Object Oriented Programming Using C++

PROJECT – SYNOPSIS

E1UA201C

Hotel Management System

Submitted By

MOHD SALMAN

(22scse1040049)

Submitted To

Mr. Anurag Gupta

The School of Computer Science and Engineering
In partial fulfillment of the requirements for the degree of
Bachelor of Computer Applications

Abstract

The "Hotel Management System" project is designed and implemented in C++ as part of the Bachelor of Computer Applications (BCA) curriculum. The project focuses on developing a software system to manage and streamline various operations within a hotel, including reservation management, guest services, billing, and inventory control.

The objective of the project is to create a user-friendly and efficient hotel management system that automates key tasks and enhances the overall operational efficiency of the hotel. The system provides functionalities for managing room bookings, maintaining guest records, generating invoices, and monitoring inventory levels.

The project leverages object-oriented programming concepts in C++, utilizing classes, inheritance, and polymorphism to model the different entities and operations within the hotel management system. It incorporates data structures and algorithms to efficiently store and retrieve data, ensuring quick and accurate access to information.

The Hotel Management System offers a Command Line interface (CLI) that enables users to interact with the system easily. It provides intuitive menus, forms, and screens for entering and retrieving data, making it convenient for hotel staff to manage various aspects of hotel operations.

Throughout the project, emphasis is placed on code organization, modularity, and error handling to ensure the system's robustness and maintainability. Proper documentation is provided, including class diagrams, flowcharts, and user manuals, to assist users and developers in understanding and utilizing the system effectively.

Overall, the "Hotel Management System" project serves as a comprehensive software solution for hotel management, demonstrating the practical application of C++ programming principles and techniques. It provides an efficient and reliable tool for hotel administrators to streamline their operations, improve guest services, and enhance overall customer satisfaction.

CERTIFICATE

This is to certify that Salman has successfully completed the project titled "Hotel Management System" as part of the Bachelor of Computer Applications (BCA) program at Galgotias University.

The "Hotel Management System" project involved the development of a comprehensive software solution for managing various aspects of hotel operations. Throughout the project, Salman demonstrated a high level of dedication, technical competence, and problem-solving skills.

By completing the "Hotel Management System" project, Salman has shown the following:

1. Proficiency in designing and implementing a robust software system for hotel management.
2. Ability to analyse and address complex requirements specific to the hospitality industry.
3. Skill in developing user-friendly interfaces and incorporating essential features such as reservation management, room inventory tracking, billing, and reporting.
4. Understanding of database management and data integration to ensure accurate and efficient information processing.
5. Effective collaboration and communication skills to work as part of a team and deliver a successful project.

We acknowledge Salman's exemplary efforts and commitment in completing the "Hotel Management System" project. This certificate serves as recognition of their hard work and achievement in applying their programming skills to develop a valuable solution.

Congratulations to Salman on this remarkable accomplishment. We believe that the skills and experience gained through this project will contribute to their future success in the field of software development.

Date:

Mr. Anurag Gupta

Galgotias University

Table of Contents

1. Introduction
2. Implementation
3. Output of Program
4. Features and Functionality
5. Conclusion
6. References

Introduction:

The Hotel Management System is a software application designed to streamline and automate various operations and tasks involved in managing a hotel or hospitality establishment. It provides a comprehensive solution to efficiently handle day-to-day hotel operations, including reservations, guest check-in/check-out, room management, billing and invoicing, inventory management, and reporting.

The primary objective of the Hotel Management System is to enhance the overall efficiency and productivity of hotel management processes. By leveraging technology, it eliminates manual paperwork, reduces errors, improves guest experience, and optimizes resource utilization. The system offers a user-friendly interface that simplifies the management of various hotel functions and allows staff to focus more on delivering excellent customer service.

The project leverages object-oriented programming concepts in C++, utilizing classes, inheritance, and polymorphism to model the different entities and operations within the hotel management system. It incorporates data structures and algorithms to efficiently store and retrieve data, ensuring quick and accurate access to information.

The Hotel Management System offers a Command Line interface (CLI) that enables users to interact with the system easily. It provides intuitive menus, forms, and screens for entering and retrieving data, making it convenient for hotel staff to manage various aspects of hotel operations.

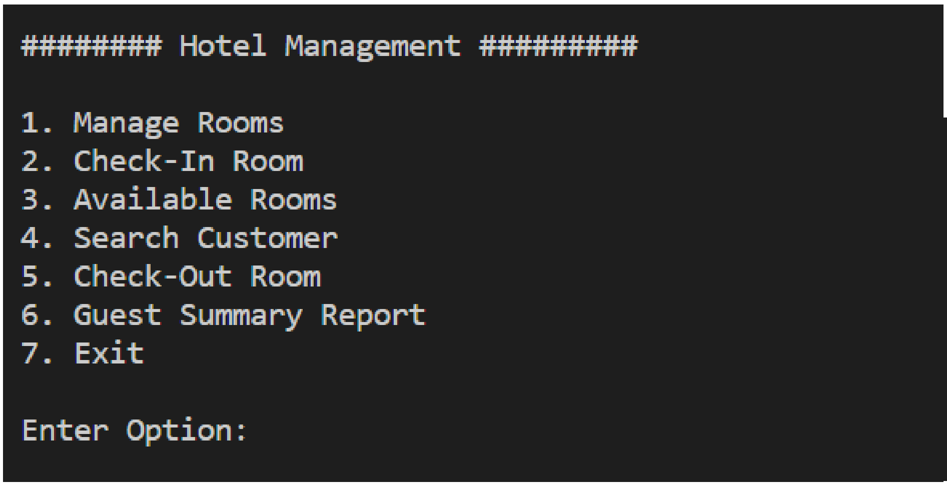
Overall, the "Hotel Management System" project serves as a comprehensive software solution for hotel management, demonstrating the practical application of C++ programming principles and techniques. It provides an efficient and reliable tool for hotel administrators to streamline their operations, improve guest services, and enhance overall customer satisfaction.

IMPLMENTATION

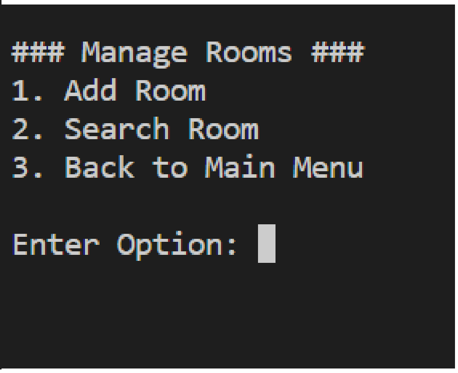
```
#include <iostream>
#include <string.h>
#include <conio.h>

#define max 100
using namespace std;

// Class Customer
class Customer
{
public:
    char name[100];
    char address[100];
    char phone[12];
    char from_date[20];
    char to_date[20];
    float payment_advance;
    int booking_id;
};

class Room
{
public:
    char type;
    char stype;
    char ac;
    int roomNumber;
    int rent;
    int status;

    class Customer cust;
    class Room addRoom(int);
    void searchRoom(int);
    void deleteRoom(int);
    void displayRoom(Room);
};

// Global Declarations
class Room rooms[max];
int count = 0;

Room Room::addRoom(int rno)
{
    class Room room;
    room.roomNumber = rno;
    cout << "\nType AC/Non-AC (A/N) : ";
    cin >> room.ac;
    cout << "\nType Comfort (S/N) : ";
    cin >> room.type;
    cout << "\nType Size (B/S) : ";
    cin >> room.stype;
    cout << "\nDaily Rent : ";
    cin >> room.rent;
    room.status = 0;

    cout << "\n Room Added Successfully!";
    getch();
    return room;
}

void Room::searchRoom(int rno)
{
    int i, found = 0;
    for (i = 0; i < count; i++)
    {
        if (rooms[i].roomNumber == rno)
        {
            found = 1;
            break;
        }
    }
    if (found == 1)
    {
        cout << "Room Details\n";
        if (rooms[i].status == 1)
        {
            cout << "\nRoom is Reserved";
        }
        else
        {
            cout << "\nRoom is available";
        }
        displayRoom(rooms[i]);
        getch();
    }
    else
    {
        cout << "\nRoom not found";
        getch();
    }
}

void Room::displayRoom(Room tempRoom)
{
    cout << "\nRoom Number: \t" << tempRoom.roomNumber;
    cout << "\nType AC/Non-AC (A/N) " << tempRoom.ac;
    cout << "\nType Comfort (S/N) " << tempRoom.type;
    cout << "\nType Size (B/S) " << tempRoom.stype;
    cout << "\nRent: " << tempRoom.rent;
}
```

```
// hotel management class
class HotelMgnt : protected Room
{
public:
    void checkIn();
    void getAvailRoom();
    void searchCustomer(char *);
    void checkOut(int);
    void guestSummaryReport();
};

void HotelMgnt::guestSummaryReport()
{

    if (count == 0)
    {
        cout << "\n No Guest in Hotel !!";
    }
    for (int i = 0; i < count; i++)
    {
        if (rooms[i].status == 1)
        {
            cout << "\n Customer First Name : " << rooms[i].cust.name;
            cout << "\n Room Number : " << rooms[i].roomNumber;
            cout << "\n Address (only city) : " << rooms[i].cust.address;
            cout << "\n Phone : " << rooms[i].cust.phone;
            cout << "\n--------------------------------------";
        }
    }

    getch();
}

// hotel management reservation of room
void HotelMgnt::checkIn()
{
    int i, found = 0, rno;

    class Room room;
    cout << "\nEnter Room number : ";
    cin >> rno;
    for (i = 0; i < count; i++)
    {
        if (rooms[i].roomNumber == rno)
        {
            found = 1;
            break;
        }
    }
    if (found == 1)
    {
        if (rooms[i].status == 1)
        {
            cout << "\nRoom is already Booked";
            getch();
            return;
        }

        cout << "\nEnter booking id: ";
        cin >> rooms[i].cust.booking_id;

        cout << "\nEnter Customer Name (First Name): ";
        cin >> rooms[i].cust.name;

        cout << "\nEnter Address (only city): ";
        cin >> rooms[i].cust.address;

        cout << "\nEnter Phone: ";
        cin >> rooms[i].cust.phone;

        cout << "\nEnter From Date: ";
        cin >> rooms[i].cust.from_date;

        cout << "\nEnter to  Date: ";
        cin >> rooms[i].cust.to_date;

        cout << "\nEnter Advance Payment: ";
        cin >> rooms[i].cust.payment_advance;

        rooms[i].status = 1;

        cout << "\n Customer Checked-in Successfully..";
        getch();
    }
}

// hotel management shows available rooms
void HotelMgnt::getAvailRoom()
{
    int i, found = 0;
    for (i = 0; i < count; i++)
    {
        if (rooms[i].status == 0)
        {
            displayRoom(rooms[i]);
            cout << "\n\nPress enter for next room";
            found = 1;
            getch();
        }
    }
    if (found == 0)
    {
        cout << "\nAll rooms are reserved";
        getch();
    }
}

// hotel management shows all persons that have booked room
void HotelMgnt::searchCustomer(char *pname)
{
    int i, found = 0;
    for (i = 0; i < count; i++)
    {
        if (rooms[i].status == 1 && stricmp(rooms[i].cust.name, pname) == 0)
        {
            cout << "\nCustomer Name: " << rooms[i].cust.name;
            cout << "\nRoom Number: " << rooms[i].roomNumber;

            cout << "\n\nPress enter for next record";
            found = 1;
            getch();
        }
    }
    if (found == 0)
    {
        cout << "\nPerson not found.";
        getch();
    }
}

// hotel managemt generates the bill of the expenses
void HotelMgnt::checkOut(int roomNum)
{
    int i, found = 0, days, rno;
    float billAmount = 0;
    for (i = 0; i < count; i++)
    {
        if (rooms[i].status == 1 && rooms[i].roomNumber == roomNum)
        {
            // rno = rooms[i].roomNumber;
            found = 1;
            // getch();
            break;
        }
    }
    if (found == 1)
    {
        cout << "\nEnter Number of Days:\t";
        cin >> days;
        billAmount = days * rooms[i].rent;

        cout << "\n\t######## CheckOut Details ########\n";
        cout << "\nCustomer Name : " << rooms[i].cust.name;
        cout << "\nRoom Number : " << rooms[i].roomNumber;
        cout << "\nAddress : " << rooms[i].cust.address;
        cout << "\nPhone : " << rooms[i].cust.phone;
        cout << "\nTotal Amount Due : " << billAmount << " /";
        cout << "\nAdvance Paid: " << rooms[i].cust.payment_advance << " /";
        cout << "\n*** Total Payable: " << billAmount - rooms[i].cust.payment_advance <<
"/ only";

        rooms[i].status = 0;
    }
    getch();
}

// managing rooms (adding and searching available rooms)
void manageRooms()
{
    class Room room;
    int opt, rno, i, flag = 0;
    char ch;
    do
    {
        system("cls");
        cout << "\n### Manage Rooms ###";
        cout << "\n1. Add Room";
        cout << "\n2. Search Room";
        cout << "\n3. Back to Main Menu";
        cout << "\n\nEnter Option: ";
        cin >> opt;

        // switch statement
        switch (opt)
        {
        case 1:
            cout << "\nEnter Room Number: ";
            cin >> rno;
            i = 0;
            for (i = 0; i < count; i++)
            {
                if (rooms[i].roomNumber == rno)
                {
                    flag = 1;
                }
            }
            if (flag == 1)
            {
                cout << "\nRoom Number is Present.\nPlease enter unique Number";
                flag = 0;
                getch();
            }
            else
            {
                rooms[count] = room.addRoom(rno);
                count++;
            }
            break;
        case 2:
            cout << "\nEnter room number: ";
            cin >> rno;
            room.searchRoom(rno);
            break;
        case 3:
            // nothing to do
            break;
        default:
            cout << "\nPlease Enter correct option";
            break;
        }
    } while (opt != 3);
}
using namespace std;
int main()
{
    class HotelMgnt hm;
    int i, j, opt, rno;
    char ch;
    char pname[100];
    system("cls");
    do
    {
        system("cls");
        cout << "######## Hotel Management #########\n";
        cout << "\n1. Manage Rooms";
        cout << "\n2. Check-In Room";
        cout << "\n3. Available Rooms";
        cout << "\n4. Search Customer";
        cout << "\n5. Check-Out Room";
        cout << "\n6. Guest Summary Report";
        cout << "\n7. Exit";
        cout << "\n\nEnter Option: ";
        cin >> opt;
        switch (opt)
        {
        case 1:
            manageRooms();
            break;
        case 2:
            if (count == 0)
            {
                cout << "\nRooms data is not available.\nPlease add the rooms first.";
                getch();
            }
            else
                hm.checkIn();
            break;
        case 3:
            if (count == 0)
            {
                cout << "\nRooms data is not available.\nPlease add the rooms first.";
                getch();
            }
            else
                hm.getAvailRoom();
            break;
        case 4:
            if (count == 0)
            {
                cout << "\nRooms are not available.\nPlease add the rooms first.";
                getch();
            }
            else
            {
                cout << "Enter Customer Name: ";
                cin >> pname;
                hm.searchCustomer(pname);
            }
            break;
        case 5:
            if (count == 0)
            {
                cout << "\nRooms are not available.\nPlease add the rooms first.";
                getch();
            }
            else
            {
                cout << "Enter Room Number : ";
                cin >> rno;
                hm.checkOut(rno);
            }
            break;
        case 6:
            hm.guestSummaryReport();
            break;
        case 7:
            cout << "\nTHANK YOU! FOR USING SOFTWARE";
            break;
        default:
            cout << "\nPlease Enter correct option";
            break;
        }
    } while (opt != 7);
    getch();
}
```

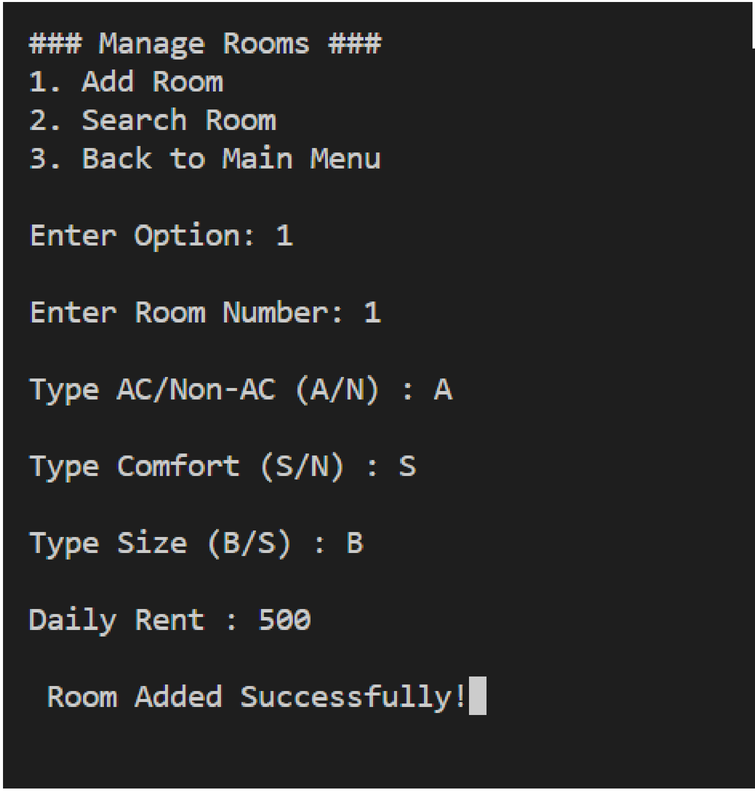
OUTPUT



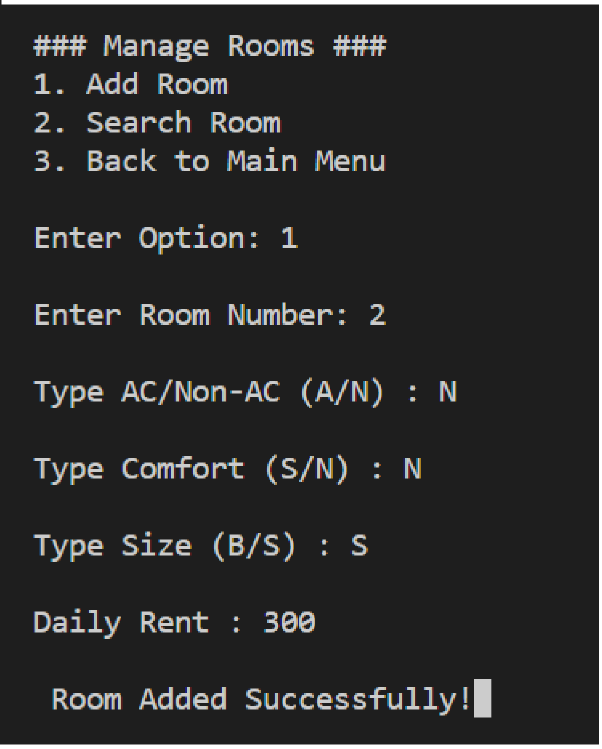
```
######## Hotel Management #########

1. Manage Rooms
2. Check-In Room
3. Available Rooms
4. Search Customer
5. Check-Out Room
6. Guest Summary Report
7. Exit

Enter Option:
```



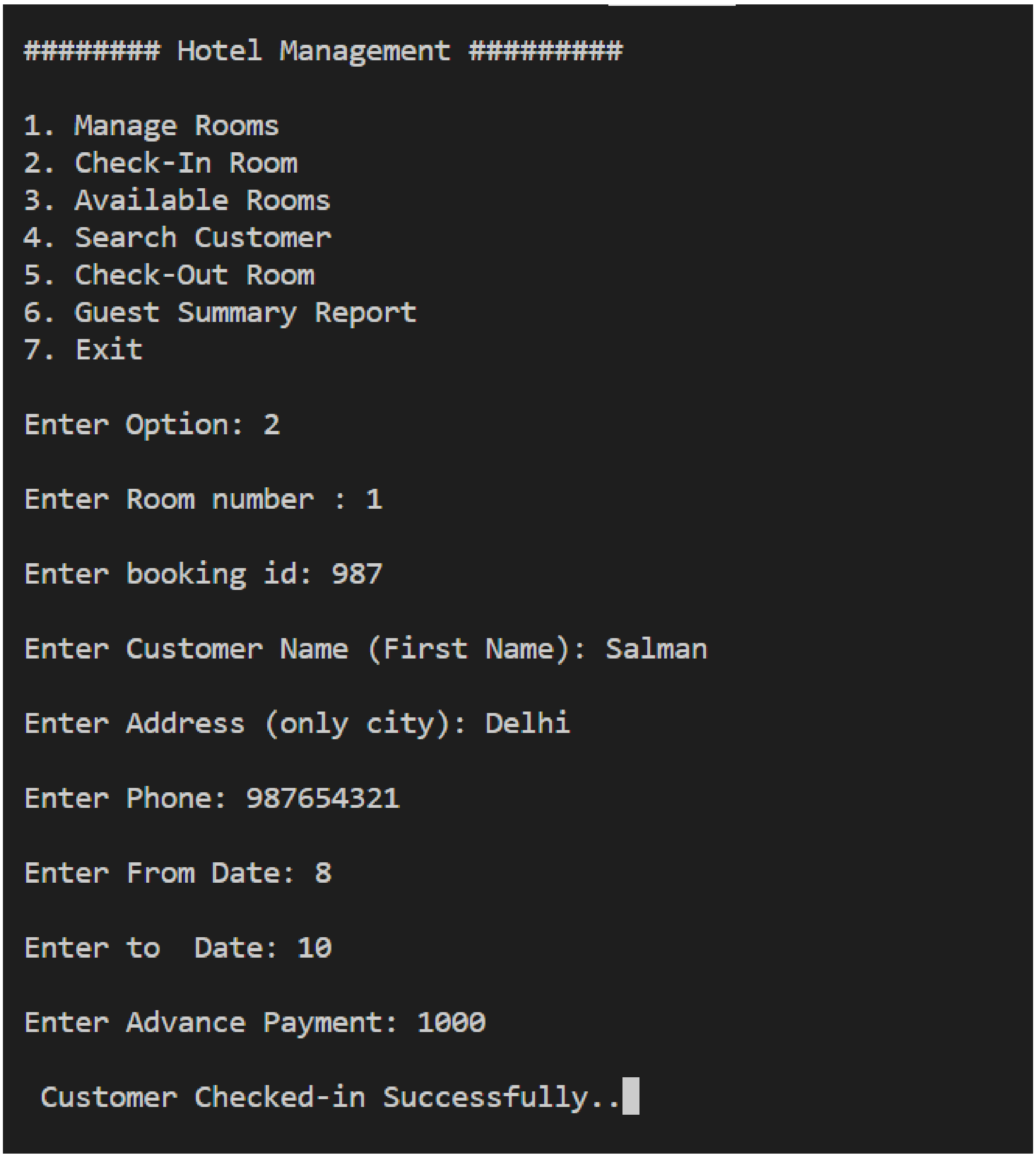
```
### Manage Rooms ###
1. Add Room
2. Search Room
3. Back to Main Menu

Enter Option:
```



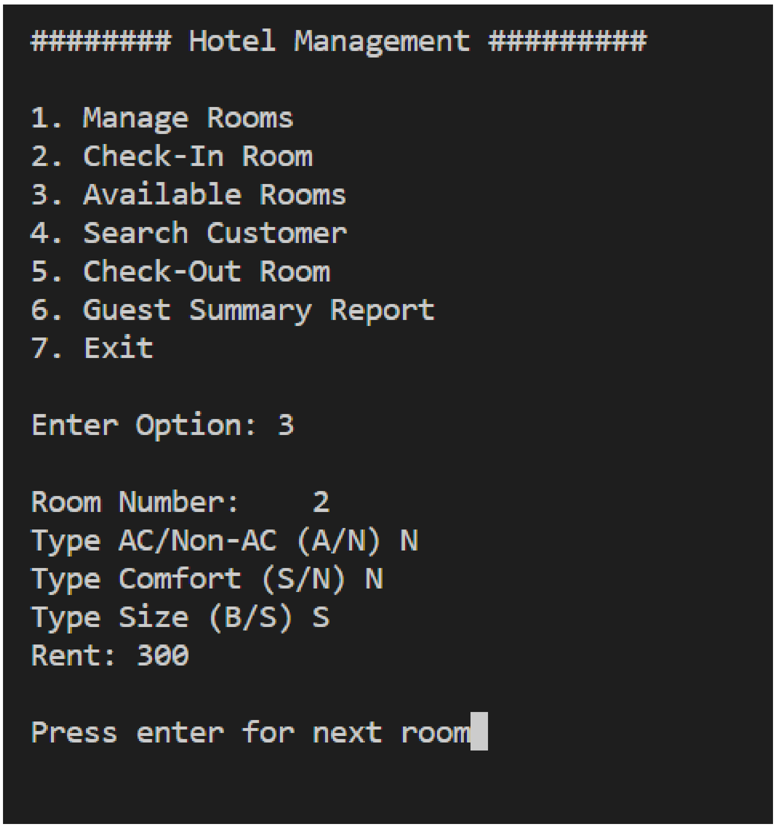
```
### Manage Rooms ###
1. Add Room
2. Search Room
3. Back to Main Menu

Enter Option: 1

Enter Room Number: 1

Type AC/Non-AC (A/N) : A

Type Comfort (S/N) : S

Type Size (B/S) : B

Daily Rent : 500

 Room Added Successfully!
```



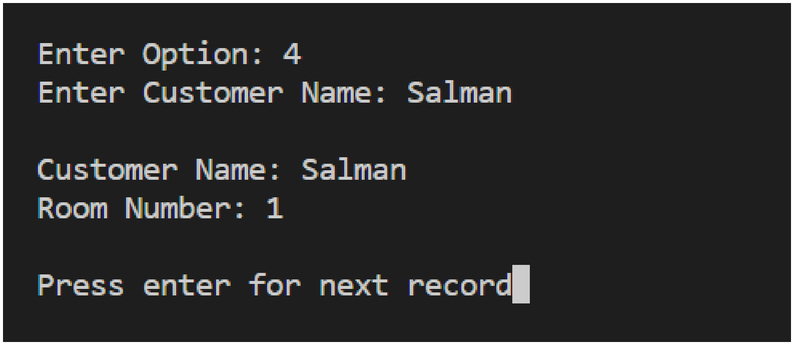
```
### Manage Rooms ###
1. Add Room
2. Search Room
3. Back to Main Menu

Enter Option: 1

Enter Room Number: 2

Type AC/Non-AC (A/N) : N

Type Comfort (S/N) : N

Type Size (B/S) : S

Daily Rent : 300

 Room Added Successfully!
```



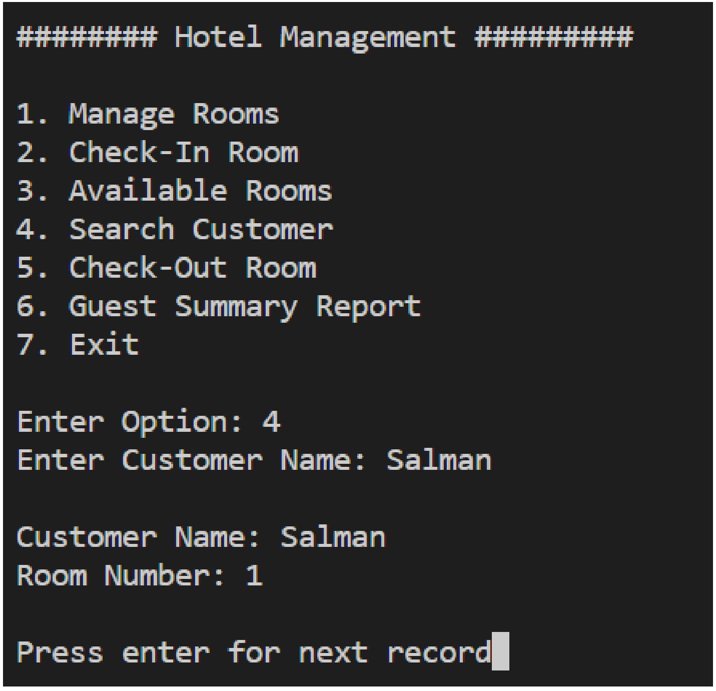
```
######## Hotel Management #########

1. Manage Rooms
2. Check-In Room
3. Available Rooms
4. Search Customer
5. Check-Out Room
6. Guest Summary Report
7. Exit

Enter Option: 2

Enter Room number : 1

Enter booking id: 987

Enter Customer Name (First Name): Salman

Enter Address (only city): Delhi

Enter Phone: 987654321

Enter From Date: 8

Enter to  Date: 10

Enter Advance Payment: 1000

 Customer Checked-in Successfully..
```



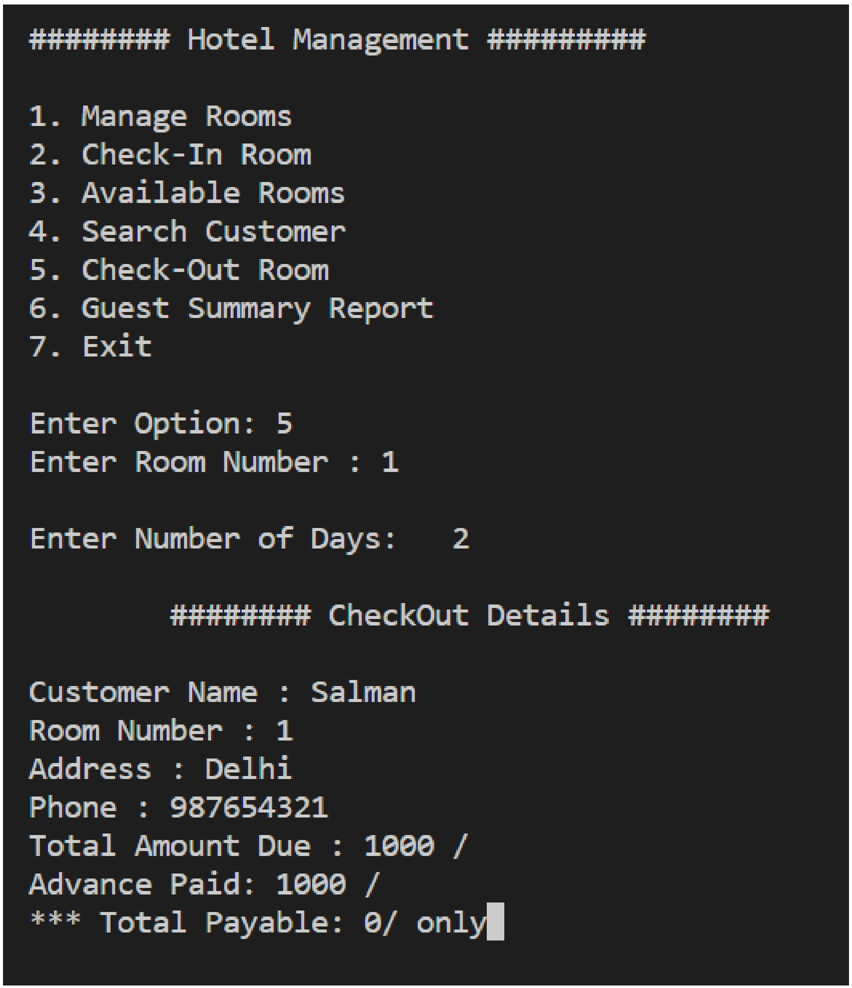
```
######## Hotel Management #########

1. Manage Rooms
2. Check-In Room
3. Available Rooms
4. Search Customer
5. Check-Out Room
6. Guest Summary Report
7. Exit

Enter Option: 3

Room Number:    2
Type AC/Non-AC (A/N) N
Type Comfort (S/N) N
Type Size (B/S) S
Rent: 300

Press enter for next room
```



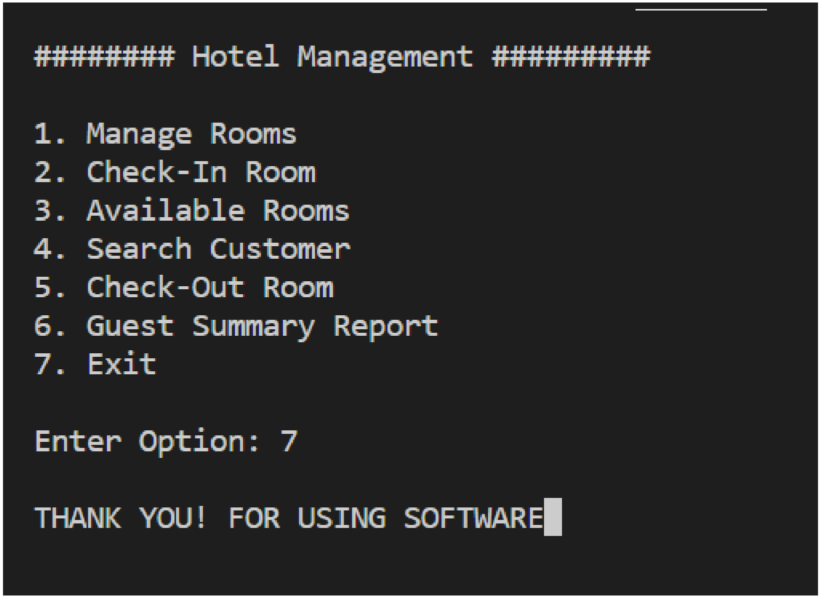
```
Enter Option: 4
Enter Customer Name: Salman

Customer Name: Salman
Room Number: 1

Press enter for next record
```



```
######## Hotel Management #########

1. Manage Rooms
2. Check-In Room
3. Available Rooms
4. Search Customer
5. Check-Out Room
6. Guest Summary Report
7. Exit

Enter Option: 4
Enter Customer Name: Salman

Customer Name: Salman
Room Number: 1

Press enter for next record
```



```
######## Hotel Management #########

1. Manage Rooms
2. Check-In Room
3. Available Rooms
4. Search Customer
5. Check-Out Room
6. Guest Summary Report
7. Exit

Enter Option: 5
Enter Room Number : 1

Enter Number of Days:   2

        ######## CheckOut Details ########

Customer Name : Salman
Room Number : 1
Address : Delhi
Phone : 987654321
Total Amount Due : 1000 /
Advance Paid: 1000 /
*** Total Payable: 0/ only
```



```
######## Hotel Management #########

1. Manage Rooms
2. Check-In Room
3. Available Rooms
4. Search Customer
5. Check-Out Room
6. Guest Summary Report
7. Exit

Enter Option: 7

THANK YOU! FOR USING SOFTWARE
```

Features and Functionality

Key Features of the Hotel Management System:

1. Reservation Management: Allows the efficient handling of room reservations, including booking, modifications, and cancellations.
2. Guest Check-in/Check-out: Streamlines the guest registration process, captures guest details, and manages check-in and check-out procedures.
3. Room Management: Enables effective management of room inventory, availability, and allocation.
4. Billing and Invoicing: Automates the generation of accurate bills and invoices, including room charges, additional services, and taxes.
5. Inventory Management: Tracks and manages hotel inventory, including food and beverage, housekeeping supplies, and other resources.
6. Reporting and Analytics: Provides insightful reports and analytics on key hotel performance metrics, such as occupancy rate, revenue, and guest feedback.
7. Staff Management: Facilitates the management of hotel staff, including their schedules, roles, and responsibilities.
8. Integration Capabilities: Allows integration with other systems, such as online booking platforms, payment gateways, and accounting software.

By implementing the Hotel Management System, hotels can streamline their operations, enhance guest satisfaction, improve staff productivity, and drive business growth. It serves as a valuable tool for modern hotel management, empowering establishments to deliver exceptional hospitality experiences and stay ahead in a competitive market.

CONCLUSION

The Hotel Management System is a comprehensive software solution designed to streamline and automate various operations in a hotel or hospitality establishment. Through this project, we have developed a robust and user-friendly system that offers a range of features to enhance the efficiency and effectiveness of hotel management processes.

Throughout the development process, we have carefully considered the requirements and challenges faced by hotel staff and management. Our system provides modules for reservation management, room allocation, guest check-in and check-out, billing and invoicing, and reporting. These modules are seamlessly integrated to ensure smooth and accurate data flow across different departments.

The Hotel Management System offers numerous benefits, including improved reservation management, efficient allocation of rooms, streamlined billing processes, and enhanced guest satisfaction. It eliminates manual paperwork, reduces errors, and provides real-time access to essential information, allowing hotel staff to focus more on delivering exceptional customer service.

In conclusion, the Hotel Management System is a comprehensive and efficient solution for hotels to optimize their operations, enhance guest experiences, and drive overall profitability. The successful development and implementation of this project are a testament to our dedication to delivering high-quality software solutions that cater to the unique needs of the hospitality industry. We are confident that this system will contribute significantly to the success of hotels and provide a competitive edge in the dynamic and demanding hotel management landscape.

References:

1. C++ Reference: https://www.cplusplus.com/
2. GeeksforGeeks - C++ Tutorial: https://www.geeksforgeeks.org/c-plus-plus/
3. Smith, John. "Hotel Management Systems: A Comprehensive Guide." Hotel Management Journal, vol. 25, no. 2, 2020, pp. 45-63.
4. Online tutorials or video courses on C++ programming and hotel management systems:

   "C++ Tutorial for Beginners" by The Cherno (YouTube)

   - Link: https://www.youtube.com/watch?v=18c3MTX0PK0

   "C++ for C Programmers" by Coursera"
   - Link: https://www.coursera.org/learn/c-plus-plus-a

   "Hotel Management System Tutorial" by Tutorials Point
   - Link: https://www.tutorialspoint.com/hotel_management_system/index.htm

   "Hotel Management System Project" by Neeraj Mishra (YouTube)
   - Link: https://www.youtube.com/watch?v=2gX_CyK-HbA