HOTEL MANAGEMENT SYSTEM

CS23333 – Object Oriented Programming using Java Project Report

Submitted by

SRIMAN VIYASEN SJ - 231001208

THARUN V - 231001231

VISHAL S - 231001248

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE

NOVEMBER-2024

BONAFIDE CERTIFICATE

Certified that this project titled “HOTEL MANAGEMENT SYSTEM” is the bonafide work of **“SRIMAN VIYASEN SJ (231001208), THARUN V (231001231), VISHAL S (231001248)”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.P.Valarmathie	**Mr.K.E.Narayana**
HEAD OF THE DEPARTMENT	**Assistant Professor,**
Information Technology,	**Information Technology,**
Rajalakshmi Engineering College	**Rajalakshmi Engineering College**
(Autonomous),	**(Autonomous),**
Thandalam,Chennai - 602 105	**Thandalam,Chennai - 602 105**

This project is submitted for CS23333 – Object Oriented Programming using Java held on __________

INTERNAL EXAMINAR	**EXTERNAL EXAMINAR**

TABLE OF CONTENTS

1. Abstract

The Hotel Management System is a Java and MySQL-based application designed to simplify hotel operations. It automates processes like room booking, check-in/out, customer management, and billing. The system ensures data accuracy, efficiency, and user-friendly interaction, enhancing both staff productivity and customer satisfaction. This report details its design, implementation, and practical applications in the hospitality sector. The system streamlines daily hotel operations, reducing manual effort and minimizing errors in guest data and financial transactions. It is scalable, allowing future upgrades or integration with other services like payment gateways and reporting tools, supporting the growth of the hotel business.

2. Introduction

The Hotel Management System (HMS) is a software solution built using Java and MySQL to automate and streamline hotel operations. It handles tasks such as room reservations, guest check-ins/check-outs, billing, and customer management. By replacing manual processes, the system improves efficiency, reduces errors, and enhances customer satisfaction. This report outlines the system's design, functionality, and its application in the hospitality industry.

3. Purpose

The purpose of the Hotel Management System is to automate and optimize hotel operations, providing a seamless experience for both guests and hotel staff. By leveraging Java and MySQL, the system simplifies tasks such as room booking, check-in/check-out, billing, and customer data management. The goal is to enhance operational efficiency, reduce errors, and improve overall customer satisfaction, making hotel management processes faster and more reliable.

4. Scope of the Project:

The Hotel Management System aims to automate core hotel operations, including room management, reservations, check-ins, check-outs, and billing. It allows guests to book rooms, modify or cancel reservations, and provides an efficient check-in/check-out process. Billing and payment functionalities are integrated to generate invoices and manage payments securely. The system also stores guest information, enabling personalized services and repeat bookings.

Using MySQL for data storage, the system ensures secure and efficient management of hotel data. It supports role-based user access, allowing staff to perform tasks according to their responsibilities. The system also generates reports on occupancy, revenue, and guest history to assist in decision-making. Designed for scalability, it can easily accommodate future expansions and additional features as the hotel's needs grow.

5. Software Requirement Specification

Introduction

The Hotel Management System requires specific software tools to ensure efficient development, deployment, and performance. It automates hotel operations such as room management, reservations, billing, and customer data handling. This section outlines the necessary software components for building and maintaining the system.

Product Scope

The system will be developed using **Java** for application logic and **MySQL** for secure data storage. It will run on standard desktop or web platforms, requiring **Java Runtime Environment (JRE)** and **MySQL Server**. Development will be done using **IDEs** like **Eclipse** or **IntelliJ IDEA**. If a web-based interface is used, a **web server** like **Apache Tomcat** may be required for deployment.

Overall Description

The Hotel Management System automates hotel operations like room bookings, guest check-ins/outs, billing, and customer data management using **Java** and **MySQL**. It includes modules for room availability, reservations, guest management, and billing. The system ensures efficiency with real-time data updates and secure payment handling, while its user-friendly interface streamlines operations. The system is scalable for future expansion and feature integration.

Product Perspective

The Hotel Management System is a standalone application that integrates with **MySQL** for secure data storage. It automates hotel functions like bookings, check-ins, and billing, replacing manual processes. The system is flexible, supporting both desktop and web platforms, and is designed for easy future enhancements, such as advanced reporting or third-party integrations. It improves operational efficiency and customer satisfaction.

Product Functionality

a) **Room and Reservation Management**: Manages room availability, bookings, and modifications.

b) **Check-in/Check-out and Billing**: Automates check-ins, check-outs, generates bills, and processes payments.

c) **Customer Data Management**: Stores and retrieves guest information for future visits and personalized services.

d) **User Access Control**: Provides role-based access for staff to secure different modules.

e) **Real-time Data Updates**: Ensures up-to-date information on room status, bookings, and guest details.

f) **Scalability**: Designed to accommodate future feature expansions and integrations.

Operating Environment

Hardware Requirements

- Processor: Intel i3 or higher (or equivalent AMD processor)
- Operating System: Windows 8,10, 11
- Processor Speed: 2.0 GHz
- RAM: 4GB
- Hard Disk: 500GB

Software Requirements

- Database: MySQL
- Frontend: JSP
- Technology: Java (JDBC)

Hardware Interface

a) Desktop or laptop computers with keyboard, mouse, and monitor for user workstations.

b) Dedicated server or cloud infrastructure with at least 4GB RAM and 100GB storage for hosting the system.

Software Interface

a) MS-Windows Operating System .

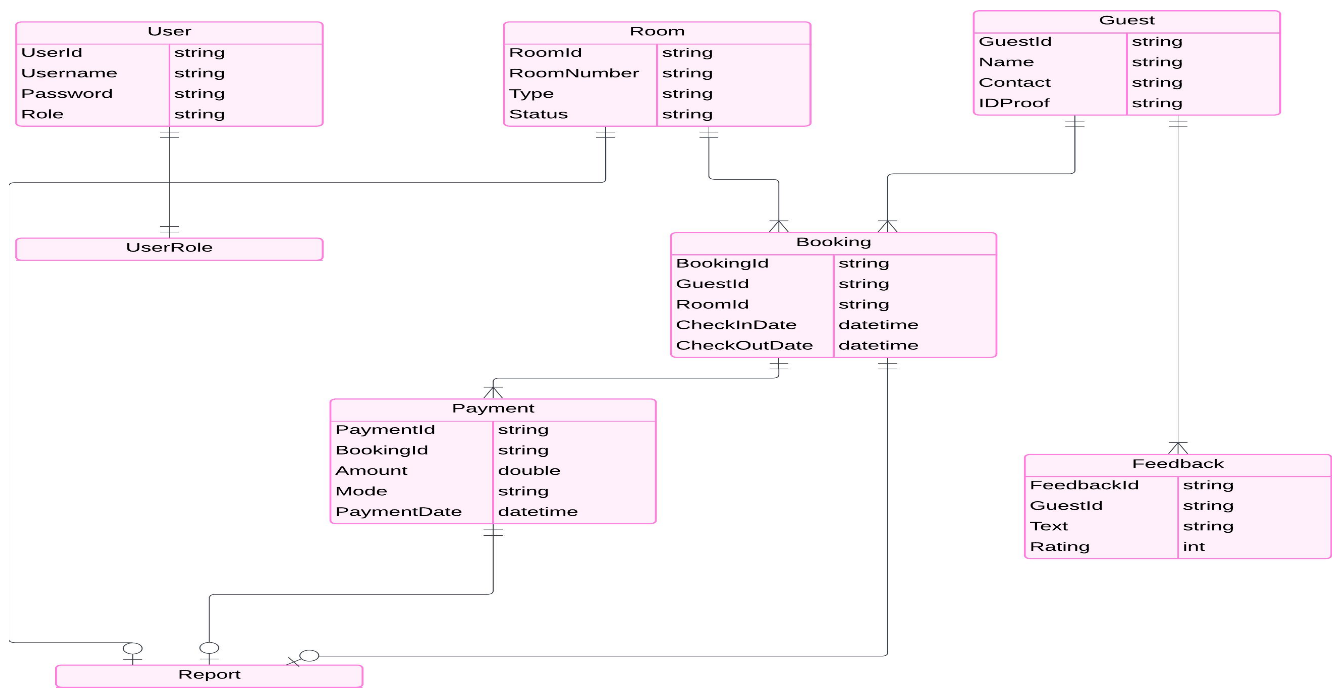
b) JSP for designing the front end .

c) java for the backend & MYSQL as database.

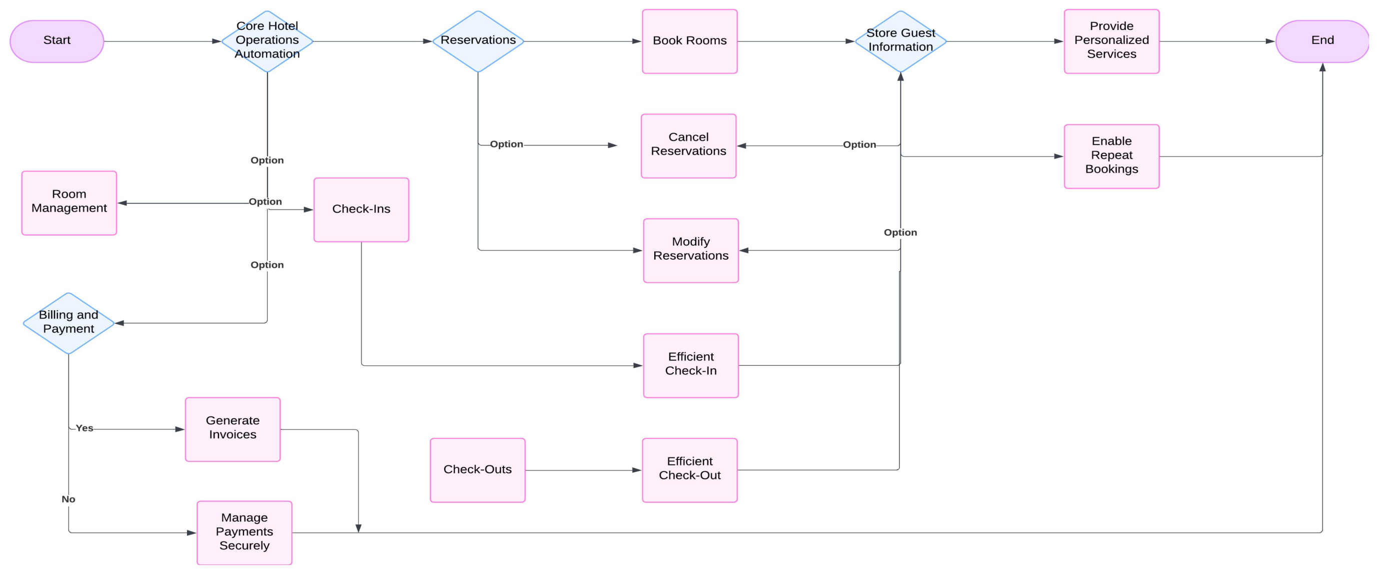
d) Platform: Java Language.

CHAPTER 2

1. Entity Relationship Diagram:



2. Flow Diagram



3. MODULE DESCRIPTION

1) **Authentication and Authorization**
   - **Description**: Handles user login and role-based access.
   - **Features**:
     - Secure login for admin, staff, and guests.
     - Password encryption and storage.
     - Role-based access control (e.g., admin vs staff).
   - **Database Tables**:
     - users: Stores user credentials and roles.
   - **JDBC Operations**:
     - Validate credentials during login.
     - Add or update user roles.
2) **Guest Management**
   - **Description**: Manages information about guests staying in the hotel.
   - **Features**:
     - Add, view, update, and delete guest records.
     - Search guests by name, phone number, or booking ID.
   - **Database Tables**:
     - guests: Stores guest details such as name, contact, and ID proof.
   - **JDBC Operations**:
     - Insert and retrieve guest details.
     - Update guest contact information.
3) **Room Management**
   - **Description**: Maintains a database of available, booked, and occupied rooms.
   - **Features**:
     - Add and manage room types (single, double, suite).
     - Update room status (available, occupied, maintenance).
     - Search for available rooms by type or date.
   - **Database Tables**:
     - rooms: Stores room details such as room number, type,

and status.

- **JDBC Operations**:
    - Fetch available rooms for a given date range.
    - Update room status after booking or checkout.

4) **Booking Management**
    - **Description**: Facilitates room bookings and ensures availability.
    - **Features**:
        - Create new bookings for guests.
        - Update or cancel existing bookings.
        - Generate booking confirmation with details.
    - **Database Tables**:
        - Bookings: Stores booking details such as guest ID, room ID, and check-in/check-out dates.
    - **JDBC Operations**:
        - Insert booking records.
        - Fetch booking history for a specific guest.

5) **Payment and Billing**
    - **Description**: Automates the billing process for guests.
    - **Features**:
        - Calculate room charges based on stay duration.
        - Add additional charges (e.g., meals, spa, laundry).
        - Generate and print invoices.
    - **Database Tables**:
        - Payments: Stores payment details such as amount, mode, and date.
    - **JDBC Operations**:
        - Insert payment records.
        - Generate invoice details using data from bookings and payments.

6) **Feedback and Reviews**
    - **Description**: Allows guests to leave feedback about their stay.
    - **Features**:
        - Record and view guest feedback.
        - Search feedback by guest or stay date.

- **Database Tables**:
    - feedback: Stores feedback text and ratings.
- **JDBC Operations**:
    - Insert and retrieve feedback records.

7) **Reports and Analytics**
    - **Description**: Generates insights and summaries for the hotel management team.
    - **Features**:
        - Generate occupancy reports by date range.
        - Analyze revenue trends.
        - View top-rated rooms or services.
    - **Database Tables**:
        - Uses bookings, rooms, and payments tables.
    - **JDBC Operations**:
        - Fetch aggregated data using SQL queries.
        - Display data in tabular or graphical formats.

8) **System Administration**
    - **Description**: Handles maintenance and configuration tasks for the system.
    - **Features**:
        - Backup and restore database.
        - Configure pricing for rooms and services.
        - Manage staff accounts and permissions.
    - **Database Tables**:
        - Various administrative tables, e.g., pricing.
    - **JDBC Operations**:
        - Update pricing for rooms/services.
        - Add or remove staff accounts.

4.1 Design

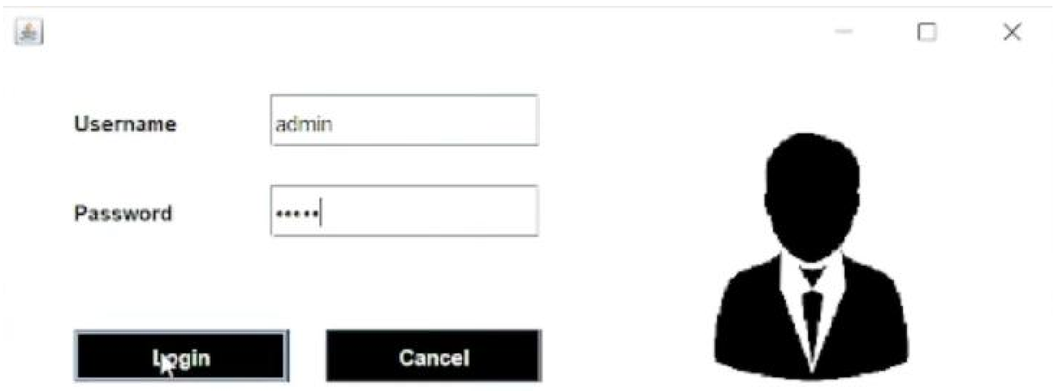


Figure 4.1.1 Login Page

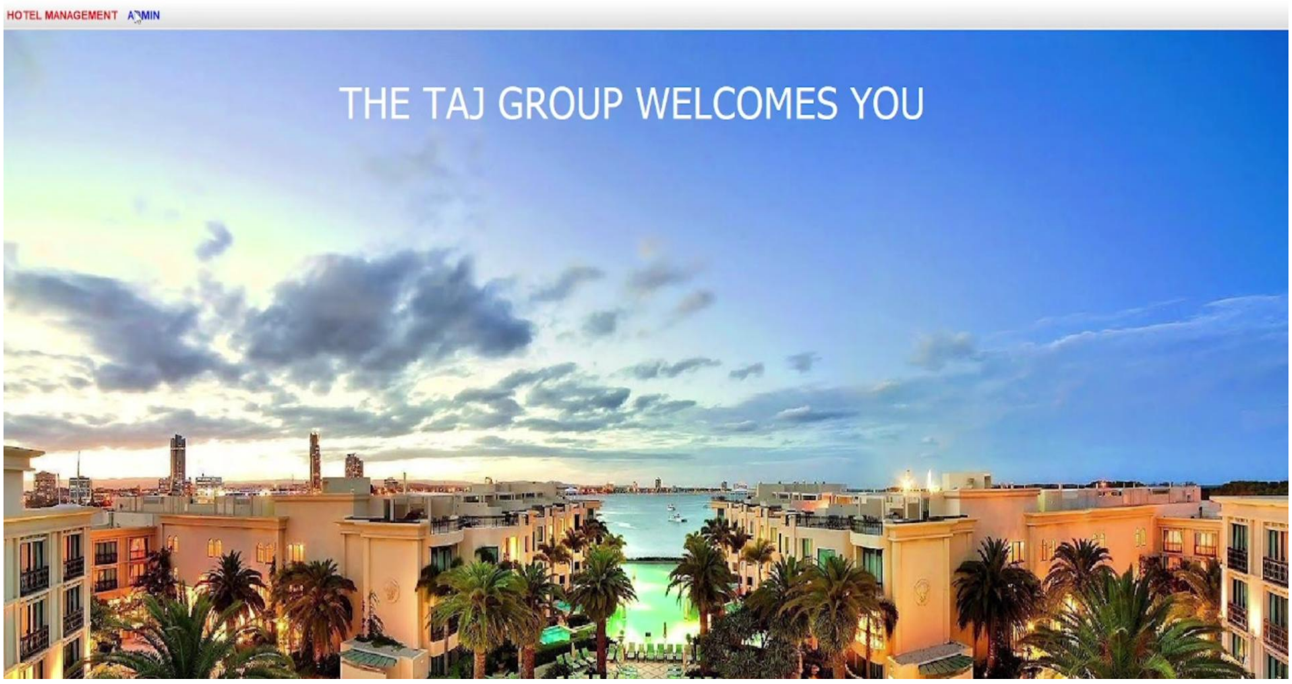


Figure 4.1.2 Home page

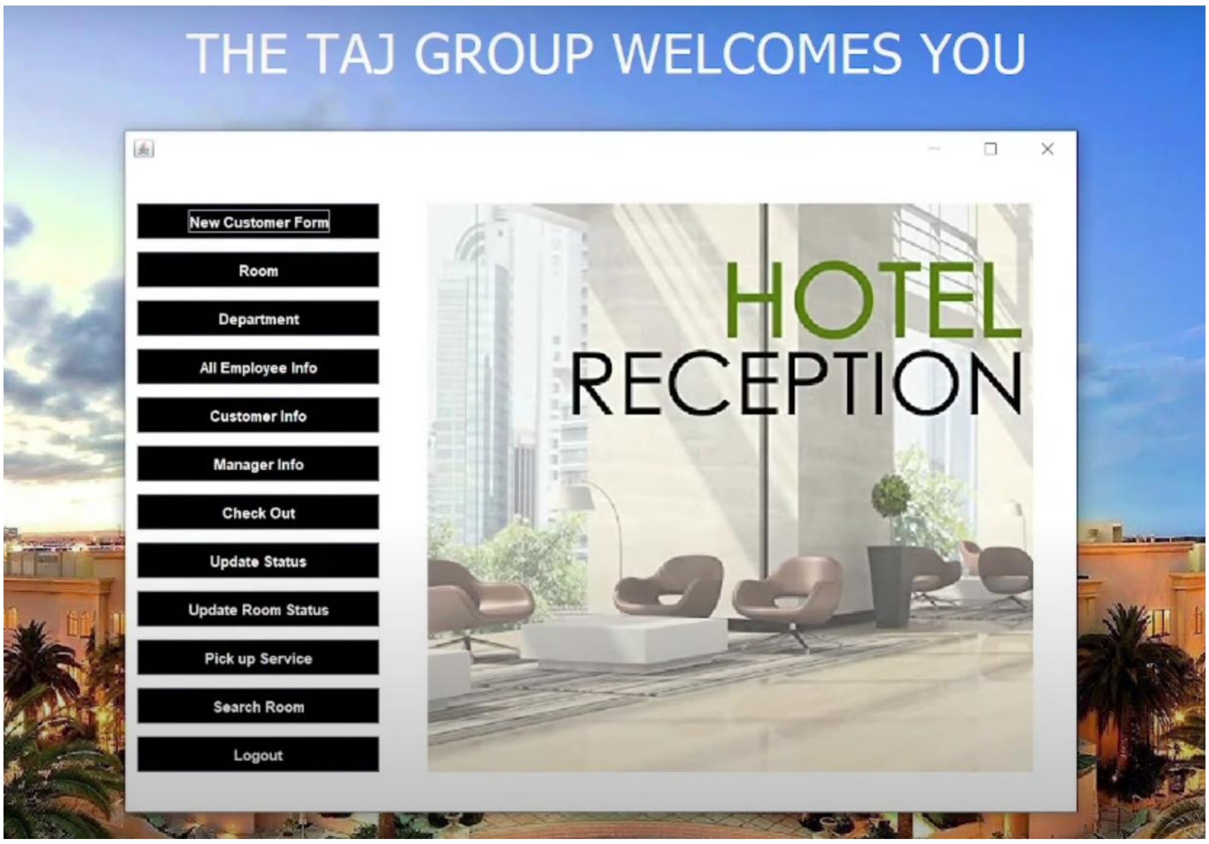


Figure 4.1.3 Hotel Reception Interface

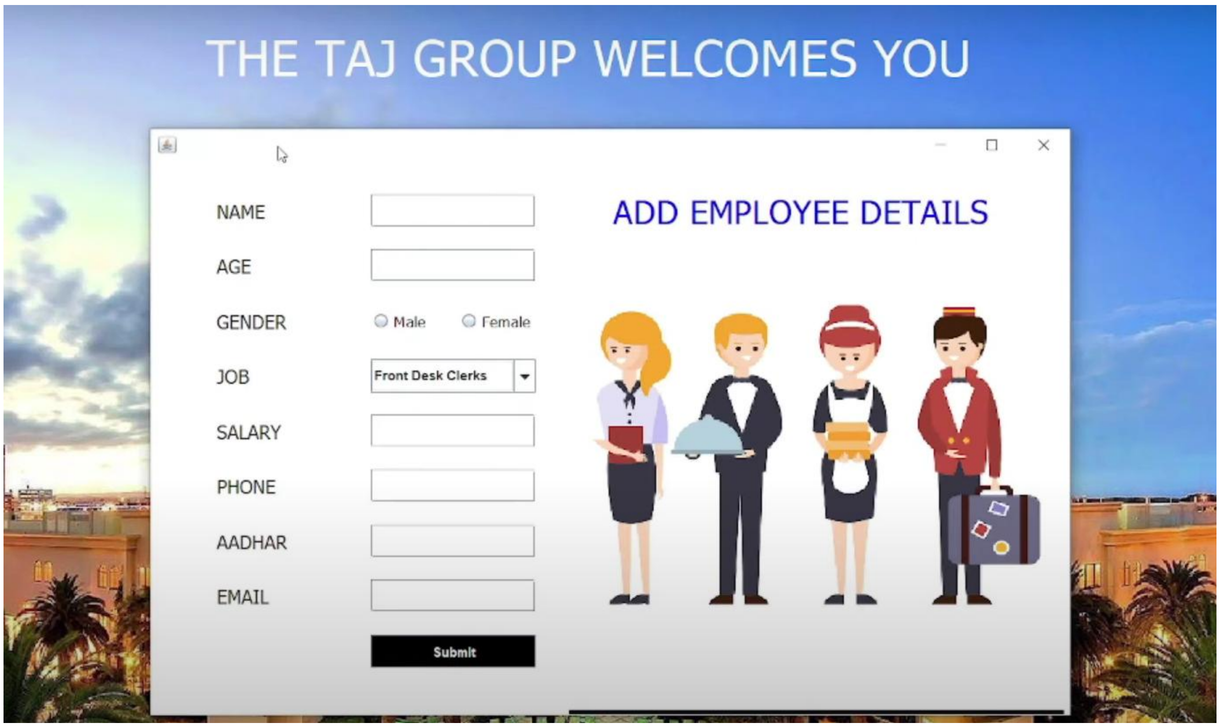


Figure 4.1.4 New Employee Details Adding Page

4.2 Database Design

The database design for a Hotel Management System project using JDBC involves creating a structured schema to efficiently manage hotel operations, including guest bookings, room availability, staff details, billing, and services. The core database typically includes tables such as **Guests**, **Rooms**, **Reservations**, **Staff**, and **Payments**. Each table is designed with unique attributes; for instance, the *Guests* table may have fields like guest_id, name, contact_info, and address, while the *Rooms* table includes room_id, room_type, price_per_night, and availability_status. Relationships between tables are established using primary and foreign keys, enabling seamless integration of data. JDBC (Java Database Connectivity) acts as the bridge between the Java application and the relational database, allowing operations like querying room availability, updating bookings, and generating invoices in real time. The use of normalization ensures the elimination of data redundancy, while indexing key fields enhances query performance, providing a robust and scalable foundation for the system.

Database Code :

```
create database hotelmanagementsystem;

show databases;

use hotelmanagementsystem;

create table login(username varchar(25), password varchar(25));

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

select * from login;

create table employee(name varchar(25), age varchar(10), gender varchar(15), job varchar(30),
salary varchar(15), phone varchar(15), email varchar(40), aadhar varchar(20));

describe employee;

select * from employee;

create table room(roomnumber varchar(10), availability varchar(20), cleaning_status varchar(20),
price varchar(20), bed_type varchar(20));
```

```
select * from room;

update room set availability = 'Available' where roomnumber = '101';

create table driver(name varchar(20), age varchar(10), gender varchar(15), company varchar(20),
branch varchar(20), available varchar(20), location varchar(40));

select * from driver;

ALTER TABLE driver RENAME COLUMN branch TO brand;

create table customer(document varchar(20), number varchar(30), name varchar(30), gender
varchar(15), country varchar(20), room varchar(10), checkintime varchar(80), deposit
varchar(20));

select * from customer;

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

insert into department values('Front Office','500000');
insert into department values('Housekeeping', '40000');
insert into department values('Food and Beverage', '23000');
insert into department values('Kitchen or Food Production', '540000');
insert into department values('Security', '320000');

select * from department;
```

4.3 IMPLEMENTATION (CODE)

```
package hotel.management.system;

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

public class HotelManagementSystem extends JFrame implements ActionListener{

    JLabel l1;
    JButton b1;

    public HotelManagementSystem() {

        setSize(1366,430);
        setLayout(null);
        setLocation(300,300);

            l1 = new JLabel("");
        b1 = new JButton("Next");

        b1.setBackground(Color.WHITE);
        b1.setForeground(Color.BLACK);



        ImageIcon i1  = new
ImageIcon(ClassLoader.getSystemResource("hotel/management/system/icons/first.jpg"));
        Image i3 = i1.getImage().getScaledInstance(1366, 390,Image.SCALE_DEFAULT);
        ImageIcon i2 = new ImageIcon(i3);
        l1 = new JLabel(i2);

        JLabel lid=new JLabel("HOTEL MANAGEMENT SYSTEM");
        lid.setBounds(30,300,1500,100);
        lid.setFont(new Font("serif",Font.PLAIN,70));
        lid.setForeground(Color.red);
        l1.add(lid);

        b1.setBounds(1170,325,150,50);
            l1.setBounds(0, 0, 1366, 390);

        l1.add(b1);
            add(l1);

        b1.addActionListener(this);
        setVisible(true);

        while(true){
            lid.setVisible(false); // lid =  j label
```

```
                try{
                    Thread.sleep(500); //1000 = 1 second
                }catch(Exception e){}
                    lid.setVisible(true);
                try{
                    Thread.sleep(500);
                }catch(Exception e){}
            }
        }

        public void actionPerformed(ActionEvent ae){
                new Login().setVisible(true);
                this.setVisible(false);

        }

        public static void main(String[] args) {
                HotelManagementSystem window = new HotelManagementSystem();
                window.setVisible(true);
            }
    }
```

4.4 Database Connectivity Code

```
    package hotel.management.system;


    import java.sql.*;

    public class conn{
        Connection c;
        Statement s;
        public conn(){
            try{
                Class.forName("com.mysql.jdbc.Driver");
                c =DriverManager.getConnection("jdbc:mysql:///hms","root","");

                s =c.createStatement();


            }catch(Exception e){
                System.out.println(e);
            }
        }
    }
```

5. CONCLUSION

The Hotel Management System project implemented using JDBC provides an efficient and robust solution for managing hotel operations. By leveraging the power of Java and JDBC for database interaction, this system streamlines various processes, including room booking, customer management, billing, and staff coordination.

The integration of a relational database ensures data consistency, integrity, and security, while the dynamic capabilities of JDBC enable seamless interaction between the application and the database. This project demonstrates how modern programming techniques and database technologies can come together to enhance operational efficiency and deliver a better user experience.

Future enhancements could include the addition of advanced features such as real-time room availability tracking, integration with online payment gateways, and support for multi-language interfaces to make the system more versatile and user-friendly. Overall, the project serves as a practical and scalable solution for managing the complex operations of a hotel.

6. REFERENCES

1 **"Java: The Complete Reference" by Herbert Schildt**

- Covers JDBC in detail, with examples on database connectivity.
- Publisher: McGraw Hill.

2 **"Core Java Volume I - Fundamentals" by Cay S. Horstmann and Gary Cornell**

- Comprehensive resource for understanding Java programming and JDBC concepts.
- Publisher: Pearson Education.

3 **"Head First Java" by Kathy Sierra and Bert Bates**

- Beginner-friendly introduction to Java and JDBC integration.
- Publisher: O'Reilly Media.