

HOTEL MANAGEMENT SYSTEM



A PROJECT REPORT

Submitted by

MOHAMED RAYAN N (2303811720521030)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

INFORMATION TECHNOLOGY

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)**

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “**HOTEL MANAGEMENT SYSTEM**” is the bonafide work of **MOHAMED RAYAN N (2303811720521030)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

CGB1201-JAVA PROGRAMMING
Dr. C. SHYAMALA, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR

CGB1201-JAVA PROGRAMMING
Ms. S. UMA MAGESHWARI, M.E.,
SUPERVISOR
ASSISTANT PROFESSOR

SIGNATURE

Dr. C. Shyamala, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of IT

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram-621112.

SIGNATURE

Ms. S. Uma Mageshwari, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram-621112.

Submitted for the viva-voce examination held on 04.12.2024

CGB1201-JAVA PROGRAMMING
Ms. S. UMA MAGESHWARI, M.E.,
INTERNAL EXAMINER
ASSISTANT PROFESSOR

INTERNAL EXAMINER

CGB1201-JAVA PROGRAMMING
Mr. N.M.K. RAMALINGAM, M.E.,
EXTERNAL EXAMINER
ASSISTANT PROFESSOR

PAVAI ENGINEERING COLLEGE, NAMAKKAL.
EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**HOTEL MANAGEMENT SYSTEM**” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING**.

Signature

A handwritten signature in black ink, appearing to be 'Mohamed Rayan N', written over a horizontal line.

MOHAMED RAYAN N

Place: Samayapuram

Date:04.12.2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. C. SHYAMALA, M.E.,Ph.D.**, Head of the department, **INFORMATION TECHNOLOGY** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Ms. S. UMA MAGESHWARI, M.E.**, Department of **COMPUTER SCIENCE AND ENGINEERING**, for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Hotel Management System is a software solution designed to automate and simplify hotel operations. It integrates key functionalities like room booking, user management, payment processing, and reporting into a unified platform. Using object-oriented programming principles, the system ensures efficient handling of data and processes. It enhances customer experience by providing a user-friendly interface and secure payment options, while also offering administrators insightful reports for decision-making. This project aims to improve operational efficiency, reduce manual errors, and deliver a seamless experience for both customers and hotel staff. This system offers a seamless and secure platform for managing hotel operations. The User Management Module handles secure registration and login processes, role-based access control, and user activity tracking to ensure a personalized and secure interaction. The Room Management Module facilitates real-time tracking of room availability, booking, and cancellations, allowing for efficient resource allocation and reducing manual errors.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Hotel Management System is a software solution designed to automate and simplify hotel operations. It integrates key functionalities like room booking, user management, payment processing, and reporting into a unified platform. Using object-oriented programming principles, the system ensures efficient handling of data and processes. It enhances customer experience by providing a user-friendly interface and secure payment options, while also offering administrators insightful reports for decision-making. This project aims to improve operational efficiency, reduce manual errors, and deliver a seamless experience for both customers and hotel staff. This system offers a seamless and secure platform for managing hotel operations. The User Management Module handles secure registration and login processes, role-based access control, and user activity tracking to ensure a personalized and secure interaction. The Room Management Module facilitates real-time tracking of room availability, booking, and cancellations, allowing for efficient resource allocation and reducing manual errors.	PO1 -3 PO2 -3 PO3 -3 PO4 -3 PO5 -3 PO6 -3 PO7 -3 PO8 -3 PO9 -3 PO10 -3 PO11-3 PO12 -3	PSO1 -3 PSO2 -3 PSO3 -3

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 Objective	1
	1.2 Overview	1
	1.3 Java Programming concepts	2
2	PROJECT METHODOLOGY	3
	2.1 Proposed Work	3
	2.2 Block Diagram	4
3	MODULE DESCRIPTION	5
	3.1 User Management Module	5
	3.2 Room Management Module	5
	3.3Payment Module	6
	3.4Reporting Module	6
4	CONCLUSION & FUTURE SCOPE	7
	4.1 Conclusion	7
	4.2 Future Scope	7
	REFERENCES	8
	APPENDIX A (SOURCE CODE)	9
	APPENDIX B (SCREENSHOTS)	12

CHAPTER 1

INTRODUCTION

1.1 Objective:

The primary objective of the Hotel Management System is to create an automated, efficient, and user-friendly platform for managing hotel operations. This system is designed to streamline key processes such as room reservations, user management, payment handling, and reporting. By integrating these functionalities into a unified software solution, it aims to reduce manual workload, minimize errors, and improve overall operational efficiency. Additionally, the system seeks to enhance customer satisfaction by providing a seamless booking experience, secure payment options, and personalized services. For administrators, it offers powerful tools and insights to manage resources effectively, analyze performance, and make data-driven decisions, ultimately contributing to better management and profitability of the hotel.

1.2 Overview:

The Hotel Management System is a software application that integrates core functionalities to manage hotel operations effectively. It includes modules for managing user profiles, tracking room availability, handling secure payments, and generating analytical reports. The system leverages object-oriented programming principles and data management techniques to ensure reliability, scalability, and ease of use. Designed for both customers and administrators, it simplifies the booking process, optimizes resource allocation, and provides actionable insights for better decision-making. The User Management Module allows customers to register, log in, and manage their profiles securely. The Room Management Module tracks room availability, facilitates booking and cancellations, and dynamically updates room status.

1.3 Java Programming Concepts:

The Java Hotel Management System leverages several key Java programming concepts to ensure efficiency, scalability, and maintainability. Object-Oriented Programming (OOP) is at the core, with classes and objects representing real-world entities like Room, User, and Booking. Concepts such as encapsulation protect data through private fields and public access methods, while inheritance promotes code reuse for shared functionality, and polymorphism enables method overloading for flexibility. The system uses the Collection Framework, particularly ArrayList, to store and manipulate dynamic data like rooms and bookings. Control flow statements, including loops and conditional statements, manage program logic, such as validating room availability or processing bookings. Exception handling ensures stability by managing errors with try-catch blocks, while file handling or database connectivity (JDBC) can be used for persistent data storage. Additional features, such as abstraction for defining shared behaviors and multithreading for simultaneous operations, enhance functionality. If required, Swing or JavaFX can be implemented to create user-friendly graphical interfaces. These concepts collectively enable the development of a robust and efficient system. The system employs multithreading to handle concurrent user requests, enhancing performance and responsiveness, especially during peak usage. Database Connectivity (JDBC) is utilized for seamless interaction with the backend database, enabling CRUD operations for managing user profiles, room details, bookings, and transactions. The Collections Framework is used to dynamically store and manage data, employing classes like ArrayList and HashMap to handle lists of bookings and room availability efficiently. Additionally, file handling is integrated for storing logs, reports, and backups securely, while a **Graphical User Interface (GUI)** provides an intuitive and user-friendly interface, ensuring ease of use for both administrators and customers. These concepts collectively ensure a scalable, maintainable, and user-centric application.

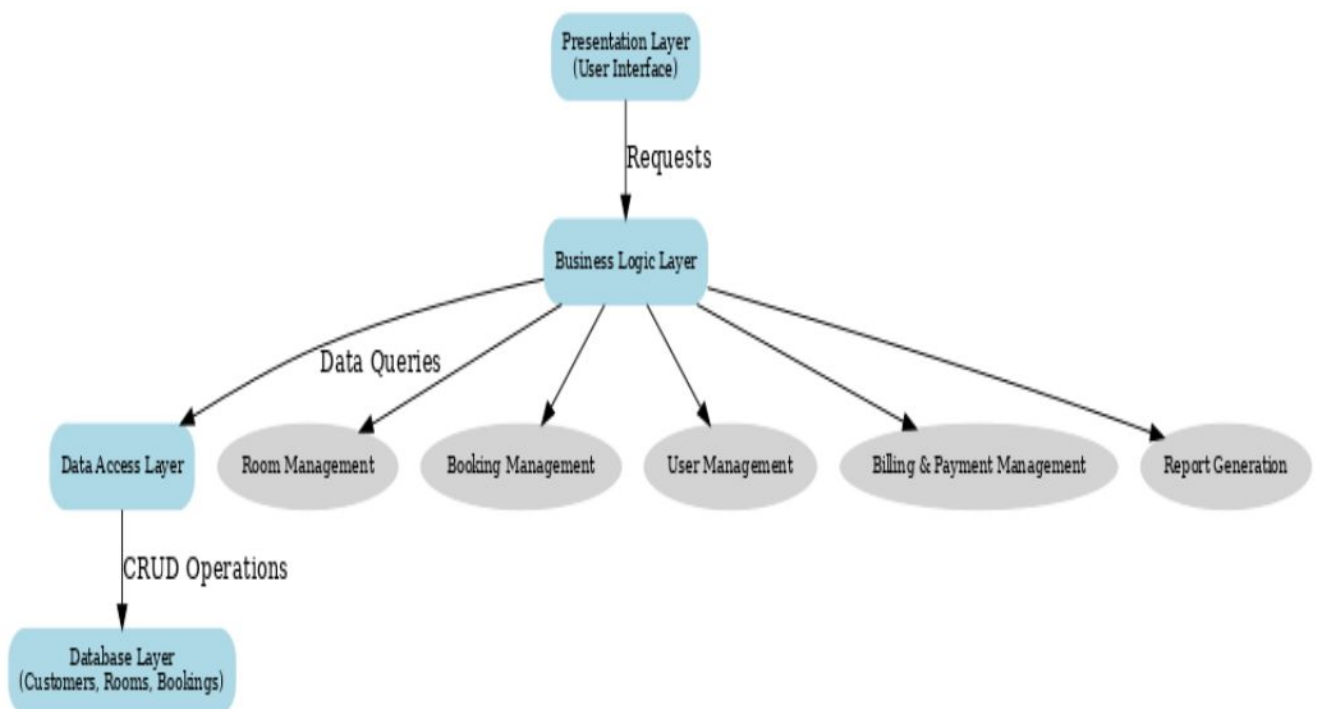
CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

To develop a robust and user-friendly Hotel Management System, begin with gathering requirements. Identify functional requirements such as user registration, room booking, payment processing, and reporting, along with non-functional requirements like security, scalability, and performance. Consult stakeholders to finalize the project scope. Next, move to system design, where you will design the database schema for managing users, rooms, bookings, and payments, and plan the user interface for both customer and admin functionalities. In the project setup phase, set up the development environment with the necessary tools such as IDE, Java SDK, and libraries, and develop core modules using Java, adhering to object-oriented principles. For the database setup, create and configure a relational database (e.g., MySQL), define tables for user data, room details, booking records, and transaction history, and implement CRUD operations for seamless data handling. During the implementation and testing phase, implement individual modules like user management, room management, payment, and reporting, and perform unit testing for each module, followed by integration testing for the entire system. Next, integrate the modules, ensuring seamless communication between them using APIs or shared data structures, and validate workflows to ensure proper module interaction. Implement security measures by applying authentication and authorization mechanisms for secure user access, encrypting sensitive data like passwords and payment details, and conducting security testing to detect and fix vulnerabilities. Conduct thorough testing, including unit testing for individual components, integration testing to ensure all components work together smoothly, and user acceptance testing with real users to gather feedback. Optimize performance by improving code for faster execution, efficient resource usage, and optimizing database queries for quick data retrieval. Finally, deploy the system on a local or cloud server, provide user training and documentation, and perform regular updates and maintenance for smooth operation.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1 User management Module

The **User Management Module** is a crucial component of the Hotel Management System, responsible for handling all user-related operations. It facilitates user registration by allowing new customers or administrators to create accounts with details like name, email, and credentials. The module ensures secure access through authentication mechanisms, including password encryption and validation, to protect user accounts. It also supports profile management, enabling users to view, update, or delete their information as needed, while administrators can manage roles and permissions. With role-based access control, the module differentiates between customers, administrators, and staff, granting access based on user roles. Additionally, it tracks user activities, such as bookings and cancellations, for record-keeping and auditing purposes. This module ensures secure, personalized, and efficient user interactions within the system.

3.2 Room management Module

The **Room Management Module** is a vital part of the Hotel Management System, designed to handle all operations related to room availability and booking. This module allows administrators to add, update, or remove room details, such as room types, pricing, and availability status. It dynamically updates room statuses based on bookings, cancellations, and check-ins/check-outs to ensure real-time accuracy. Customers can browse available rooms, view details, and make bookings through the system. The module also includes functionality to manage special requests, room maintenance schedules, and occupancy reports. By streamlining room operations, this module ensures efficient resource utilization and a seamless booking experience for customers. This module allows administrators to add, update, or remove room details, such as room types, pricing, and availability status. It dynamically updates room statuses based on bookings, cancellations, and check-ins/check-outs to ensure real-time accuracy.

3.3Payment Module

The **Payment Module** in the Hotel Management System manages all financial transactions securely and efficiently. It allows customers to make payments for room bookings using various payment methods such as credit/debit cards, digital wallets, or net banking. The module ensures secure transactions by integrating encryption and secure payment gateways. It also supports features like payment validation, generating invoices, and sending payment confirmations to users via email or SMS. Administrators can track payment records and generate financial reports for analysis. This module is critical for ensuring a smooth and secure payment process, enhancing the overall user experience. The module also includes features for administrators, such as tracking payment histories, monitoring transaction statuses, and generating financial reports for revenue analysis. Refund processing and dispute resolution mechanisms are also supported to address cancellations or errors.

3.4Reporting Module

The **Reporting Module** is an essential part of the Hotel Management System, enabling administrators to generate and manage reports that provide valuable insights into hotel operations. This module consolidates data from various sources, such as bookings, payments, and user activities, to create detailed and customizable reports. Administrators can generate reports for different purposes, such as revenue analysis, room occupancy rates, customer feedback, and staff performance. These reports can be filtered by specific time frames, such as daily, monthly, or yearly, to identify trends and patterns. The module also provides graphical representations, like charts and graphs, for better data visualization, aiding in strategic decision-making. Additionally, the Reporting Module supports export functionality, allowing reports to be downloaded in formats like PDF, Excel, or CSV for easy sharing and record-keeping. Alerts and automated reporting features can be configured to deliver periodic updates directly to the management team. This module ensures that hotel operations are data-driven, helping administrators improve efficiency, optimize resources, and enhance overall customer satisfaction.

CHAPTER 4

CONCLUSION AND FUTURE SCOPE

4.1 CONCLUSION:

In conclusion, the **Hotel Management System** is a comprehensive and efficient solution that addresses the dynamic needs of the hospitality industry. By integrating core functionalities such as user management, room operations, secure payment processing, and insightful reporting, the system streamlines hotel operations and enhances both customer and administrative experiences. Developed using Java programming concepts, the system ensures scalability, reliability, and maintainability. Its modular design minimizes manual errors and operational inefficiencies, while the user-friendly interface caters to a seamless experience for all stakeholders. Through secure data handling and real-time updates, the system fosters trust and convenience. This project not only simplifies complex hotel processes but also lays a foundation for future enhancements, making it a valuable tool for modern hotel management.

4.2 FUTURE SCOPE:

The Hotel Management System has significant potential for future enhancements to meet evolving industry demands. One key area of expansion is integrating advanced technologies such as Artificial Intelligence (AI) for personalized recommendations, customer behavior predictions, and dynamic pricing strategies, as well as Internet of Things (IoT) for smart room controls and real-time monitoring of hotel infrastructure. Additionally, the system can be extended by developing a dedicated mobile application, enabling customers and staff to access features like mobile booking, notifications, and payments on the go. Transitioning to a cloud-based deployment is another promising direction, offering scalability, improved data accessibility, and easier maintenance. Furthermore, integrating global payment gateways and multi-language support would make the system more suitable for international use, catering to a broader customer base. These future improvements will enhance the system's versatility, efficiency, and user experience, making it a more powerful tool for modern hotel management.

REFERENCES:

1. Schildt, H. (2018). *Java: The Complete Reference*. McGraw-Hill Education.
2. "W3Schools: Database Management Basics
3. Oracle Java Documentation: <https://docs.oracle.com/javase/>
4. Eclipse IDE (for Java development),MySQL Database for backend storage, Payment Gateway APIs (e.g., Stripe, PayPal).

APPENDIX A

(Project Source Code)

```
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import javax.swing.JOptionPane;
// Main Class
public class HotelManagementSystem extends Frame implements ActionListener {
    // Declare Components
    Label heading;
    Button roomBtn, bookingBtn, userBtn, billingBtn, reportBtn, exitBtn;
    TextArea outputArea;
    // Sample data storage
    ArrayList<String> rooms = new ArrayList<>();
    ArrayList<String> bookings = new ArrayList<>();
    ArrayList<String> users = new ArrayList<>();

    public HotelManagementSystem() {
        // Frame Setup
        setTitle("Hotel Management System");
        setSize(800, 500);
        setLayout(null);
        setBackground(Color.lightGray);
        // Heading
        heading = new Label("Hotel Management System", Label.CENTER);
        heading.setFont(new Font("Arial", Font.BOLD, 24));
        heading.setBounds(200, 50, 400, 30);
        add(heading);

        // Buttons for modules
        roomBtn = new Button("Room Management");
        roomBtn.setBounds(100, 120, 200, 40);
        roomBtn.addActionListener(this);
        add(roomBtn);
        bookingBtn = new Button("Booking Management");
        bookingBtn.setBounds(100, 180, 200, 40);
        bookingBtn.addActionListener(this);
        add(bookingBtn);

        userBtn = new Button("User Management");
        userBtn.setBounds(100, 240, 200, 40);
        userBtn.addActionListener(this);
        add(userBtn);

        billingBtn = new Button("Billing Management");
        billingBtn.setBounds(100, 300, 200, 40);
        billingBtn.addActionListener(this);
```

```

        add(billingBtn);

        reportBtn = new Button("Report Generation");
        reportBtn.setBounds(100, 360, 200, 40);
        reportBtn.addActionListener(this);
        add(reportBtn);

        exitBtn = new Button("Exit");
        exitBtn.setBounds(350, 420, 100, 40);
        exitBtn.addActionListener(this);
        add(exitBtn);
    // Output Area
        outputArea = new TextArea("", 5, 50,
TextArea.SCROLLBARS_VERTICAL_ONLY);
        outputArea.setBounds(350, 120, 400, 300);
        outputArea.setEditable(false);
        add(outputArea);

        // Add Window Listener for Closing
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

    // Event Handling
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == roomBtn) {
            manageRooms();
        } else if (e.getSource() == bookingBtn) {
            manageBookings();
        } else if (e.getSource() == userBtn) {
            manageUsers();
        } else if (e.getSource() == billingBtn) {
            manageBilling();
        } else if (e.getSource() == reportBtn) {
            generateReports();
        } else if (e.getSource() == exitBtn) {
            System.exit(0);
        }
    }

    // Room Management Module
    private void manageRooms() {
        String roomNumber = JOptionPane.showInputDialog(this, "Enter Room Number:");
        String roomType = JOptionPane.showInputDialog(this, "Enter Room Type
(Single/Double):");
        rooms.add("Room " + roomNumber + ": " + roomType);
        outputArea.setText("Room Management:\n");
    }

```

```

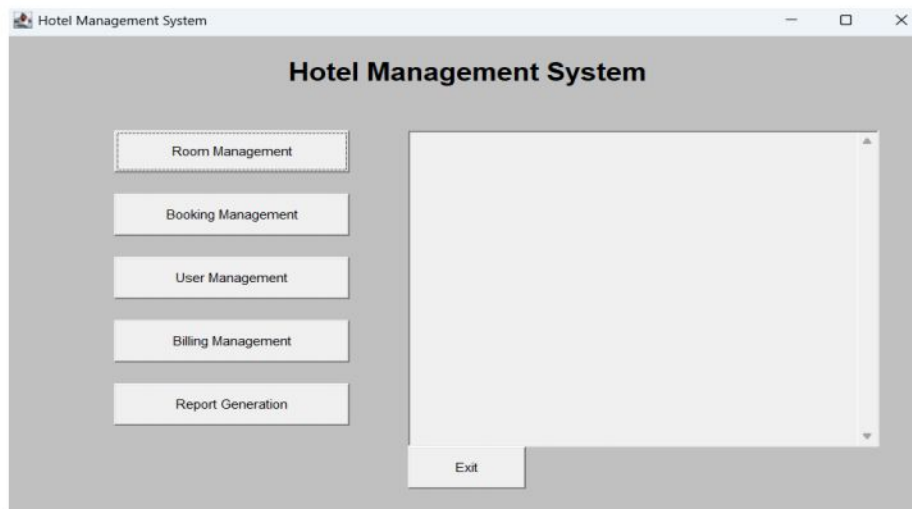
        for (String room : rooms) {
            outputArea.append(room + "\n");
        }
    }
// Booking Management Module
private void manageBookings() {
    String roomNumber = JOptionPane.showInputDialog(this, "Enter Room Number:");
    String guestName = JOptionPane.showInputDialog(this, "Enter Guest Name:");
    bookings.add("Booking: Room " + roomNumber + " - " + guestName);
    outputArea.setText("Booking Management:\n");
    for (String booking : bookings) {
        outputArea.append(booking + "\n");
    }
}
// User Management Module
private void manageUsers() {
    String userName = JOptionPane.showInputDialog(this, "Enter User Name:");
    String userType = JOptionPane.showInputDialog(this, "Enter User Type
(Customer/Staff):");
    users.add("User: " + userName + " - " + userType);
    outputArea.setText("User Management:\n");
    for (String user : users) {
        outputArea.append(user + "\n");
    }
}
// Billing Management Module
private void manageBilling() {
    outputArea.setText("Billing Management:\n");
    outputArea.append("Invoice for Booking:\n");
    outputArea.append("Room: $100\n");
    outputArea.append("Total: $100\n");
}

// Report Generation Module
private void generateReports() {
    outputArea.setText("Report Generation:\n");
    outputArea.append("Total Bookings: " + bookings.size() + "\n");
    outputArea.append("Total Rooms: " + rooms.size() + "\n");
    outputArea.append("Total Users: " + users.size() + "\n");
}

// Main Method
public static void main(String[] args) {
    HotelManagementSystem hms = new HotelManagementSystem();
    hms.setVisible(true);
}
}

```

APPENDIX B (SCREENSHOT)



CREATING A ROOM:


This is an 'Input' dialog box with a green header bar and a close button (X) in the top right corner. It contains a green square icon with a white question mark. The text 'Enter Room Number:' is followed by a text input field containing the value '101'. At the bottom are two buttons: 'OK' and 'Cancel'.

This is an 'Input' dialog box with a green header bar and a close button (X) in the top right corner. It contains a green square icon with a white question mark. The text 'Enter Room Type (Single/Double):' is followed by a text input field containing the value 'Single'. At the bottom are two buttons: 'OK' and 'Cancel'.

BOOKING A ROOM:

This is an 'Input' dialog box with a green header bar and a close button (X) in the top right corner. It contains a green square icon with a white question mark. The text 'Enter Room Number:' is followed by a text input field containing the value '101'. At the bottom are two buttons: 'OK' and 'Cancel'.

Input ×


 Enter Guest Name:

A

OK Cancel

BILLING:

Input ×

 Enter User Name:

A

OK Cancel

Billing Management:
Invoice for Booking:
Room: \$100
Total: \$100