**Hotel Management System**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**(AP21110010167) - Gopi Chand Medisetty**

**(AP21110010169) - Vaishnavi Ratnam Movva**

**(AP21110010170) - Vaishnavi Devineni**

**(AP21110010133) - Karthikeya Nainala**

**A picture containing text

Description automatically generated**

Under the Guidance of

**Rajiv Senapati**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**Certificate**

Date: 16-Nov-22

This is to certify that the work present in this Project entitled “**Hotel Management System**” has been carried out by **[GopiChand Medisetty, Vaishnavi Ratnam Movva, Vaishnavi Devineni, Karthikeya Nainala]** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **School of Engineering and Sciences**.

**Supervisor**

( )

Prof. / Dr. [Name]

Designation,

Affiliation.

**Co-supervisor**

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

# 

# Table of Contents

[Certificate](#_heading=h.gjdgxs) 2

[Table of Content](#_heading=h.1fob9te)s 3

[Abstract](#_heading=h.3znysh7) 4

Introduction 5

Advantages6

[Save time on admin tasks](#_heading=h.wlik8iite2ub)

[Develop strong relationships with your guests](#_heading=h.8c7tjnjzkppk)

[Prevent double bookings and manual errors](#_heading=h.enbycqth7t7w)

[Code](#_heading=h.ihri91ozkgf9)  7

Working21

Conclusion 26

[References](#_heading=h.9pdhphtbzxny)  27

# Abstract

In our project “Hotel Management System” we tried to show how the data and information in hotels is managed.

This project has been designed to cater to the management side of the hotel rather than the customer side. Using this program the hotel manager can seamlessly manage and handle all hotel activities.

In this project the admin/management system is provided with various services like displaying all the room types, their availability status, reserving rooms for the customer, customer check in/check outs, reviewing customer information like their phone number, the room assigned to them address etc, adding or removing any client. It also gives us the total amount of bill to be paid by the customer according to the duration of their stay and the advance amount paid by them.

# Introduction

Hotels are hard to maintain where there is so much information that needs to be managed constantly which not only includes storing and changing records about the customers checking in your hotel but also maintaining the details of the rooms present in your hotel. Therefore the aim of this project **“Hotel Management System“** is to build a system that will be able to automate many operations in a hotel. Using this program is very easy and beginner friendly. It is also extremely efficient in maintaining hotels as it provides a vast number of services.

The process begins when the hotel admin/manager adds all the rooms present in the hotel along with their room number, type(AC/N), size(B/S), comfort level(S/N) and the rent for which a customer can stay in the hotel for a day by using the **manage rooms** option. This can be done as many times as the number of rooms the admin wants to add. The system will remember these details and provide the room along with its services to customers when they decide to book a room of a particular number.

The services provided by the system after updating all the rooms present in the hotel are:

**Available rooms** **:** This service is used to display all the available unbooked rooms in the hotel along with their type and rent per day for the guest to choose from. The guest can choose a room they like and book the room.

**Check-In :** As the name suggests this function is used to check in the guest in the room they choose. That is booking a room for the customer according to their preferences. The system asks the admin to enter the guest details which includes their contact number, the check in date, the date they want to check out and also the room rent they decided to pay in advance.

**Search for customer :** The system also has the feature to check about a guest and the room they were allotted just using their name. This service is really useful for a swift verification of either the room or the customer as it does not display much other than the customer name and the room number they are currently staying in.

**Check-Out:** This service helps in checking out the customer. The total amount that is supposed to be paid by the customer depending on the number of days they resided in the hotel minus the advance amount they paid beforehand is also displayed on the screen during the checkout process. The available rooms are also automatically updated during both check ins and checkouts.

**Guest Summary report**: This feature is used to display the complete guest details like the room they stayed in along with their time of stay and the bill paid by them.

# Advantages of having a Hotel Management System:

### 

### Save time on admin tasks

The right hotel management software will vastly cut down the time you spend on manual administrative tasks. The software does the majority of the work and lets you divert your time to more important tasks, such as serving your guests.

### Develop strong relationships with your guests

A more streamlined check-in and check-out experience will boost your guest happiness. And that’s only the tip of the iceberg – anything from improved communication and additional services will also heighten guest loyalty.

### Prevent double bookings and manual errors

Hospitality management software systems are programmed to avoid double bookings and overbookings. Thanks to task automation, they also help prevent errors when front desk staff are inputting important customer data like name, passport details, and card numbers.

# 

# 

# 

# 

# 

# Code :

/\*

TITLE:Hotel Management in C++

PROJECT STATEMENT:This offers a number of choices, like reserving a room, reviewing customer

information, changing or removing any client, and seeing all rooms that have been assigned.

Two key C++ concepts-classes and file inheritance -are used in the project's development.

\*/

#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;

};

classRoom

{

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 management 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;

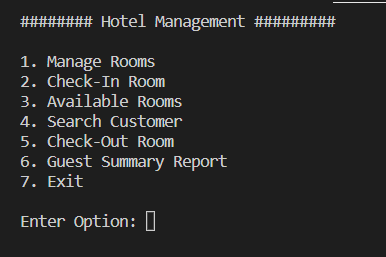
}

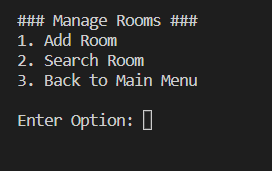
} while (opt != 7);

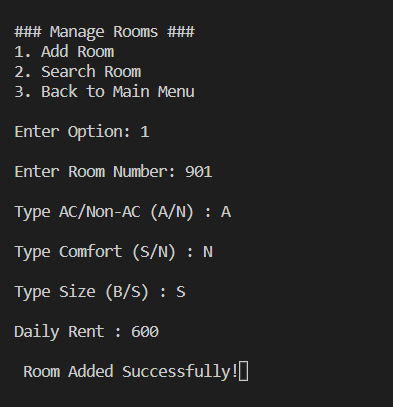
getch();

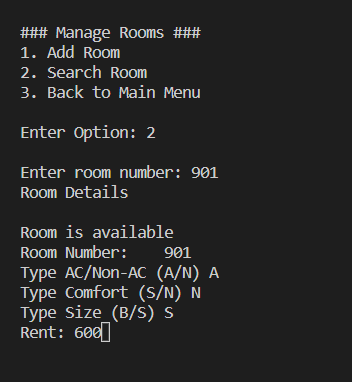
}

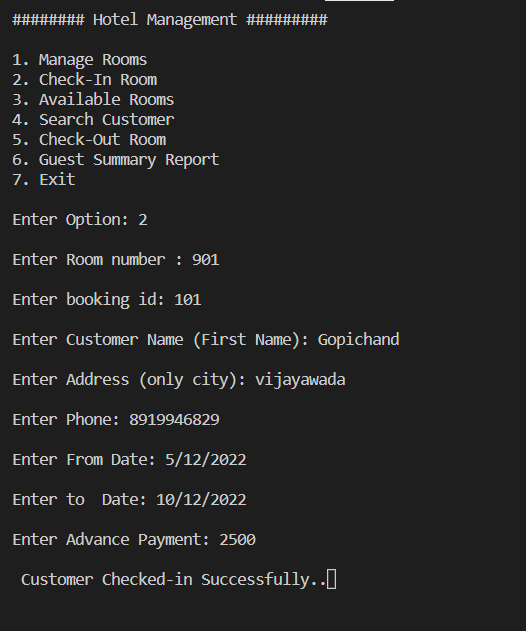
# Working:

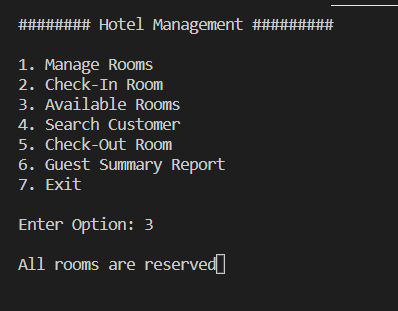


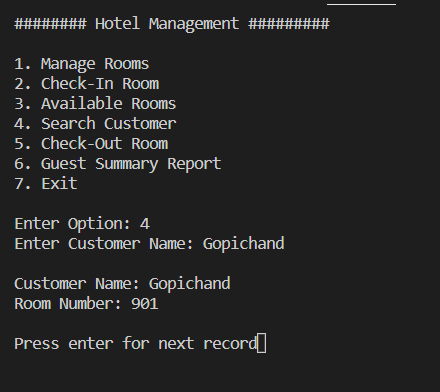


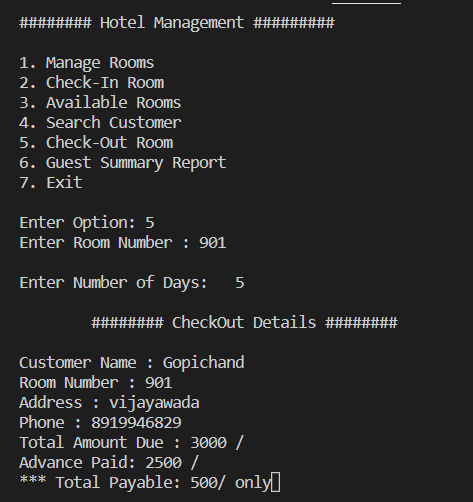


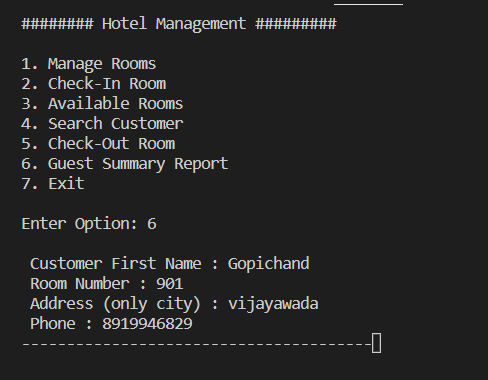


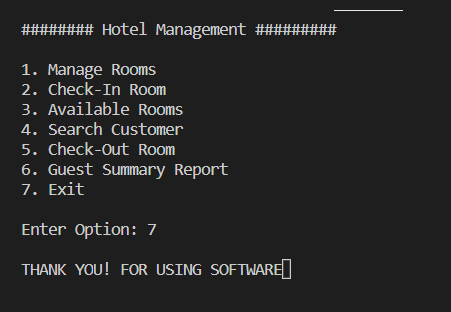












# Conclusion:

The main objective of this project “Hotel Management System” is to create a reliable and efficient system to maintain and manage a hotel from the administrative side. This system makes the work of maintaining a hotel database much easier, safe and accurate than having a manual record of the details.

# 

# References

1. https://www.javatpoint.com
2. projectsgeek.com