A Project proposal on

# “Hotel Management System”

### Submitted to

Amrutvahini Polytechnic, Sangamner Department of Information Technology

In partial fulfilment of the requirement for the diploma in Information Technology

Submitted By

56] Kadlag Karan Rajendra (2100800930)

60] Khandagale Anurag Balasaheb (2100800935)

**Under The Guidance Of Prof. Kadlag.S.U**



### Amrutvahini Polytechnic, Sangamner

**(Approved by AICTE, NEW DELHI and affiliated To MSBTE) 2022-2023**

### Amrutvahini Polytechnic, Sangamner Department of Information Technology



**CERTIFICATE**

**This is to that the project report entitled,**

# “Hotel Management System”

### It is benefited work carrier out by,

56] Kadlag Karan Rajendra (2100800930)

60] Khandagale Anurag Balasaheb (2100800935)

### In partial fulfilment of the requirement for the diploma in Information Technology

During the academic year 2022-2023

**Prof.Kadlag.S.U Prof.ChaudhariN.K (Project Guide) (H.O.D) IT**

**ACKNOWLEDGEMENT**

We have taken lots of efforts in this project. However, it would have been possible without the kind support and help of many individuals and organization. We would to kind to extend our sincere thanks to all of them.

First and foremost we want to thanks Prof. Chaudhari N.K H.O.D (Information Technology) Amrutvahini Polytechnic

Sangamner for giving us an opportunity to work on this project.

We are highly indebted to Prof. Kadlag S.U. (Project Guide) for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in the project.

We would like to express our gratitude towards our parents and members of Information Technology department for their kind

Co-operation and encouragement which helps us in completion of this micro project.

Our thanks and appreciations also go to our colleagues in developing the micro-project and people who have willingly helped us with their abilities.

56] Kadlag Karan Rajendra (2100800930)

60] Khandagale Anurag Balasaheb (2100800935)

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Contents** | **Page.no** |
| **1.** | Rationale | 5 |
| **2.** | Aims and Benefits | 5 |
| **3.** | Course Outcomes | 5 |
| **4.** | Literature Review | 6 |
| **5.** | Actual Methodology Followed | 7-22 |
| **6.** | Conclusion | 23 |
| **7.** | Reference | 23 |
| **8.** | Actual resource used | 23 |
| **9.** | Skills Developed/learning outcomes of the  micro-project | 23 |

A Project Report On

# “Hotel Management System”

#### RATIONALE:-

The project for **Hotel Management system in Java.** The Title of the Micro project is to "**Develop Hotel Management Application in Java".**

In this project users can perform general hotel management operations related to hotel like Manage Room, check in room, Available rooms, search customer information, check out rooms etc. In hotel management system project in Java, file handling use to perform various functions like adding customer details and add room etc.

This system provides various options like booking a room, checking customer details, editing or deleting any customer, checking all allotted rooms. The project is developed using two important Java concepts that are classes, objects and file handling.

#### AIMS AND BENEFITS:- AIMS:-

Create a **Hotel Management System** using **Java**.

#### BENEFITS:-

* + Develop strong relationships with your guests.
  + Increase your online visibility.
  + Implement an effective revenue management system.
  + Increase bookings.
  + Accurate daily reports.
  + Prevent double bookings and manual errors.
  + Analyse your customer base.
  + Hotel management system online provide room booking, staff management and other necessary hotel management feature.

#### COURSE OUTCOMES:-

|  |  |
| --- | --- |
| CI401.1 | Implement programs using Object Oriented methodology and basics of java |
| CI401.2 | Apply concept of inheritance for code reusability |
| CI401.3 | Implement programs using multithreading and Exception Handling |
| CI401.4 | Execute programs using graphics and applet |
| CI401.5 | Use concept of file handling I/O and file streams |

1. **LITERATURE REVIEW:-**

The program allows the user to view all the rooms, add customers to a room, display empty rooms, delete customers from a room, find a room from a customer name, store program array data into a text file, load program data back from the file, view rooms ordered alphabetically by name, and quit the program.

The program uses a simple menu-driven approach where the user is presented with a list of options to choose from. The program also uses an array to represent the rooms in the hotel, where each element of the array represents a room in the hotel.

The program starts by initializing the hotel array with null values for each room. The main method then calls the menu method which displays the menu options and waits for user input. Based on the user's choice, the program calls the appropriate method to perform the desired action.

Overall, the code above provides a basic implementation of a Hotel Management System using Java. However, there are some limitations to this implementation, such as the lack of error handling for invalid user input and the lack of data validation for customer names. Additionally, the program does not handle multiple customers sharing a room, which may be a requirement for a real-world hotel management system.

#### ACTUAL METHODOLOGY FOLLOWED:-

* **What is Hotel Management System?**

A hotel management system is a software application that is used to manage various operations of a hotel such as room reservations, check-ins, check-outs, room inventory, billing, guest management, and more. The system automates many of the tasks that were previously done manually, allowing hotel staff to work more efficiently and effectively. The hotel management system can also help improve the guest experience by providing them with easy access to information, allowing them to make reservations, and more. Overall, a hotel management system can help streamline operations, improve customer service, and increase revenue.

#### Introduction:-

The project for **Hotel Management system in Java.** The Title of the Micro project is to "**Develop Hotel Management Application in Java".**

In this project users can perform general hotel management operations related to hotel like Manage Room, check in room, Available rooms, search customer information, check out rooms etc. In hotel management system project in Java, file handling use to perform various functions like adding customer details and add room etc.

This system provides various options like booking a room, checking customer details, editing or deleting any customer, checking all allotted rooms. The project is developed using two important Java concepts that are classes, objects and file handling.

On execution of this code users get a menu with 6 points 1 to 6 and then program runs according to user selection. There is output at the end pages of file.

This is a Java program for a hotel management system. The program provides various options for the user to manage hotel rooms and customer data. The program uses an array to store data for the rooms in the hotel, and provides options to add, delete, and view customer data for each room.

The program starts by initializing an array called 'hotel' with a length of 10. Each element of the array is set to 'e' to indicate that the room is empty. A menu

is then displayed, which provides options for the user to manage the hotel rooms.

The program uses a static String array called hotel to store the names of the customers occupying a room. The program initializes all the elements in the array with the value "e" to indicate that all the rooms are empty.

The menu() method displays the menu of options to the user and uses a do-while loop to ensure that the user enters a valid option. The program uses a switch statement to execute the selected option.

* Packets Used:-

1. import java.io.\*;
2. import java.util.InputMismatchException;
3. import java.util.Scanner;

* The menu with six points are :-
* View Room
* Add Customer
* Display Empty Rooms
* Delete Customer
* Find Room
* Store Data
* Retrieve Data
* Alphabetical Order
* Exit

#### View Room

The first option is to view all the rooms. This option loops through the hotel array and displays the room number and the name of the customer occupying

the room. If the room is empty, it displays a message indicating that the room is empty.

The viewRooms() method displays the room number and the current owner's name for each occupied room in the hotel array. If a room is empty, the method displays the room number and the message "is empty".

**To View the data** of room such as Add Customer, Store Data, Retrieve Data. }

In option of view room that is user can Data/room details such as Room no is empty or occupied by Costumer.

View Room is for viewing the details for room.

#### Add Customer

The addCustomer() method prompts the user to enter a room number and then checks if the room is already occupied or if the room number is within the range of valid room numbers. If the input is invalid, the program displays an error message and prompts the user to enter a valid room number.

The second option is to add a customer to a room. The user is prompted to enter a room number between 0 and 9. If the room is already occupied, the program displays a message indicating that the room is occupied by a customer. If the room is empty, the user is prompted to enter the name of the customer. The name of the customer is then stored in the corresponding element of the 'hotel' array.

In this options is for check in customer. In this option we have to enter customer details.

In option of Add Customer the user can check in Customer Data

After Adding Costumer Details there are 2 options:-

* + Add Customer
  + menu
* If user have to continue next entry of customer User can press ‘y’ it will call addCustomer();
* If user want to exit after adding customer User can press ‘y’ it will call addCustomer();

#### Display Empty Rooms

The displayEmptyRooms() method displays the room number for all the empty rooms in the hotel array.

The third option is to display all empty rooms. This option loops through the 'hotel' array and displays the room number for each empty room.

This option Display the available Rooms

#### Delete customer

The deleteCustomer() method prompts the user to enter a room number and then checks if the room is occupied. If the room is occupied, the method removes the customer's name from the hotel array and sets the corresponding element to the value "e" to indicate that the room is empty.

The fourth option is to delete a customer from a room. The user is prompted to enter a room number. If the room is empty, the program displays a message indicating that the room is already empty. If the room is occupied, the program clears the corresponding element of the 'hotel' array.

This options is for check in customer. In this option user have to enter customer room no that user have to vacate.

#### Find Room

The findRoom() method prompts the user to enter a customer name and then searches the hotel array for the room number corresponding to that name. If the name is not found, the method displays an error message.

The fifth option is to find a room by customer name. The user is prompted to enter a customer name. The program then loops through the 'hotel' array and displays the room number for any room that is occupied by a customer with the given name.

This option is for Find out the customer.

In this we have enter Customer name to find out thee name.

#### Store Data

The storeData() method writes the contents of the hotel array to a text file named Data.txt.

The sixth option is to store the data from the 'hotel' array into a text file. This option is use for to save the data in data.txt file.

If we exit from execution we can use data that we saved in data.txt using Retrieve data function

#### Retrieve Data

The retrieveData() method reads the contents of the HotelData.txt file and stores them in the hotel array.

The seventh option is to load data from a text file back into the 'hotel' array. The program reads the data from the 'hotel.txt' file and stores it in the 'hotel' array.

In this method we used file Handling.

If we exit from execution we can use data that we saved in data.txt using Retrieve data function

#### Alphabetical Order

The alphabeticalOrder() method sorts the hotel array alphabetically by customer name and displays the room number and the corresponding customer name for each occupied room.

The eighth option is to view rooms in alphabetical order by customer name. The program creates a new array called 'hotelCopy' and copies the elements of the 'hotel' array into it. It then sorts the 'hotelCopy' array in alphabetical order by customer name and displays the room number and customer name for each room.

#### Exit

The ninth option is used for exiting the program.

The program uses exception handling to catch input errors such as entering a non-integer value or a value outside the range of 0-9 for the room number.

Overall, this is a basic hotel management system program that provides a user- friendly interface for managing hotel rooms and customer data. The program uses a combination of loops, conditional statements, and arrays to implement the hotel management system functionalities. The program also uses

exception handling to catch invalid user inputs and prevent the program from crashing.

#### CODE:-

import java.io.\*;

import java.util.InputMismatchException; import java.util.Scanner;

public class HotelManagementSystem {

//Created static variables including the array and scanner,

// so it can be used in all the static methods without being declared again static String choice;

static String answer;

static String customerName; static int roomNumber = 0;

static String[] hotel = new String[10];

static Scanner input = new Scanner(System.in); public static void main(String[] args) {

initialize(hotel); menu();

}

private static void initialize(String hotelRef[]) {

//Hotel array will only have Null characters at the beginning.

//Assigning a String value of "e" to all of it's elements. for (int x = 0; x < 10; x++) {

hotelRef[x] = "e";

}

}

public static void menu() {

System.out.println("======================================================");

System.out.println("\* Hotel Management System \*"); System.out.println("======================================================");

System.out.println("\* V. View all the rooms \*");

System.out.println("\* A. Add customer to room \*");

System.out.println("\* E. Display Empty rooms \*");

System.out.println("\* D. Delete customer from room \*"); System.out.println("\* F. Find room from customer name \*"); System.out.println("\* S. Store program array data into a text file \*"); System.out.println("\* L. Load program data back from the file \*"); System.out.println("\* O. View rooms Ordered alphabetically by name \*"); System.out.println("\* Q. Quit Program \*");

System.out.println("======================================================");

System.out.println("");

System.out.println("Choose one of the options from above. (E.g: Type 'V' to view all the rooms)"); do {

System.out.println();

System.out.print("Choice : "); choice = input.next();

String selection = choice.toLowerCase(); //This will convert the input value to lowercase. this will help avoid case sensitive issues.

switch (selection) { case "v":

viewRooms();

break; case "a":

addCustomer(); break;

case "e": displayEmptyRooms(); break;

case "d": deleteCustomer(); break;

case "f": findRoom(); break;

case "s":

storeData(); break;

case "l": retrieveData(); break;

case "o": alphabeticalOrder(); break;

case "q": System.out.println("Thanks"); break;

default:

System.out.println("Invalid input! Please Enter one of these letters: V,A,E,D,F,S,L,O,Q");

}

//viewRooms();

}

while (!(choice.equalsIgnoreCase("v") || choice.equalsIgnoreCase("a") || choice.equalsIgnoreCase("e") || choice.equalsIgnoreCase("d") ||

choice.equalsIgnoreCase("f") || choice.equalsIgnoreCase("s") || choice.equalsIgnoreCase("l") || choice.equalsIgnoreCase("o") ||

choice.equalsIgnoreCase("q"))); //condition to only let valid range of inputs through.

}

private static void viewRooms() {

while (roomNumber < 10) { for (int x = 0; x < 10; x++) {

//This will display the room number and the current owner's name if (!(hotel[x].equals("e"))) {

System.out.println("Room No. " + x + " is occupied by Mr. " + hotel[x]);

//This will display the rooms which are currently Empty

} else {

System.out.println("Room No. " + x + " is empty");

}

}

break;//This statement is just being used to break the loop

}

System.out.println(""); menu();

}

private static void addCustomer() {

boolean invalidRoomNumber;//declaring a boolean value so it is easier to break or catch data from a loop do {

invalidRoomNumber = false;

try {

System.out.println("Enter room number (0-9)"); roomNumber = input.nextInt();

//if the room is already occupied this message will get printed if (!(hotel[roomNumber].equals("e"))) {

invalidRoomNumber = true;

System.out.println("This room is occupied by: Mr. " + hotel[roomNumber]); System.out.println("");

//if the room is empty and the input value is within the range it accepts the input

} else if (roomNumber >= 0 && roomNumber < 10) { invalidRoomNumber = false;

//if the input exceeds the range then this error message will be displayed

} else {

invalidRoomNumber = true;

System.out.println("Invalid input! Please Enter a value between 0-9");

System.out.println("");

}

//if the input is not an integer value then this will catch it

} catch (InputMismatchException e) { invalidRoomNumber = true;

System.out.println("Invalid input! Please Enter a value between 0-9"); System.out.println("");

input.next();

//if the input is out of the range of the hotel array this will catch it

} catch (IndexOutOfBoundsException e) { invalidRoomNumber = true;

System.out.println("Invalid input! Please Enter a value between 0-9"); System.out.println("");

input.next();

}

} while (invalidRoomNumber);//these steps will follow if all of the above is valid System.out.println("Enter the name of the customer :"); /\*+ roomNumber +\*/ customerName = input.next();

hotel[roomNumber] = customerName;

//this will let you choose whether to add more data or not do {

System.out.println("Do you want to continue adding records?(Y/N)"); answer = input.next();

String selection = answer.toLowerCase();

switch (selection) { case "y":

addCustomer();

case "n": System.out.println(""); menu();

}

} while (!(answer.equalsIgnoreCase("y") || answer.equalsIgnoreCase("n")));

}

private static void displayEmptyRooms() {

//this method will display all the empty rooms for (int x = 0; x < 10; x++) {

if (hotel[x].equals("e")) { System.out.println("room " + x + " is empty");

}

}

System.out.println(""); menu();

}

private static void deleteCustomer() {

boolean invalidInput; do {

invalidInput = false; try {

System.out.println("please enter the Room's number which you want to vacate"); roomNumber = input.nextInt();

//if the hotel room is not empty then this will delete the customer from that room if (!(hotel[roomNumber].equals("e"))) {

invalidInput = false; hotel[roomNumber] = "e";

//if the room is already empty this message will be displayed

} else {

invalidInput = true;

System.out.println("Room " + roomNumber + " is already Empty"); System.out.println("");

}

//if the input is not an integer value then this will catch it

} catch (InputMismatchException e) { invalidInput = true;

System.out.println("Invalid input! Please Enter a value between 0-9"); System.out.println("");

input.next();

//if the input is out of the range of the hotel array this will catch it

} catch (IndexOutOfBoundsException e) { invalidInput = true;

System.out.println("Invalid room number. Please enter a value between 0-9"); input.next();

}

} while (invalidInput);//This will print the room's number which has been successfully vacated System.out.println("Room " + roomNumber + " has successfully been vacated");

System.out.println(""); menu();

}

private static void findRoom() {

System.out.println("Please enter the name of the customer"); boolean found = false;

String find = input.next();

for (int n = 0; n < 10; n++) {

//used equalsIgnoreCase to avoid case sensitive issues while searching for a customer

//this method will find the room's number which is currently being occupied by the mentioned customer if (hotel[n].equalsIgnoreCase(find)) {

found = true;

System.out.println("Mr. " + find + " is staying in room No. " + n); System.out.println("");

menu();

}

}

//this will let ou know if the customer is not there in the database if (found == false) {

System.out.println(find + " doesn't exist on our database"); System.out.println("");

menu();

}

}

private static void alphabeticalOrder() {

int index = 0;

String[] names = new String[10];

//copy hotel array data to names array for (int x = 0; x < 10; x++) {

names[x] = hotel[x].toLowerCase();//used this to avoid case sensitive issues.

}

//used Bubble sort

for (int i = 0; i < names.length - 1; i++) { for (int j = i + 1; j < names.length; j++) {

if (names[j].compareTo(names[i]) < 0) { String temp = names[j];

names[j] = names[i]; names[i] = temp;

}

}

}

//This will add the list of names in ascending order in our new array for (int x = 0; x < names.length; x++) {

if (!(names[x].equals("e"))) {

for (int i = 0; i < hotel.length; i++) {

if (hotel[i].toLowerCase().equals(names[x])) { index = i;

}

}

System.out.println("Mr. " + names[x] + " is staying in room No. " + index);

}

}

System.out.println(""); menu();

}

private static void storeData() { try {

//saving Data and overwriting

BufferedWriter bw = new BufferedWriter(new FileWriter("data.txt", false)); for (int x = 0; x < hotel.length; x++) {

String file; file = hotel[x];

// writes Empty room if it find "e" on the array if (file.equals("e")) {

bw.write("Empty Room " + x);

// writes the name of the customer from the array

} else {

bw.write(file);

}

bw.newLine(); //Line Seperator bw.flush(); //Flushes the stream.

}

//will catch this exception if the Text file is not found

} catch (IOException e) { System.err.println("File not found!");

}

//message to show the user that the array data has been saved to a Text file successfully System.out.println("Data successfully saved!");

System.out.println(""); menu();

}

private static void retrieveData() {

try {

//reading Data from the Text File

BufferedReader reader = new BufferedReader(new FileReader("data.txt"));

for (int x = 0; x < hotel.length; x++) {

String read = reader.readLine(); //Reads String value stored in the Text File if (read.equals("Empty Room " + x)) {

read = "e";

}

//Stored the data in the Hotel Array hotel[x] = read;

}

//will catch this exception if the Text file is not found

} catch (IOException e) { System.out.println("File not found!");

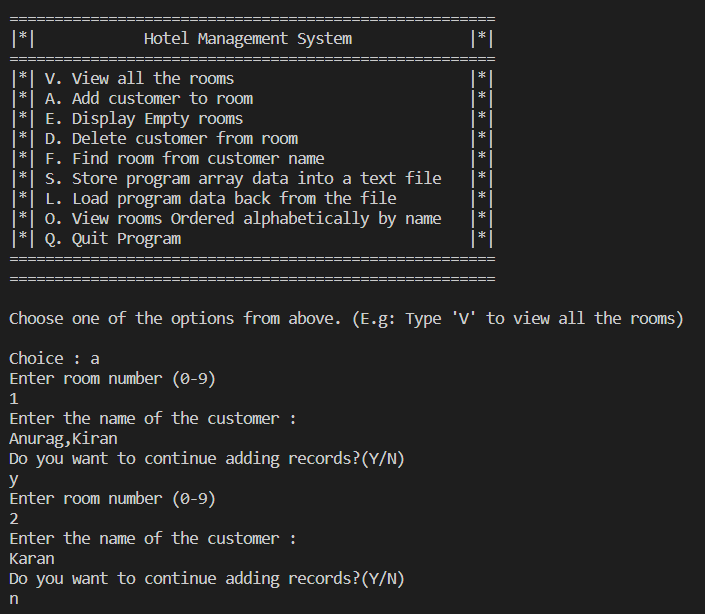
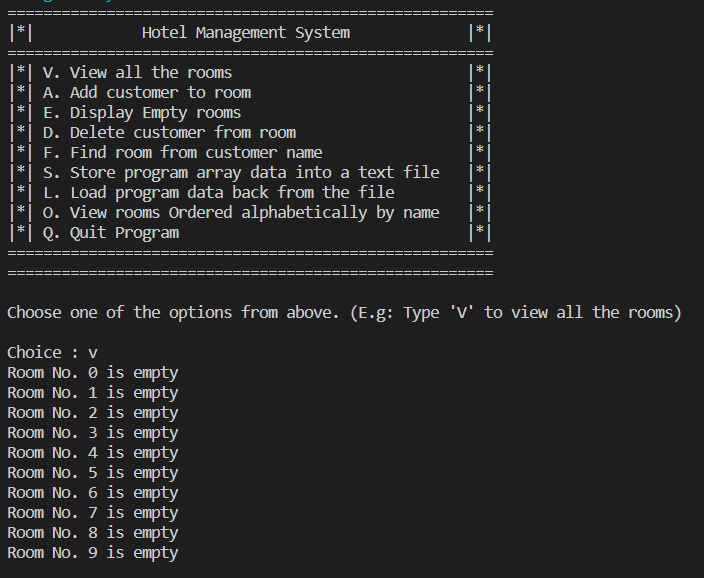
}

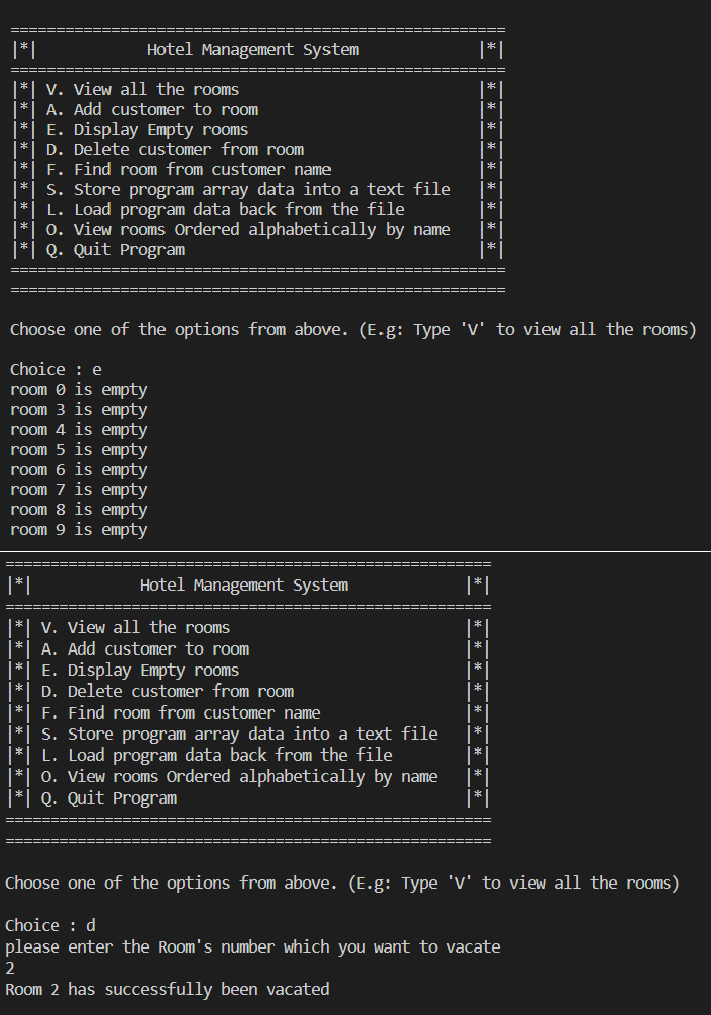
//Displays this message if the file is found and successfully loaded the data back from it System.out.println("File successfully loaded");

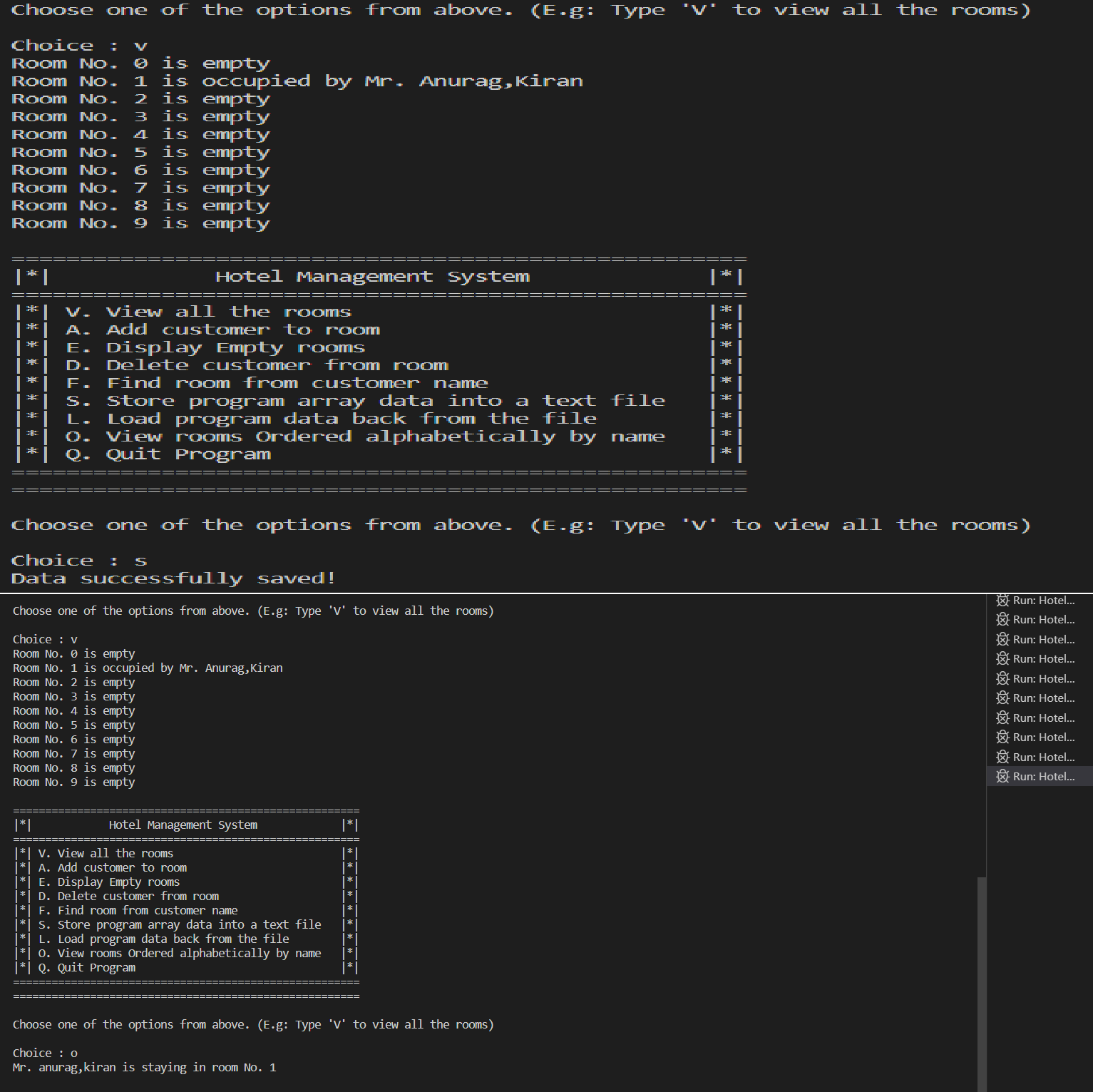
System.out.println(""); menu();

}

* **Output of the Program:-**







1. **CONCLUSION:-**

In conclusion, the development of a hotel management system in Java is an active research area. The reviewed studies presented different designs, implementations, and evaluation methods of hotel management systems. These studies provide valuable insights into the design and implementation of a hotel management system using Java, which can be used as a reference for future research in this area.

## REFERENCES:-

<https://chat.openai.com/chat> <https://www.geeksorgeeks.org/> <https://www.javatpoint.com/>

## RESOURCE REQUIRED:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. no** | **Instrument/Object** | **Specifications** | **Quantity** | **Remark** |
| 01 | Computer system | Computer(5-Ipreferabl e), RAM minimum 8GB | 1. | . |
| 02 | Software | VS CODE | 1. |  |
| 03 | Any other resources  use | MS WORD | 1. |  |

#### SKILLS DEVELOPMENT:-

1. Communication skill
2. Leadership skill
3. Team work skill
4. Scientific approach development
5. Data collection skill
6. Research skill

#### Prof.Kadlag .S.U Prof.Chaudhari N.K (Project Guide) (H.O.D) IT