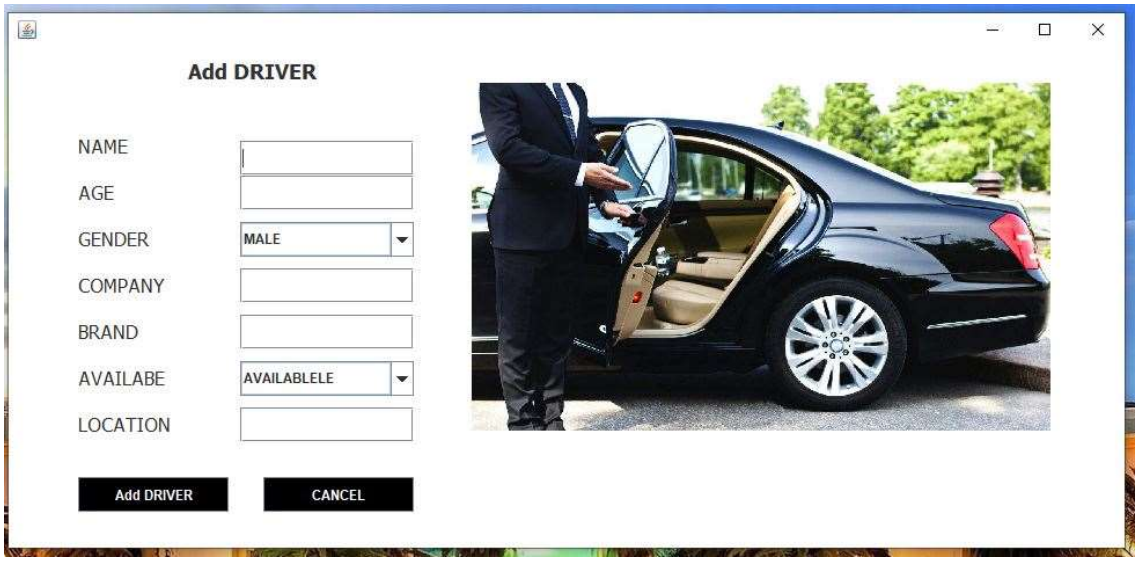


Add Employee

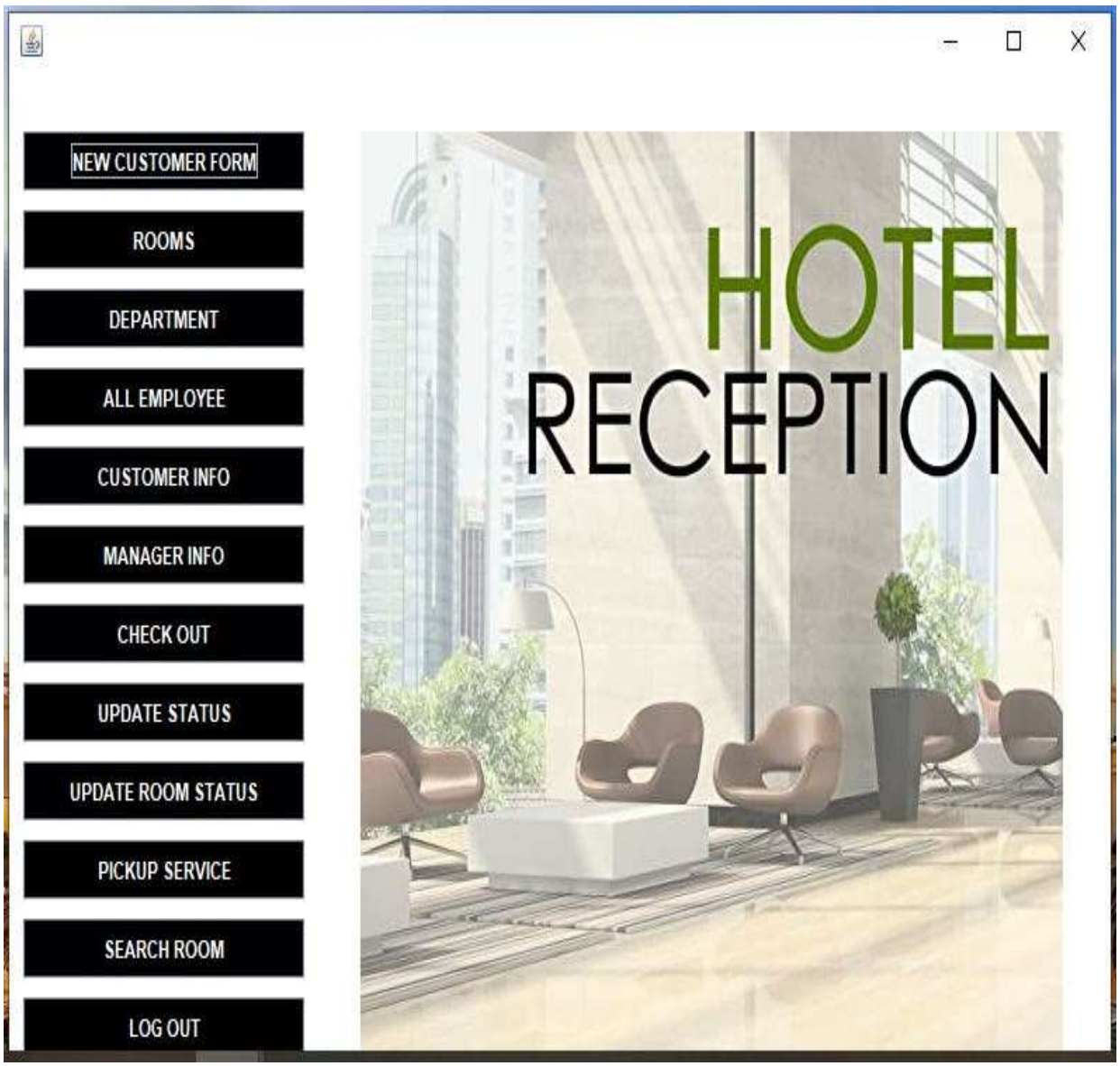
Add Rooms



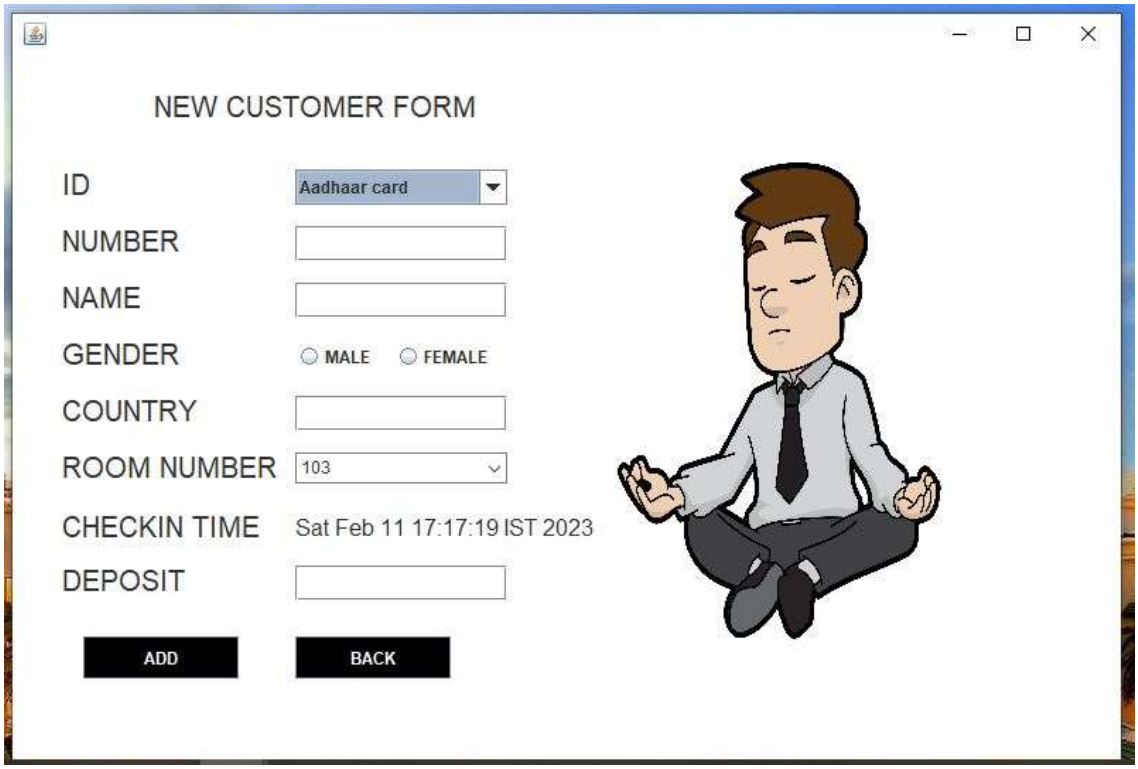
Add Drivers



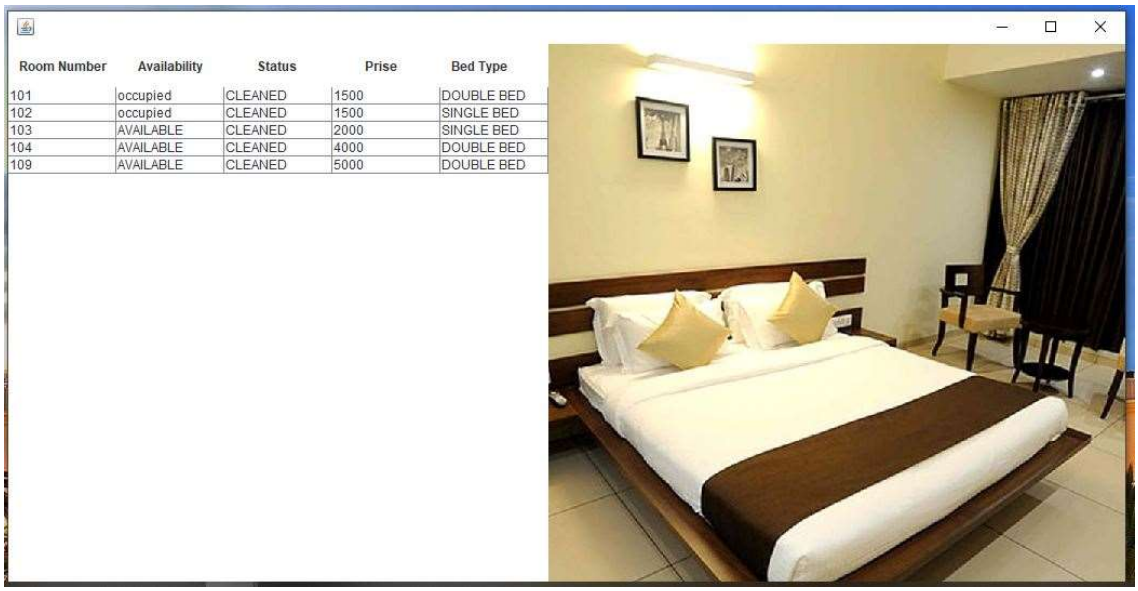
Reception


NEW CUSTOMER FORM
ROOMS
DEPARTMENT
ALL EMPLOYEE
CUSTOMER INFO
MANAGER INFO
CHECK OUT
UPDATE STATUS
UPDATE ROOM STATUS
PICKUP SERVICE
SEARCH ROOM
LOG OUT
HOTEL
RECEPTION

New Customer



Rooms List



Room Number	Availability	Status	Prise	Bed Type
101	occupied	CLEANED	1500	DOUBLE BED
102	occupied	CLEANED	1500	SINGLE BED
103	AVAILABLE	CLEANED	2000	SINGLE BED
104	AVAILABLE	CLEANED	4000	DOUBLE BED
109	AVAILABLE	CLEANED	5000	DOUBLE BED

SOURCE CODE

Index :-

```
package hotelmanagementsystem;

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 i=new ImageIcon(ClassLoader.getSystemResource("icons/first.jpg"));
      JLabel image=new JLabel(i);
      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 ae){
          setVisible(false);
          new Login();
        }

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

    }

}
```

LOGIN PAGE

```
package hotelmanagementsystem;

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

public class Login extends JFrame implements ActionListener {

    JTextField username,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 JTextField();
        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(ClassLoader.getSystemResource("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(350,10,300,200);
         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();

  }

}
```

DASHBOARD

```
package hotelmanagementsystem;
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(ClassLoader.getSystemResource("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("THE TAJ GROUP WELCOMES YOU");
         text.setBounds(400,80,1000,50);
         text.setFont(new Font("Tahoma",Font.PLAIN,46));
         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");
        hotel.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 AddRooms();

    }else if (ae.getActionCommand().equals("ADD DRIVERS")){
      new AddDriver();
    }else if(ae.getActionCommand().equals("RECEPTION")){
      new Reception();
    }


}

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

}
```

Admin Panel

1 :- Add Employee

```
package hotelmanagementsystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;


public class AddEmployee extends JFrame implements ActionListener{

    JTextField tfname,tfage,tfsalary,tfemail,tfphone,tfaadhar;
    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 bg=new ButtonGroup();
    bg.add(rbmale);
    bg.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 CLERK","HOUSELEEPING" , "KITCHENSTAFF",
"ROOM SERVICE", "CHEFS",
"WAITER/WAITRESS","MANAGER","ACCOUNTENT"};
    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("E-MAIL");
    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 lblaadhar=new JLabel("AADHAR");
```

```
lblaadhar.setBounds(60,380,120,30);
  lblaadhar.setFont(new Font("Tahoma",Font.PLAIN,17));
  add(lblaadhar);

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

  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(ClassLoader.getSystemResource("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 aadhar=tfaadhar.getText();

     String gender=null;



     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)cbjob.getSelectedItem();
       try{
          Conn conn=new Conn();
          String query="insert into employee
values('"+name+"','"+age+"','"+gender+"','"+job+"','"+salary+"','"+phone+"','"+email+"','"+aa
dhar+"')";

          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();
   }
}
```

2 :- Add Rooms

```
package hotelmanagementsystem;

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


public class AddRooms extends JFrame implements ActionListener {

   JButton add,cancel;
   JTextField tfroom,tfprise;
   JComboBox typecombo,availablecombo,cleancombo;

   AddRooms(){

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

      JLabel heading=new JLabel("Add ROOMS");
      heading.setFont(new Font("Tahoma",Font.BOLD,18));
      heading.setBounds(150,20,200,20);
      add(heading);


      JLabel lblroomno=new JLabel("ROOM NUMBER");
      lblroomno.setFont(new Font("Tahoma",Font.PLAIN,16));
```

```
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.PLAIN,16));
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.PLAIN,16));
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 lblprise=new JLabel("PRISE");
lblprise.setFont(new Font("Tahoma",Font.PLAIN,16));
lblprise.setBounds(60,230,120,30);
add(lblprise);

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


JLabel lbltype=new JLabel("BED TYPE");
lbltype.setFont(new Font("Tahoma",Font.PLAIN,16));
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 ROOMS");
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(ClassLoader.getSystemResource("icons/twelve.jpg"));
        JLabel image=new JLabel(i1);
        image.setBounds(400,30,500,300);
        add(image);

        setBounds(330,200,940,470);
        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=tfprise.getText();
            String  type=(String)typecombo.getSelectedItem();

              try{
                 Conn conn=new Conn();
                 String str="insert into room
values('"+roomnumber+"','"+availability+"','"+status+"','"+price+"','"+type+"')";

                 conn.s.executeUpdate(str);

                 JOptionPane.showMessageDialog(null,"New Rooms Added Successfully");
                  setVisible(false);



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

        }else{
            setVisible(false);
        }

    }

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

```
 new AddRooms();
    }
}
```

3:- Add Drivers

```
package hotelmanagementsystem;

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


public class AddDriver extends JFrame implements ActionListener {

    JButton add,cancel;
    JTextField tfname,tfage,tfmodel,tfcompany,tfbrand,tflocation;
    JComboBox availablecombo,gendercombo;

    AddDriver(){

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

        JLabel heading=new JLabel(" Add DRIVER");
        heading.setFont(new Font("Tahoma",Font.BOLD,18));
        heading.setBounds(150,10,200,20);
        add(heading);


        JLabel lblroomno=new JLabel("NAME");
        lblroomno.setFont(new Font("Tahoma",Font.PLAIN,16));
        lblroomno.setBounds(60,70,120,30);
        add(lblroomno);

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

        JLabel lblage=new JLabel("AGE");
        lblage.setFont(new Font("Tahoma",Font.PLAIN,16));
        lblage.setBounds(60,110,120,30);
        add(lblage);

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

        JLabel lblclean=new JLabel("GENDER");
```

```
lblclean.setFont(new Font("Tahoma",Font.PLAIN,16));
    lblclean.setBounds(60,150,120,30);
    add(lblclean);

    String cleanOptions[]={"MALE","FEMALE"};
    gendercombo=new JComboBox(cleanOptions);
    gendercombo.setBounds(200,150,150,30);
    gendercombo.setBackground(Color.WHITE);
    add(gendercombo);

    JLabel lblprise=new JLabel("COMPANY");
    lblprise.setFont(new Font("Tahoma",Font.PLAIN,16));
    lblprise.setBounds(60,190,120,30);
    add(lblprise);

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



    JLabel lbltype=new JLabel("BRAND");
    lbltype.setFont(new Font("Tahoma",Font.PLAIN,16));
    lbltype.setBounds(60,230,120,30);
    add(lbltype);

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

    JLabel lblavailable=new JLabel("AVAILABE");
    lblavailable.setFont(new Font("Tahoma",Font.PLAIN,16));
    lblavailable.setBounds(60,270,120,30);
    add(lblavailable);

    String driverOptions[]={"AVAILABLELE","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.PLAIN,16));
     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(ClassLoader.getSystemResource("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(300,200,980,470);
        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=tfbrand.getText();
            String available=(String) availablecombo.getSelectedItem();
            String location=tflocation.getText();


              try{
                 Conn conn=new Conn();
                 String str = "insert into driver
            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 AddDriver();
   }

}
```

RECEPTION

1 :- New Customer

```
package hotelmanagementsystem;

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

public class AddCustomer extends JFrame implements ActionListener{

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

   AddCustomer(){
      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[] = {"Aadhaar card","Passport","Driving License","Voter-id
card","Ration-card"};
      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(270,200,100,25);
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(35,280,165,20);
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("CHECKIN TIME");
    lbltime.setBounds(35,320,150,25);
    lbltime.setFont(new Font("Raleway",Font.PLAIN,20));
    add(lbltime);

    Date date=new Date();


    checkintime=new JLabel(""+date);
    checkintime.setBounds(200,320,250,25);
    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(50,410,110,30);
    add.addActionListener(this);
    add(add);

    back=new JButton("BACK");
    back.setBackground(Color.BLACK);
    back.setForeground(Color.WHITE);
    back.setBounds(200,410,110,30);
    back.addActionListener(this);
    add(back);

    ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("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 room = croom.getSelectedItem();
          String time = checkintime.getText();
          String deposit = tfdeposit.getText();

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

              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 AddCustomer();
    }

}
```

2 :- Rooms

```
package hotelmanagementsystem;
```

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


public class AddRooms extends JFrame implements ActionListener {

    JButton add,cancel;
    JTextField tfroom,tfprise;
    JComboBox typecombo,availablecombo,cleancombo;

    AddRooms(){

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

        JLabel heading=new JLabel("Add ROOMS");
        heading.setFont(new Font("Tahoma",Font.BOLD,18));
        heading.setBounds(150,20,200,20);
        add(heading);


        JLabel lblroomno=new JLabel("ROOM NUMBER");
        lblroomno.setFont(new Font("Tahoma",Font.PLAIN,16));
        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.PLAIN,16));
        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.PLAIN,16));
        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 lblprise=new JLabel("PRISE");
    lblprise.setFont(new Font("Tahoma",Font.PLAIN,16));
    lblprise.setBounds(60,230,120,30);
    add(lblprise);

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


    JLabel lbltype=new JLabel("BED TYPE");
    lbltype.setFont(new Font("Tahoma",Font.PLAIN,16));
    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 ROOMS");
    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(ClassLoader.getSystemResource("icons/twelve.jpg"));
    JLabel image=new JLabel(i1);
    image.setBounds(400,30,500,300);
    add(image);

    setBounds(330,200,940,470);
    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=tfprise.getText();
        String  type=(String)typecombo.getSelectedItem();
```

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

                conn.s.executeUpdate(str);

                JOptionPane.showMessageDialog(null,"New Rooms Added Successfully");
                 setVisible(false);



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

        }else{
            setVisible(false);
        }

    }

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

}
```

3 :- Customer Info

```
package hotelmanagementsystem;



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(l1);


   JLabel l2 = new JLabel("Number");

   l2.setBounds(160,10,100,20);

   add(l2);


   JLabel l3 = new JLabel("Name");

   l3.setBounds(290,10,100,20);

   add(l3);


   JLabel l4 = new JLabel("Gender");
```

```
l4.setBounds(410,10,100,20);
add(l4);

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

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

JLabel l7 = new JLabel("Checkin time");
l7.setBounds(760,10,100,20);
add(l7);

JLabel l8 = new JLabel("Deposit");
l8.setBounds(900,10,100,20);
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 customer");
       table.setModel(DbUtils.resultSetToTableModel(rs));

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

    back=new JButton("Back");
    back.setBackground(Color.BLACK);
    back.setForeground(Color.WHITE);
    back.addActionListener(this);
    back.setBounds(420,500,120,30);
    add(back);
    setBounds(300,200,1000,600);
    setVisible(true);
  }
  public void actionPerformed(ActionEvent ae){
```

```
        setVisible(false);

        new CustomerInfo();

    }

    public static void main(String[] args) {

        new CustomerInfo();

    }

}
```

4 :- Update :-

```
package hotelmanagementsystem;

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


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 Status");
      text.setFont(new Font("Tahoma",Font.PLAIN,20));
      text.setBounds(90,20,200,30);
      text.setBackground(Color.BLUE);
      add(text);

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

      ccustomer=new Choice();
      ccustomer.setBounds(200,80,150,30);
      add(ccustomer);

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

```
    ResultSet rs = c.s.executeQuery("selcect * 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("Checkin 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);
     check.addActionListener(this);
     add(check);

     update=new JButton("Update");
     update.setBackground(Color.BLACK);
     update.setForeground(Color.WHITE);
     update.setBounds(150,340,100,30);
     update.addActionListener(this);
     add(update);

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

     ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/nine.jpg"));
     JLabel image=new JLabel(i1);
     image.setBounds(400,30,500,300);
     add(image);

     setBounds(300,200,980,500);
     setVisible(true);

  }
  public void actionPerformed(ActionEvent ae){

     if (ae.getSource()==check){
        String id = ccustomer.getSelectedItem();
        String query="select * from customer from 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("checktime"));
              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 nmber = ccustomer.getSelectedItem();
        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+"'
,'"+checkin+"' , '"+deposit+"' ");



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


    }else{

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

}
```

DATABASE – MANAGEMENT

DATABASE

```
MySQL Shell
MySQL Shell 8.0.30

Copyright (c) 2016, 2022, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
 MySQL  JS  \sql
Switching to SQL mode... Commands end with ;
 MySQL  SQL  \connect root@localhost
Creating a session to 'root@localhost'
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 141 (X protocol)
Server version: 8.0.30 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost:33060+ ssl  SQL  use hotelmanagementsystem;
Default schema set to `hotelmanagementsystem`.
Fetching table and column names from `hotelmanagementsystem` for auto-completion... Press ^C to stop.
 MySQL  localhost:33060+ ssl  hotelmanagementsystem  SQL  SHOW TABLES FROM hotelmanagementsystem;
+--------------------------------+
| Tables_in_hotelmanagementsystem |
+--------------------------------+
| customer                       |
| department                     |
| driver                         |
| employee                       |
| login                          |
| room                           |
+--------------------------------+
6 rows in set (0.0016 sec)
```

DEPARTMENT

```
 MySQL  localhost:33060+ ssl  hotelmanagementsystem  SQL  select * from department;
+---------------------------+--------+
| department                | budget |
+---------------------------+--------+
| Front Office              | 500000 |
| Housekeeping              | 4000   |
| Food and Beverage         | 23000  |
| Kitchen or food production | 540000 |
| Security                  | 320000 |
+---------------------------+--------+
```

Driver

```
MySQL localhost:33060+ ssl hotelmanagementsystem SQL select * from driver;
+------+------------+--------+---------+------------+-------------+------------+
| name | age        | gender | company | brand      | available   | location   |
+------+------------+--------+---------+------------+-------------+------------+
| 18   | shashi     | MALE   | TATA    | XUV        | AVAILABLELE | lko        |
| 37   | Aman       | MALE   | jhkhfofl| jkdyl.kkk7 | AVAILABLELE | meeurt     |
| 36   | StunSultan | MALE   | pubg    | anokha     | AVAILABLELE | sabkedilme |
+------+------------+--------+---------+------------+-------------+------------+
```

Employee

```
MySQL localhost:33060+ ssl hotelmanagementsystem SQL select * from employee;
+------------------+-----+--------+----------------+----------+------------+-------------------------+-------------+
| name             | age | gender | job            | salary   | phone      | email                   | aadhar      |
+------------------+-----+--------+----------------+----------+------------+-------------------------+-------------+
| Aaditya Kasaudhan | 18 | Male   | MANAGER        | 20,00,000 | 9559812026 | adityakas1907@gmail.com | 635199833958 |
| Aastha Chaturvedi | 18 | Female | ACCOUNTENT     | 11,50,000 | 70077553561 | asthachaturved@gmail.com | 123456789009 |
| Anubhav          | 14  | Male   | FRONT DESK CLERK | 55000    | 1234567890 | annu@gnwil.cok          | 1245678     |
| aditya           | 19  | Male   | MANAGER        | 9000000  | 9559812026 | adityakas1907@gmail.com | 635199833958 |
| Ramesh Agrahari  | 45  | Male   | CHEFS          | 2000     | 234567891  | ramesh@gmail.cpm        | 123456789123 |
| Anurag Sonin     | 75  | Male   | WAITER/WAITRESS | 5000    | 8707767171 | anuragsoni@gmail.com    | 789456123321 |
+------------------+-----+--------+----------------+----------+------------+-------------------------+-------------+
6 rows in set (0.0011 sec)
```

LOGIN

```
MySQL localhost:33060+ ssl hotelmanagementsystem SQL select * from login;
+----------+----------+
| username | password |
+----------+----------+
| admin    | 12345    |
+----------+----------+
1 row in set (0.0008 sec)
```

ROOM

```
MySQL localhost:33060+ ssl hotelmanagementsystem SQL select * from room;
+------------+--------------+-----------------+-------+------------+
| roomnumber | availability | cleaning_status | prise | bed_type   |
+------------+--------------+-----------------+-------+------------+
| 101        | occupied     | CLEANED         | 1500  | DOUBLE BED |
| 102        | occupied     | CLEANED         | 1500  | SINGLE BED |
| 103        | AVAILABLE    | CLEANED         | 2000  | SINGLE BED |
| 104        | AVAILABLE    | CLEANED         | 4000  | DOUBLE BED |
| 109        | AVAILABLE    | CLEANED         | 5000  | DOUBLE BED |
+------------+--------------+-----------------+-------+------------+
5 rows in set (0.0009 sec)
```

CUSTOMER

```
MySQL localhost:33060+ ssl hotelmanagementsystem SQL select * from customer;
+-------------+------------+---------+--------+---------+------+-----------------------------+---------+
| document    | number     | name    | gender | country | room | chechtime                   | deposit |
+-------------+------------+---------+--------+---------+------+-----------------------------+---------+
| Aadhaar card | 123456789987 | Divyansh | MALE | India | 102 | Tue Aug 16 17:51:57 IST 2022 | 1000 |
| Passport    | 12345      | JOHN    | MALE   | U.S.    | 101  | Tue Aug 16 18:12:13 IST 2022 | 1500    |
+-------------+------------+---------+--------+---------+------+-----------------------------+---------+
2 rows in set (0.0009 sec)
```

Chapter 9

CONCLUSION

At last it can be concluded that the Hotal Management System was a real learning experience. The principles of software production were well implemented throughout the system. The whole project undergoes with full of enthusiasm and with full of joyous moments. The project has been made as per as the given specification. Working on the Project was really a learning experience and we have come a long way in building our concepts of Software engineering. The "Hotal Management System" developed by us is purely based on Php. The overall purpose of this system is to computerized the whole process and thus prevent the intervening errors. During the course of this assignment we have gone through many obstacles which made us to research and though increased our knowledge. After applying all the data modeling, object modeling and process modeling techniques now we are very well clear with all these concepts and fundamentals which will be going to help us in the future

Chapter 10

FUTURE SCOPE

India is being considered as one of the most popular travel destinations in the world. Tourism contributed 6.9% to our country's GDP in 2015 and almost 8.7% of the employment. And this sector is expected to grow tremendously, roughly 7.5% of our GDP by 2025. Today, Hotel Management is not only restricted to hotels but has gone a long way to catering, clubs, food and beverage industry,resorts, airlines, cruise and many more. The hotel management forms a big part of the tourism, and the government too is in favor of investing in it and supporting it in every way possible.

THANK YOU !