VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590 018

A Mini -Project Work on

Hotel Management System

A Dissertation work submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Engineering

In

Information Science & Engineering

Submitted by

Chethan Holla BV	**1AY18IS030**
Jeevan V	**1AY18IS048**

Under the guidance of

Prof. Ramesh Sengodan

Assistant Professor

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ACHARYA INSTITUTE OF TECHNOLOGY

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI.APPROVED BY AICTE, NEW DELHI, ACCREDITED BY NAAC, NEW DELHI)

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107

2020-21

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ACHARYA INSTITUTE OF TECHNOLOGY

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI.APPROVED BY AICTE, NEW DELHI, ACCREDITED BY NAAC, NEW DELHI)

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107

ACHARYA

Certificate

This is to Certify that the Mini-Project work entitled **“Hotel Management System ”** is a bonafide work carried out by **Chethan Holla BV (1AY18IS030) and Jeevan V (1AY18IS030** in partial fulfillment for the award of the degree of **Bachelor of Engineering** in **Information Science and Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Bachelor of Engineering Degree.

Prof. Ramesh Sengodan	**Prof. Marigowda C K**
Guide	HOD

Name of the Examiners	**Signature with date**
1. ____________________	____________________
2. ____________________	____________________

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this mini-project would be incomplete without the mention of the people who made it possible through constant guidance and encouragement.

We would take this opportunity to express our heart-felt gratitude to **Sri. B. Premnath Reddy**, Chairman, Acharya Institutes and **Dr. Prakash M R**, Principal, Acharya Institute of Technology for providing the necessary infrastructure to complete this mini-project.

We wish to express our deepest gratitude and thanks to **Prof. Marigowda C K**, Head of the Department, Information Science and Engineering and the project coordinator **Prof Pakruddin B** and **Prof Raushan Kashypa** for their constant support.

We wish to express sincere thanks to our guide **Prof. Ramesh Sengodan**, Assistant Professor, Department of Information Science and Engineering for helping us throughout and guiding us from time to time.

A warm thanks to all the faculty of Department of Information Science and Engineering, who have helped us with their views and encouraging ideas.

Chethan Holla BV (1AY18IS030)
Jeevan V (1AY18IS048)

ABSTRACT

This project is all about Hotel application to makes ease the work of users by this computerized software. By this application a user can store customer details, modify customer details, maintain records, room details, and personal data, generate bills for customers on single platform. Thus, the user can manage his contacts and daily working schedules through this application. This application avoids user to make manual contact diary entries to store the details of the customer and rooms.

A user who is working on system can input customer details for the allotted room while checking-out other customers. Employees are able to generate bills according to their suitability. So, this application is convenient platform for a user to manage, search, modify and to reduce manual work of the user.

As manual computing system becomes more numerous, complex the need for the systematic approaches development becomes increasingly apparent. The objective of this project is designing a convenient framework including room details, customer details and bill generation to the user on one platform. A primary goal of this project is to develop good software to overcome the existing problem caused by manual systems.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction to File Structures

File Structures is the Organization of Data in Secondary Storage Device in such a way that minimize the access time and the storage space. A File Structure is a combination of representations for data in files and of operations for accessing the data. A File Structure allows applications to read, write and modify data. It might also support finding the data that matches some search criteria or reading through the data in some particular order.

"File organization" refers to the logical relationship among the various records that constitute the file, particular with respect to means of identification access to any specific record. "File structure" refers to the format of label and data blocks and of any logical record control, information. The organization of the given file maybe sequential relative, or indexed.

Co-Sequential Processing

Co-sequential Processing is the coordinated processing of two or more sequential lists to produce single output list. Sometimes the processing results in a merging, or union, sometimes it results in a matching, or intersection and other times the operation is a combination of matching and merging of the items in the lists.

Aim of the File Structure

The need for file structures

- Ideally, we would like to get the information we need once access to the disk
- If it is impossible to get what we need in one access, we want structures that allow us to find the target information with as few accesses as possible.
- We want our file structures to group information so we are likely to get thing

  We need with only trip to the disk.

1.2 Fundamental Operations on File

Open: A function or system call that makes a file ready for use. It may also bind a logical filename to a physical file. Its arguments include the logical filename, the physical filename and may also include information on how the file is to be accessed.

Close: A function or system call that breaks the link between the logical filename and the corresponding physical filename.

Create: A function or system call that causes a file to be created on secondary storage and may also binds a logical filename to the file's physical filename.

Read: A function or a system call used to obtain input from a file or device. When viewed at the lowest level, it requires three arguments: a source file logical name corresponding to an open file; the destination address for the address and the size or amount of data to be read.

Write: A function or system call used to provide output capabilities. When viewed at the lowest level, it requires three arguments: a destination filename corresponding to an open file; the source address of the bytes that are to be written and the size or amount of data to be written.

Seek: A function or a system call that sets the read/write pointer to a specified position in a file. Languages that provide seeking functions allow programs to access specific.

1.3 Services provided to the User

The functions that the Source Code Crime Management System provides are as follows:

1. INSERT: This operation is performed when new data needs to be added to the system, for e.g. When user lodges a new complaint /criminal, the record is inserted into the files.

2. DELETE: This operation clears the existing records in the various files. It is used when for e.g. User can delete the record of complaint/criminal, that particular record is deleted from the file.

3. SEARCH: This function is used to search particular record from the System, for e.g.This function can search for a particular record using complaint/criminal type as a unique id.

5. DISPLAY: It displays each and every records of complaint/criminal data, for e.g When display all function is given it displays all the records present in the file.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 Hardware Requirements

- **Processor:** Intel Core2 Quad @ 2.4Ghz on Windows® Vista 64-Bit / Windows® 7 64-Bit / Windows® 8 64-Bit / Windows® 8.1 64-Bit.
- **RAM:** 2GB of RAM
- **Memory:** 256GB Hard drive
- **Keyboard:** MS compatible keyboard
- **Mouse:** MS compatible mouse

2.2 Software Requirements

- Documentation Tool: MS Word
- Development Tool: Visual Studio Code

CHAPTER 3

SYSTEM OVERVIEW

3.1 Problem Definition

In the existing system people who want to book a room or order a meal must wait a long time to get the work done which is time consuming. Hotel staff usually maintain records manually which is again time consuming and it is difficult to manage those records.There can be loss of customer records and other details which we need to overcome.

3.1.1 Solution on problem

This project is all about Hotel management application to make ease the work of users by this computerized software. This application manages Room activities, Admission of a New Customer, Assign a room according to customer’s demand, ordering of meals, checkout of a customer and releasing the room and finally compute the bill.

This software addresses these problems by making it online and computerized management of these bookings along with maintaining records of customers. This saves a lot of time of both staff and customer..

3.2 The Sequential Search

When data items are stored in a collection such as a list, we say that they have a linear or sequential relationship. Each data item is stored in a position relative to the others. In Python lists, these relative positions are the index values of the individual items. Since these index values are ordered, it is possible for us to visit them in sequence. This process gives rise to our first searching technique, the sequential search.

Figure 3.1 shows how this search works. Starting at the first item in the list, we simply move from item to item, following the underlying sequential ordering until we either find what we are looking for or run out of items. If we run out of items, we have discovered that the item we were searching for was not present.

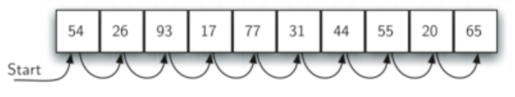


Fig 3.1: The Sequential Search

3.2.1 Analysis of Sequential Search

To analyse searching algorithms, we need to decide on a basic unit ofcomputation.

Recall that this is typically the common step that must be repeated in order to solve the problem. For searching, it makes sense to count the number of comparisons performed. Each comparison may or may not discover the item we are looking for. In addition, we make another assumption here. The list of items is not ordered in any way. The items have been placed randomly into the list. In other words, the probability that the item we are looking for is in any particular position is exactly the same for each position of the list.

If the item is not in the list, the only way to know it is to compare it against every item present. If there are n items, then the sequential search requires nn comparisons to discover that the item is not there. In the case where the item is in the list, the analysis is not so straightforward. There are actually three different scenarios that can occur. In the best case we will find the item in the first place we look, at the beginning of the list. We will need only one comparison. In the worst case, we will not discover the item until the very last comparison, the *nth* comparison.

What about the average case? On average, we will find the item about halfway into the list; that is, we will compare against n2n2 items. Recall, however, that as *n* gets large, the coefficients, no matter what they are, become insignificant in our approximation, so the complexity of the sequential search, is O(n)O(n). Table below summarizes these results.

Case	**Best Case**	**Worst Case**	**Average Case**
item is present	1	n	n/2
item is not present	n	n	n

Assume that the list of items was constructed so that the items were in ascending order, from low to high. If the item we are looking for is present in the list, the chance of it being in any one of the *n* positions is still the same as before. We will still have the same number of comparisons to find the item. However, if the item is not present there is a slight advantage. Figure3.2 shows this process as the algorithm looks for the item 50. Notice that items are still compared in sequence until 54. At this point, however, we know something extra. Not only is 54 not the item we are looking for, but no other elements beyond 54 can work either since the list is sorted. In this case, the algorithm does not have to continue looking through all of the items to report that the item was not found. It can stop immediately.

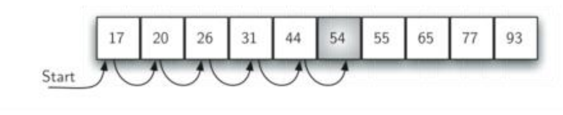


Fig 3.2: Sequential Search of an Ordered List of Integers

CHAPTER 4

DESIGN

4.1 DATA FLOW DESIGN

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

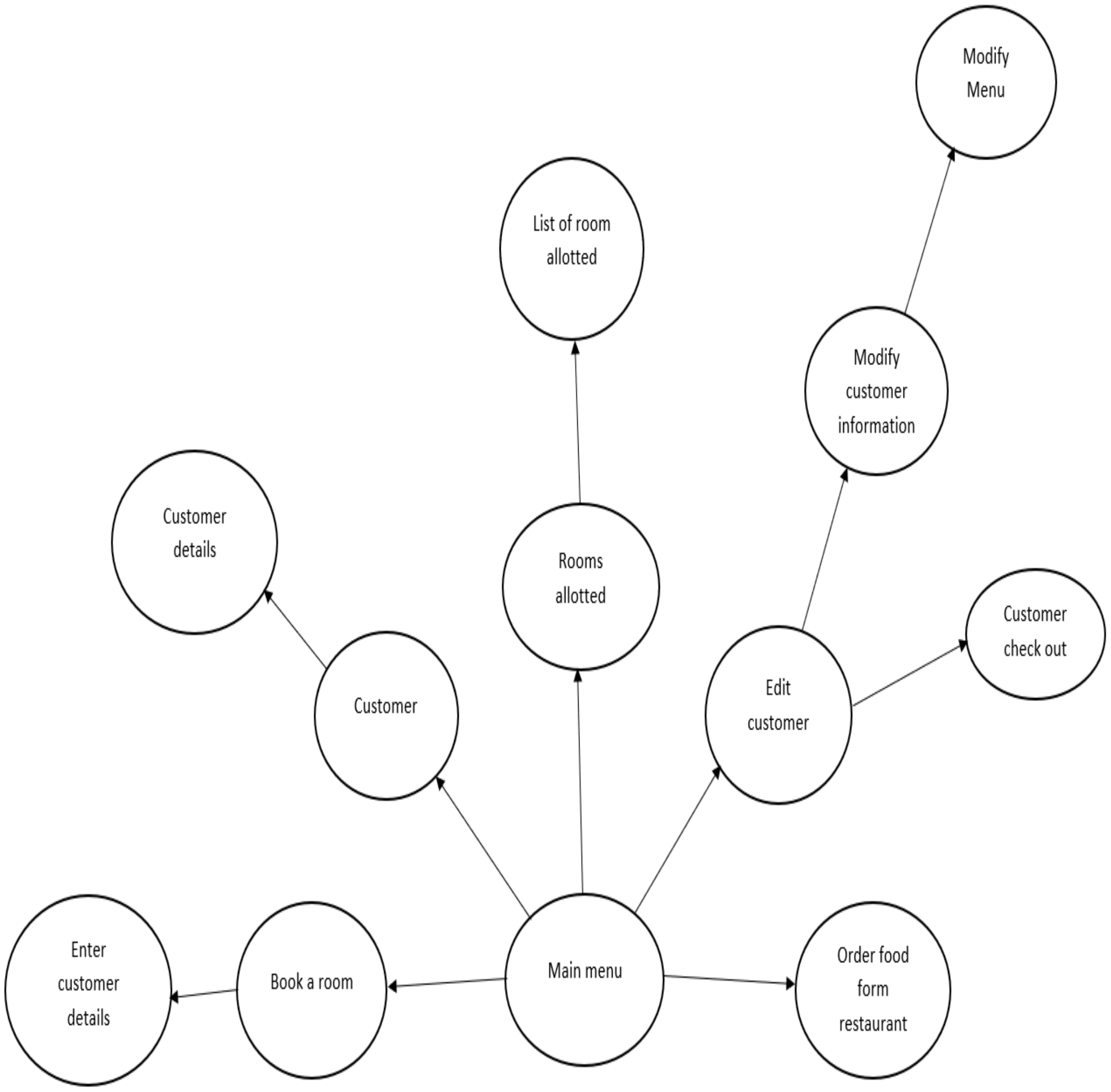


Fig 4.1: Data Flow Design

4.2 CLASS DIAGRAM

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

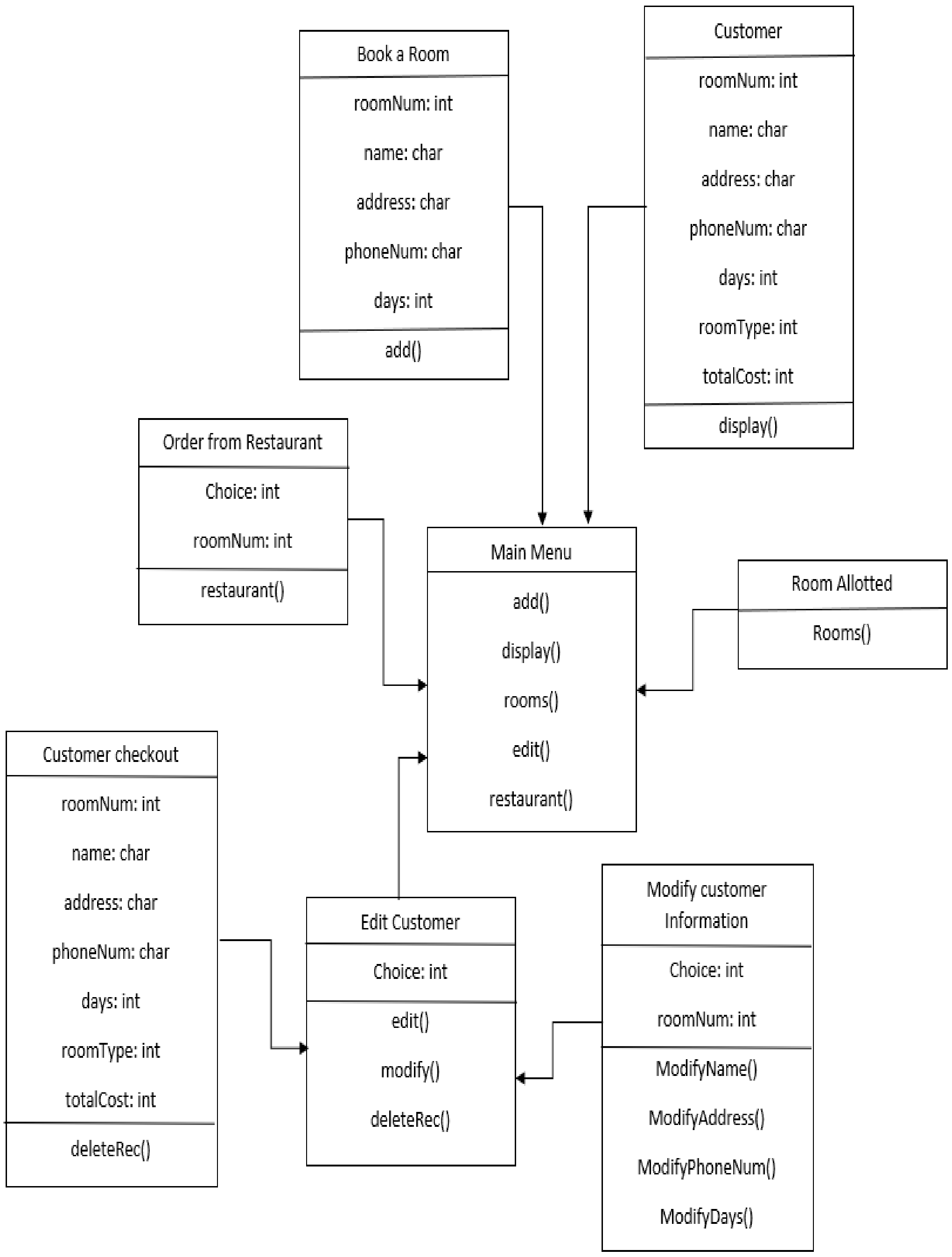


Fig 4.2: Class Diagram

CHAPTER 5

IMPLEMENTATION

Introduction:

The application is developed into seven parts, Part1 to 7. We have used the object-oriented concept in the implementation of design. The object-oriented programming language used was C++. The concept of inheritance is used to a large extent.

Functions and the members involved:

Part 1:

We design the class called Book_a_Room. Each object of class represent Booking a Room.

Members in Book_a_Room class includes roomNum, name, address, phoneNum, days.

Methods in Book_a_Room class includes add().

Part 2:

We design the class Customer. Each object of class represent Customer Information.

Members in Customer class includes roomNum, name, address, phoneNum, days, roomtype, cost.

Methods in Customer class includes display().

Part 3:

We design the class Room Allotted. Each object of class displays the Information of all Customer.

Methods in Room Allotted class includes rooms().

Part 4:

We design the class Edit Customer. Each object of class is used to edit the details of customer.

Members in Edit Customer class includes Choice.

Methods in Edit Customer class includes edit(), modify(), deleteRec().

Part 5:

We design the class Order from Restaurant. Each object of class represent ordering of Food.

Members in Order from Restaurant class includes Choice, roomNum.

Methods in Order from Restaurant class includes restaurant().

Part 6:

We design the class Modify customer Information. Each object of class is used to edit the details of customer.

Members in Modify customer Information class includes Choice, roomType.

Methods in Modify customer Information class includes ModifyName(), ModifyAddress(), ModifyPhoneNum(), ModifyDays()

Part 7:

We design the class Customer checkout. Each object of class is used to edit the details of customer.

Members in Customer checkout class includes roomNum, name, address, phoneNum, days, roomtype, cost.

Methods in Customer checkout class includes deleteRec().

CHAPTER 6
TESTING

The integral part of any system’s development life cycle is testing without which the system developed is sure to fail and result in loss of economic and manpower investments besides user’s dissatisfaction and downfall of reputation. System testing is the stage of implementation, which aims at ensuring that the system works accurately and efficiently before actual operation commences. No program or system design is perfect, communication between the user and the designer is not always complete or clear. All this can result in errors.

Another reason for system testing is its utility as a user oriented vehicle before implementation. The application system is worthless if does not meet user needs, thus the system should be tested to see whether it meets the user requirements.

Testing here is conducted in bottom up approach as follows:

- Module testing: Here testing is done at each module level. Each case has been thoroughly tested to discover pitfalls.
- System testing: Here testing is done after all the modules have been integrated.

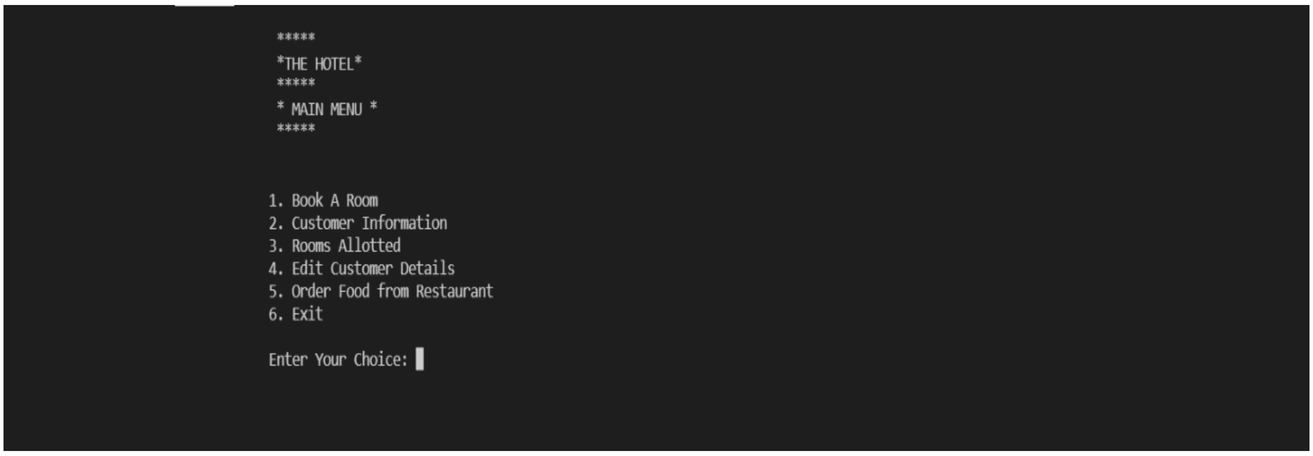
TEST CASES:

SL.NO	MODULE	TEST CASE DESCRIPTION	INPUT	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	Main Menu	Accept the suitable options given in the menu	1.Book a room 2. Customer 3. Rooms allotted 4. Edit customer 5. Order food form restaurant	Suitable options are displayed on the console, choose a specific option mentioned in menu	Suitable options are displayed on the console, choose a specific option mentioned in menu	Pass
2	Book a room	Provide input to the fields specified to book a room	roomNum, name, address, phoneNum, days	Adds the record and displays a message	Adds the record and displays a message	Pass

3	Customer	provide the type of complaint to be view the customer	roomNum, name, address, phoneNum, days, roomtype, cost.	Displays all the detail of the specific customer	Displays all the detail of the specific customer	Pass
4	Rooms allotted	Lists all the customers Information	No input given	Displays the all the customers records	Displays the all the customers records	Pass
5	Edit customer	Provide input to the fields specified to edit the customers details	Choice	Displays two options to Modify and Delete	Displays two options to Modify and Delete	Pass
6	Modify customer Information	Provide input to the fields specified to modify existing customer details	Choice, roomType, roomNum, name, address, phoneNum, days, roomtype, cost	Records are modified Successfully	Records are modified Successfully	Pass
7	Customer checkout	Provide input to the fields specified to remove the customer details	roomNum, name, address, phoneNum, days, roomtype, cost.	Records are deleted successfully,	Records are deleted successfully	Pass
8	Order from Restaurant	Lists all the existing criminal	Choice, roomNum	Displays the Food menu and cost of the food	Displays the Food menu and cost of the food	Pass
9	Exit	Allows to exit from the main menu	Choose exit options	Exit from the console.	Exit from the console.	Pass

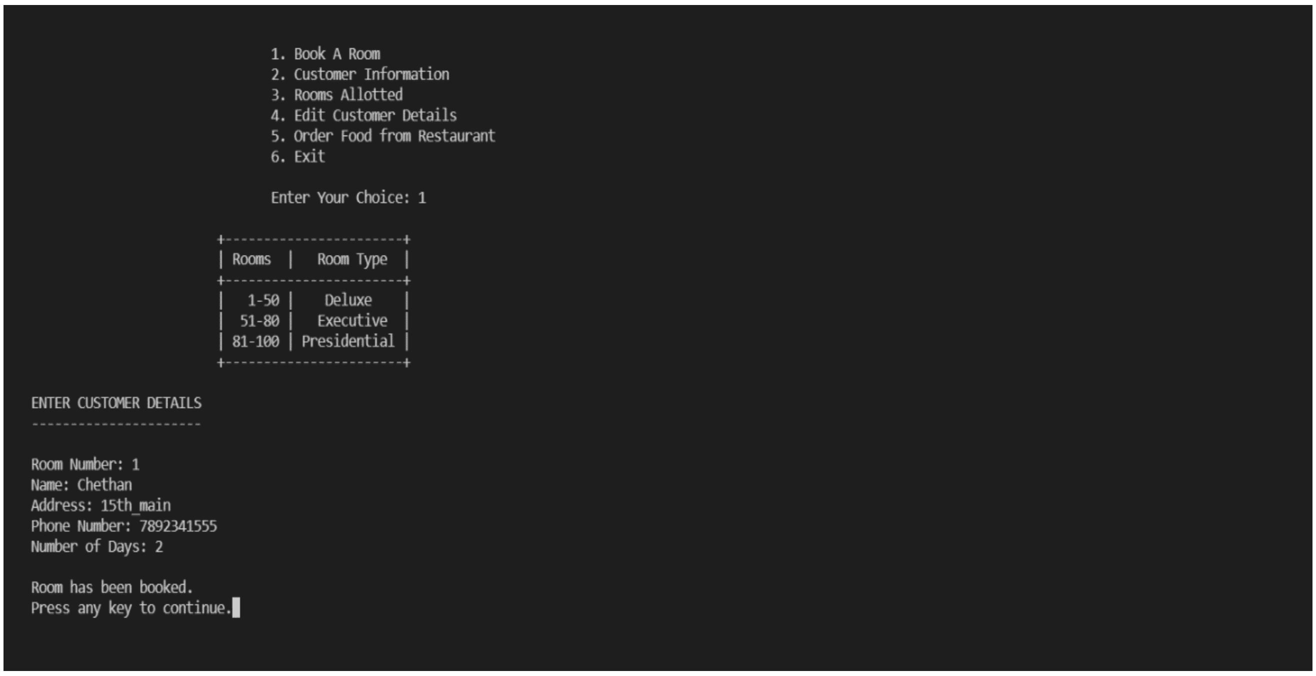
CHAPTER 7

SNAPSHOTS



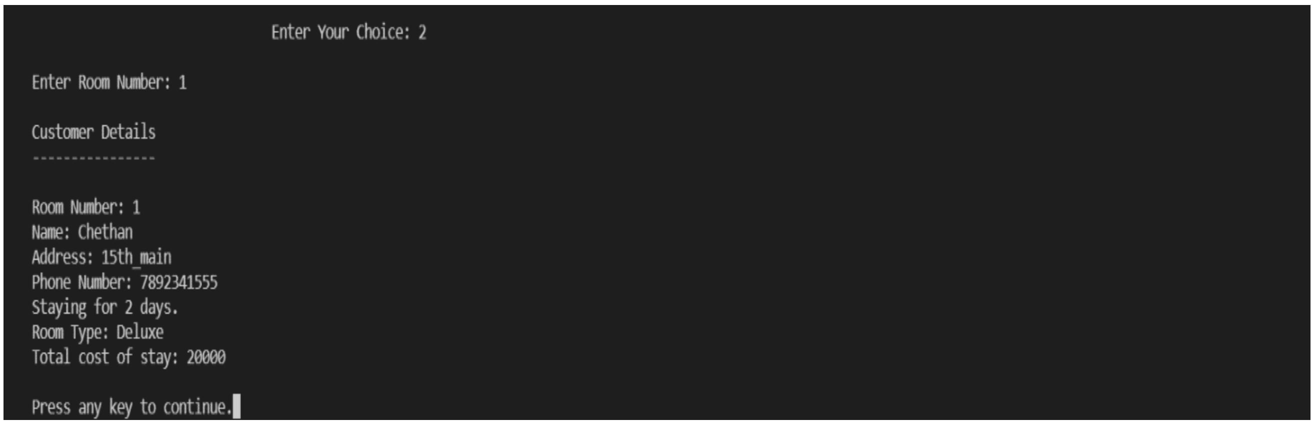
```
*****
*THE HOTEL*
*****
* MAIN MENU *
*****


1. Book A Room
2. Customer Information
3. Rooms Allotted
4. Edit Customer Details
5. Order Food from Restaurant
6. Exit

Enter Your Choice: █
```

Fig 7.1: Main Menu



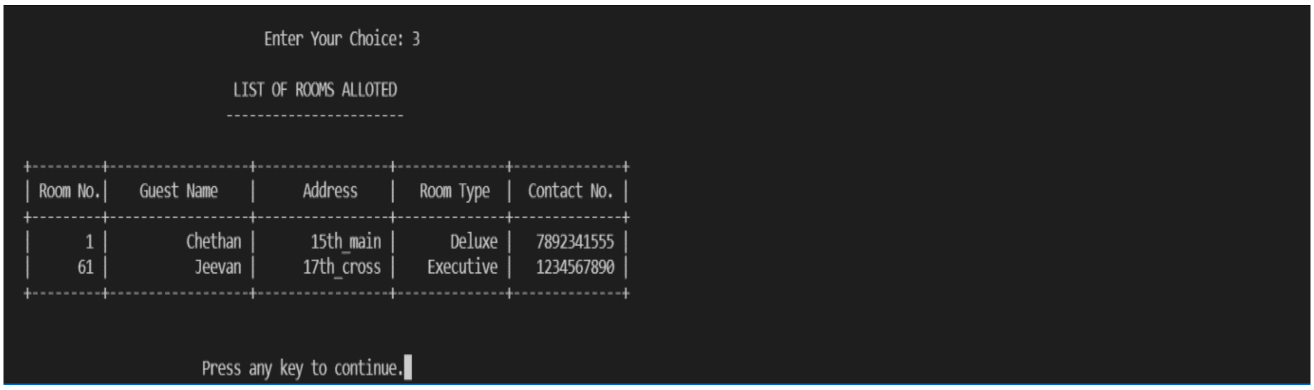
```
                    1. Book A Room
                    2. Customer Information
                    3. Rooms Allotted
                    4. Edit Customer Details
                    5. Order Food from Restaurant
                    6. Exit

                    Enter Your Choice: 1

               +----------------------+
               | Rooms  |   Room Type |
               +----------------------+
               |   1-50 |    Deluxe   |
               |  51-80 |   Executive |
               | 81-100 | Presidential|
               +----------------------+

ENTER CUSTOMER DETAILS
----------------------

Room Number: 1
Name: Chethan
Address: 15th_main
Phone Number: 7892341555
Number of Days: 2

Room has been booked.
Press any key to continue.█
```

Fig 7.2: Book A Room



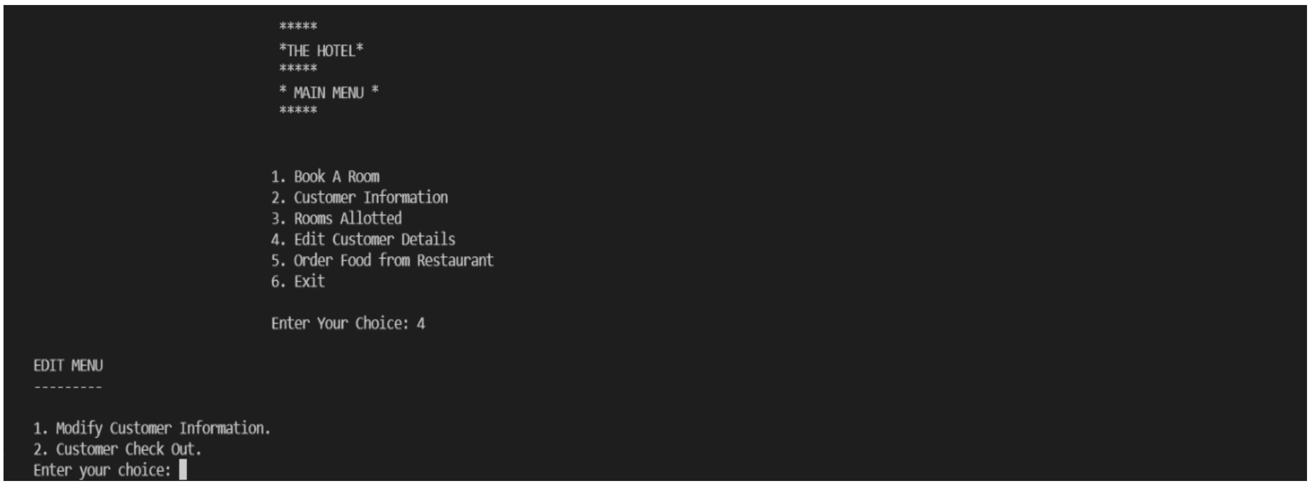
```
                              Enter Your Choice: 2

   Enter Room Number: 1

   Customer Details
   ----------------

   Room Number: 1
   Name: Chethan
   Address: 15th_main
   Phone Number: 7892341555
   Staying for 2 days.
   Room Type: Deluxe
   Total cost of stay: 20000

   Press any key to continue.
```

Fig 7.3: Customer Information



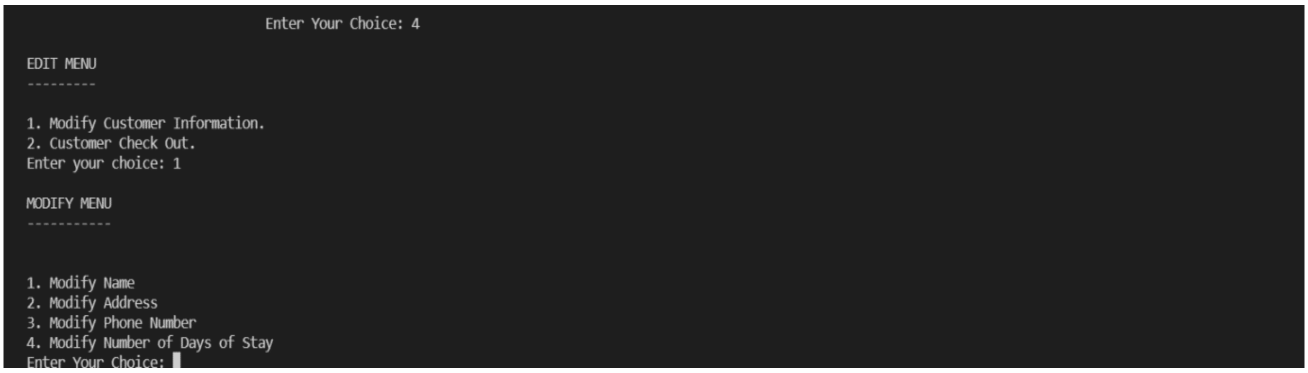
```
                              Enter Your Choice: 3

                          LIST OF ROOMS ALLOTED
                         ----------------------

  +--------+----------------+---------------+------------+------------+
  | Room No.|   Guest Name   |    Address    | Room Type  | Contact No. |
  +--------+----------------+---------------+------------+------------+
  |      1 |        Chethan |     15th_main |     Deluxe | 7892341555 |
  |     61 |         Jeevan |    17th_cross |  Executive | 1234567890 |
  +--------+----------------+---------------+------------+------------+


                     Press any key to continue.
```

Fig 7.4: Room Allotted



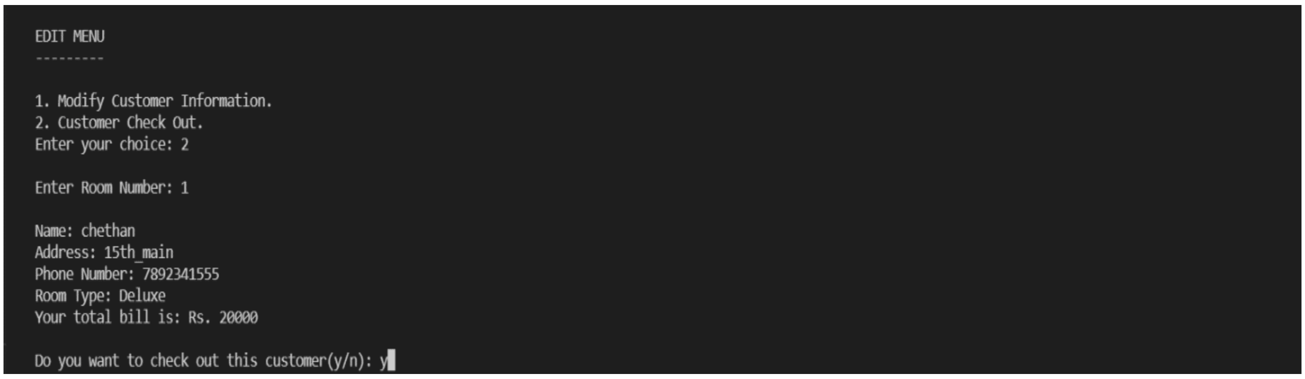
```
                                *****
                                *THE HOTEL*
                                *****
                                * MAIN MENU *
                                *****


                               1. Book A Room
                               2. Customer Information
                               3. Rooms Allotted
                               4. Edit Customer Details
                               5. Order Food from Restaurant
                               6. Exit

                               Enter Your Choice: 4

   EDIT MENU
   ---------

   1. Modify Customer Information.
   2. Customer Check Out.
   Enter your choice:
```

Fig 7.5: Edit Customer Details



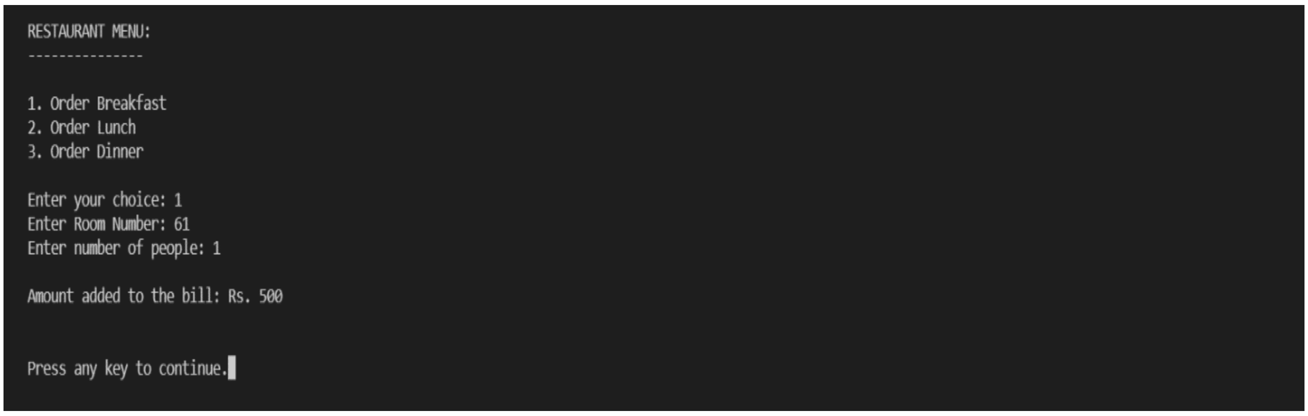
```
                              Enter Your Choice: 4

EDIT MENU
---------

1. Modify Customer Information.
2. Customer Check Out.
Enter your choice: 1

MODIFY MENU
-----------


1. Modify Name
2. Modify Address
3. Modify Phone Number
4. Modify Number of Days of Stay
Enter Your Choice:
```

Fig 7.6: Modify Customer Information



```
EDIT MENU
---------

1. Modify Customer Information.
2. Customer Check Out.
Enter your choice: 2

Enter Room Number: 1

Name: chethan
Address: 15th_main
Phone Number: 7892341555
Room Type: Deluxe
Your total bill is: Rs. 20000

Do you want to check out this customer(y/n): y
```

Fig 7.7: Customer Check Out



```
RESTAURANT MENU:
---------------

1. Order Breakfast
2. Order Lunch
3. Order Dinner

Enter your choice: 1
Enter Room Number: 61
Enter number of people: 1

Amount added to the bill: Rs. 500


Press any key to continue.
```

Fig 7.8: Order Food From Restaurant

CONCLUSION & FUTURE ENHANCEMENT

Conclusion

In conclusion, it is obvious that the system designed for the Hotel management system is a reliable system where you can perform operations that include inserting deleting searching and displaying. Here, one sorted files id maintained. We use the concept Sequential Search to retrieve the data that is stored.

Future Enhancement

We are going to make Online Web based Application for this system so customers can automatically find rooms and reserve thre room in advance.

BIBLIOGRAPHY

Book References:

- ✓ Michael J Folk –FILE STRUCTRES
- ✓ Grady Booch. – Object oriented Design with Applications
- ✓ Larson P – Performance analysis of linear hashing with extensions
- ✓ Peterson, W.W- "Addressing for Random Access Storage". IBM journal of Research.

Web References:

[1] https://runestone.academy

[2] www.alphaworks.ibm.com

[3] www.research.ibm.com/filestructures