**ABSTRACT**

The Project HOTEL MANAGEMENT SYSTEM is a standalone application that allows the hotel manager to handle all hotel activities online. Interactive GUI and the ability to manage various hotel bookings and rooms make this system very flexible and convenient. The hotel manager is a very busy person and does not have the time to sit and manage the entire activities manually on paper. This application gives him the power and flexibility to manage the entire system from a single online system.

Hotel Management project provides room booking, staff management and other necessary hotel management features. The system allows the manager to post available rooms in the system. Customers can view and book room online. Admin has the power of either approving or disapproving the customer's booking request. Other hotel services can also be viewed by the customers and can book them too. The system is hence useful for both customers and managers to portably manage the hotel activities.

**CHAPTER 1**

**INTRODUCTION**

The Hotel Management System is a tool for booking the rooms of Hotel through online by the Customer. It provides the proper management tools and easy access to the customer information. This Hotel Management System Software Requirement Specification (SRS) main objective is to provide a base for the foundation of the project. It gives a comprehensive view of how the system is supposed to work and what is to be expected by the end users. Client’s expectation and requirements are analysed to produce specific unambiguous functional and non-functional requirements, so they can be used by development team with clear understanding to build a system as per end user needs.

Hotel Management System is a system that provides us to reserving rooms, checking whether the rooms are vacant are or not etc by using online browsing. This system is very useful to all especially for business people. For Business people they don’t have sufficient time for these then they can use these type of online Hotel Management Systems. By this project we will reduce the faults in bills of their expenditure and decrease time of delay to give the bills to the customers. We can also save the bills of the customer. By this project we can also include all the taxes on the bills according to their expenditures. It has a scope to reduce the errors in making the bills. Computerized bill can be printed within fraction of seconds. Online ordering of Booking is possible by using this software.

This Project is based on C++ and File structure. If anyone wants to book the room for few days then they can specify the specific number by seeing the types of rooms we have. The bill of this online booking is based on the type of room they can select is displayed. HOTEL MANAGEMENT SYSTEM is a hotel reservation site script where site users will be able to search rooms availability with an online booking reservations system. Site users can also browse hotels, view room inventory, check availability, and book reservations in real-time. Site users enter check in date and check out date then search for availability and rates. After choosing the right room in the wanted hotel – all booking and reservation process is done on the standalone application.**CHAPTER 2**

**AIM, OBJECTIVE AND SCOPE OF THE PROJECT**

**2.1 Aim of the project**

The purpose of hotel booking system is to automate the existing manual system by the help of computerized equipment’s and full-fledged computer software, fulfilling their requirement, so that their valuable or information can be stored for a longer period with easy accessing and manipulating of the same . The required software and hardware are easily available and easy to work with.

**2.2 Objective of the project**

The objectives of the automated Hotel Management System are to simplify the day-to-day processes of the hotel. The system will be able to handle many services to take care of all customers in a quick manner. As a solution to the large amount of file handling happening at the hotel, this software will be used to overcome those drawbacks. Safety, easiness of using and most importantly the efficiency of information retrieval is some benefits the development team going to present with this system. The system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction

**2.3 Scope of this project**

The hotel management system mini project is intended for the reservations for room that can be made through online. It will be able to automate the various operations of the Hotel. Our Hotel Management System will have three end users: Customer, Receptionist, and Hotel Manager. Customers will be able to check for room’s availability, select the rooms, and pay for the room. Receptionist will have access to update or modify booking details. Manager will able to view the financial report and able to update room information such as cost and category. The main goal of this introduced automated HMS software is to simplify every day process of hotel. Day to day Hotels are increasing and they need to automate to provide customer ease of access. It will be able to take care of services to customer in a quick manner. This automation will be able to replace the drawbacks of large customer information physical files which were difficult to handle. Secure Transaction, quick retrieval of information, ease of use, quick recovery of errors, fault tolerance are some of the benefits that development team will be working on to achieve end user satisfaction.

This proposes that efficiency of hotel organizations could be improved by integrating service-oriented operations service-oriented operations with project management principles. Such integration would instill innovation, proactive attitudes and regulated risk-taking needed to pursue ongoing improvement and proactive response to change. By managing each change as a project, embedded in smoothly running operations, hotels would extend their life span by continuously reinventing themselves.

**CHAPTER 3**

**REQIREMENT SPECIFICATION**

**3.1 System Requirements**

The basic requirements for the development of this mini project are as follows

**3.1.1 Hardware Configuration**

1. Processor: Intel core i3 or above
2. Ram:512 MB
3. Hard disk:20 GB

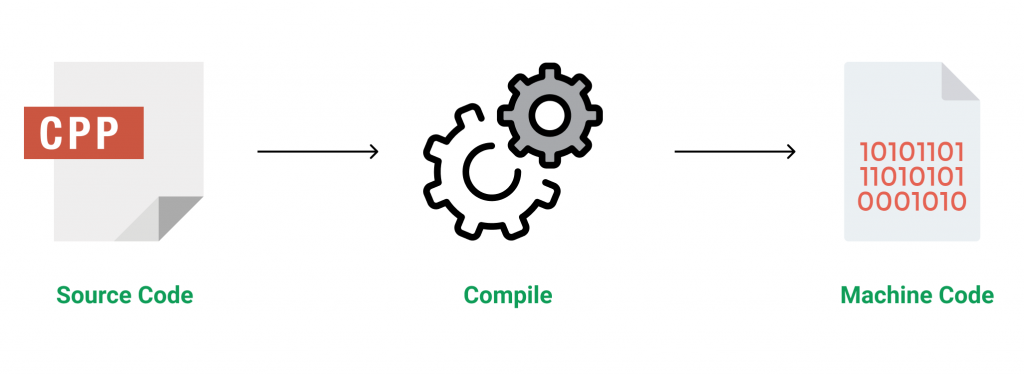
**3.1.2 Software Configuration**

1. Language: C & C++
2. IDE: DEV C++
3. Documentation tool: Microsoft office 2003 or above

**3.2 Development Environment**

**3.2.1 C++**

**C++** is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a **compiled** language[7].



**Figure 3.1 Source code compilation**

C++ is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.). The basic syntax and code structure of both C and C++ are the same.

Some of the features & key-points to note about the programming language are as follows:

* Simple: It is a simple language in the sense that programs can be broken down into logical units and parts, has a rich library support and a variety of data-types.
* Machine Independent but Platform Dependent: A C++ executable is not platform-independent (compiled programs on Linux won’t run on Windows), however they are machine independent.
* Mid-level language: It is a mid-level language as we can do both systems-programming (drivers, kernels, networking etc.) and build large-scale user applications (Media Players, Photoshop, Game Engines etc.)
* Rich library support: Has a rich library support (Both standard ~ built-in data structures, algorithms etc.) as well 3rd party libraries (e.g. Boost libraries) for fast and rapid development.
* Speed of execution: C++ programs excel in execution speed. Since, it is a compiled language, and also hugely procedural. Newer languages have extra in-built default features such as garbage-collection, dynamic typing etc. which slow the execution of the program overall. Since there is no additional processing overhead like this in C++, it is blazing fast.
* Pointer and direct Memory-Access: C++ provides pointer support which aids users to directly manipulate storage address. This helps in doing low-level programming (where one might need to have explicit control on the storage of variables).
* Object-Oriented: One of the strongest points of the language which sets it apart from C. Object-Oriented support helps C++ to make maintainable and extensible programs. i.e. Large-scale applications can be built. Procedural code becomes difficult to maintain as code-size grows.
* Compiled Language: C++ is a compiled language, contributing to its speed.

**3.2.2 File Structure**

In computing, file system or filesystem (often abbreviated to fs) is a method and data structure that the operating system uses to control how data is stored and retrieved.[1] Without a file system, data placed in a storage medium would be one large body of data with no way to tell where one piece of data stopped and the next began, or where any piece of data was located when it was time to retrieve it [5]. By separating the data into pieces and giving each piece a name, the data is easily isolated and identified. Taking its name from the way a paper-based data management system is named, each group of data is called a "file”. The structure and logic rules used to manage the groups of data and their names is called a "file system”.

A file system consists of two or three layers. Sometimes the layers are explicitly separated, and sometimes the functions are combined. The logical file system is responsible for interaction with the user application. It provides the application program interface (API) for file operations — OPEN, CLOSE, READ, etc., and passes the requested operation to the layer below it for processing. The logical file system "manage[s] open file table entries and per-process file descriptors".[5] This layer provides "file access, directory operations, [and] security and protection".

The second optional layer is the virtual file system. "This interface allows support for multiple concurrent instances of physical file systems, each of which is called a file system implementation". The third layer is the physical file system. This layer is concerned with the physical operation of the storage device (e.g. disk). It processes physical blocks being read or written. It handles buffering and memory management and is responsible for the physical placement of blocks in specific locations on the storage medium. The physical file system interacts with the device drivers or with the channel to drive the storage device.

* + 1. **Files Operation**

A file is an abstract data type. To define a file properly, we need to consider the operations that can be performed on files. Six basic file operations. The OS can provide system calls to create, write, read, reposition, delete, and truncate files.

* Creating a file: Two steps are necessary to create a file.
  + Space in the file system must be found for the file.
  + An entry for the new file must be made in the directory
* Writing a file: To write a file, we make a system call specifying both the name of the file and the information to be written to the file. The system must keep a write pointer to the location in the file where the next write is to take place. The write pointer must be updated whenever a write occurs.
* Reading a file: To read from a file, we use a system call that specifies the name of the file and where (in memory) the next block of the file should be put. The system needs to keep a read pointer to the location in the file where the next read is to take place.
* Because a process is usually either reading from or writing to a file, the current operation location can be kept as a per-process current-file-position pointer.
* Both the read and write operations use this same pointer, saving space and reducing system complexity.
* Repositioning within a file: The directory is searched for the appropriate entry, and the current-file-position pointer is repositioned to a given value. Repositioning within a file need not involve any actual I/O. This file operation is also known as a file seek.
* Deleting a file: To delete a file, we search the directory for the named file. Having found the associated directory entry, we release all file space, so that it can be reused by other files, and erase the directory entry.
* Truncating a file: The user may want to erase the contents of a file but keep its attributes. Rather than forcing the user to delete the file and then recreate it, this function allows all attributes to remain unchanged (except for file length) but lets the file be reset to length zero and its file space released.

These six basic operations comprise the minimal set of required file operations. These primitive operations can then be combined to perform other file operations (i.e., copying). The OS keeps a small table, called the open-file table, containing information about all open files. When a file operation is requested, the file is specified via an index into this table, so no searching is required. When the file is no longer being actively used, it is closed by the process, and the OS removes its entry from the open-file table. Most systems require that the programmer open a file explicitly with the $open () system call before that file can be used. The $open () operation takes a file name and searches the directory, copying the directory entry into the open-file table.

This call can also accept access-mode information (create, read-only, read-write, append-only, and so on). This mode is checked against the file's permissions. If the request mode is allowed, the file is opened for the process. The $open () system call typically returns a pointer to the entry in the open-file table. This pointer, not the actual file name, is used in all I/O operations. The implementation of the $open () and $close () operations is more complicated in an environment where several processes may open the file at the same time. This may occur in a system where several different applications open the same file at the same time.

Typically, the OS uses two levels of internal tables:

* A per-process table. The per-process table tracks all files that a process has open. For instance, the current file pointer for each file is found here. Access rights to the file and accounting information can also be included.
* A system-wide table. Each entry in the per-process table in turn points to a system-wide open-file table. The system-wide table contains process-independent information, such as the location of the file on disk, access dates, and file size. Once a file has been opened by one process, the system-wide table includes an entry for the file.

Typically, the open-file table also has an open count associated with each file to indicate how many processes have e the file open. Each $close () decreases this open count, and when the open count reaches zero, the file is no longer in use, and the file's entry is removed from the open-file table. In summary, several pieces of information are associated with an open file.

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

Hotel management system is a large project developed using graphics in the C++ language. Here, users can perform general hotel management operations related to food menu, hotel bill, customer information, and more. In hotel management system project in C++, file handling used to perform various features and functions like adding items in food menu, deleting food menu & creating, editing, listing and deleting customer details.

Understanding this project will help you effective use of file handling and graphics in a C++ project.

Here in the source code, I have listed the functions under different classes to help you understand the project better. I have briefly explained the features and functions of this hotel management project. The project may seem complex to you as it is long and utilizes graphics, so first go through the explanation provided here and try understanding and analyzing the project on your own.

1. Food menu: This function serially lists the different food items available in the hotel food menu along with their price. Here, this function also allows user to add food items or delete all food items to or from the food menu by calling the respective functions.

* Add items in food menu: Users can get to this “append in food menu” function from the displaying food menu function. Here, the information to be provided are the item name to be added and the price of the food item. Then, the hotel management system project in C++ stores all the added food items in file.
* Delete food menu: One room for improvement in this hotel management system program is regarding this function. After displaying the food menu from the first function, with this function user can delete all the items from the food menu, i.e, delete the food menu itself. I’d recommend you to work on this function, and modify it so that only certain items as chosen by the user can be deleted form the menu.

2. Customer bill: The customer bill consists of the room bill and the food bill functions. So, it is more like an overall hotel bill.

* Food bill: This function is used to issue a bill to the customer on the basis of the item and quantity of food they ate or purchased. It asks for the name of the customer, the name of the food item and the quantity. Then, the function lists these info in a bill in tabular form. If a food item asked by the customer is not found in the file, the hotel management system project in C++ displays the message – “Item is not present.”
* Room bill: Here the user needs to provided the number of the departing customer, along with the date and month of departure. Then, the room bill is displayed in a tabular form – Name, Arrival, Departure, Days in and Bill. If incorrect info regarding the customer number is provided it displays the message – “Customer is not present.”

3. Customer details: This is a very large function that consists of other sub-functions as adding, modifying and deleting customer details from the file of hotel management system project in C++. All the added, modified and deleted information regarding customer details are stored in file. Here, I will briefly describe each of these functions.

* Add customer detail: The information asked for by this function are name and address of the customer, and date, month and year of the customer’s arrival.
* Display customer details: This function displays the details of all the customers in the hotel. The info like customer number, name, address, room number and date are listed in a tabular form.
* Modify customer detail: To modify customer detail in hotel management system project in C++, customer number is sought from the user. Then, the name, address, date, month and year of arrival of a particular customer can be modified, and all the edited data are stored in file.
* Delete a customer record: For this, the user needs to provide the customer number of a particular customer whose record is to be deleted from the hotel management system file.
* Delete all customer records: If the user wants to delete the records of all the customers added in the hotel management file, this function can be used.

4. Header Files Used:

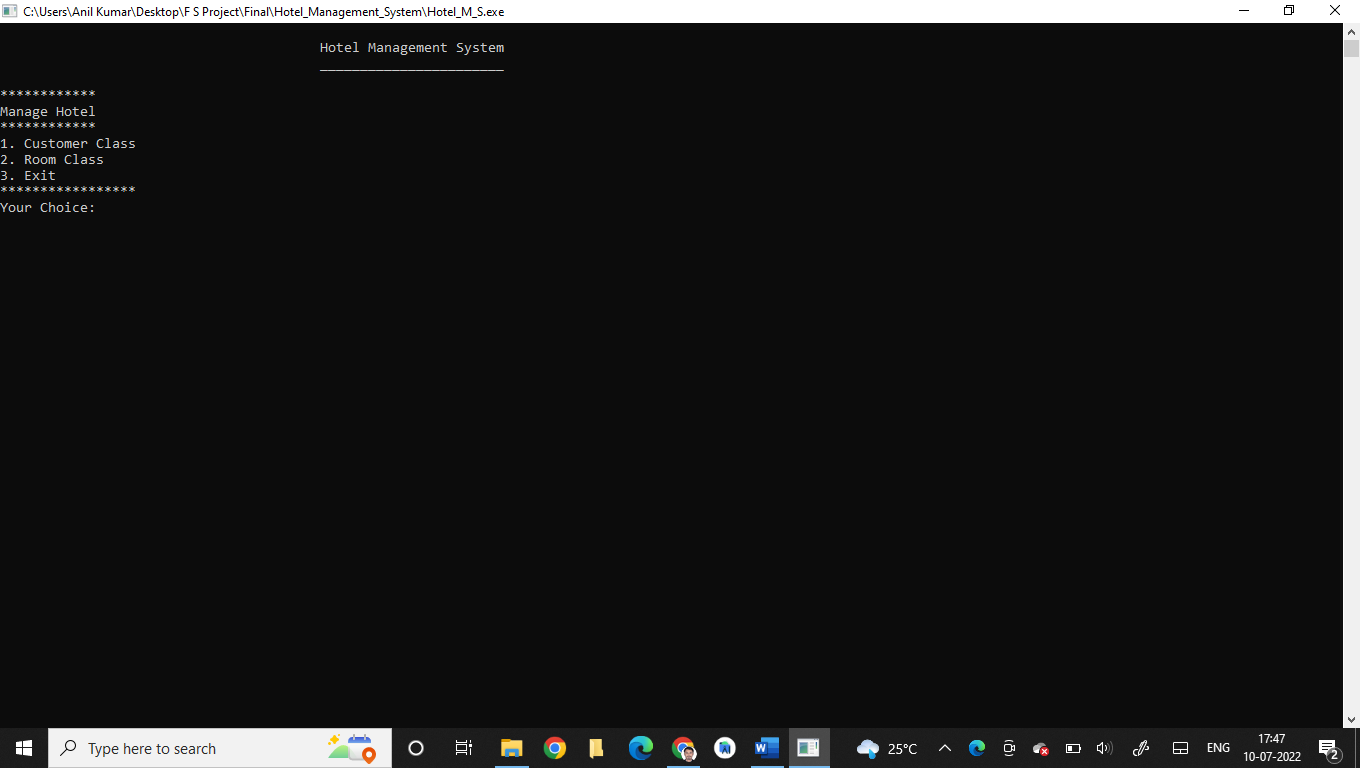
This project utilizes 9 header files, all of which are included in Turbo C++ library file. The graphics.h header file has been used to make the project better, and its use is simple to understand and analyze from the source code.

* #include<iostream .h>
* #include<conio .h>
* #include<string .h>
* #include<graphics .h>
* #include<dos .h>
* #include<stdio .h>
* #include<fstream .h>
* #include<iomanip .h>
* #include<stdlib .h>

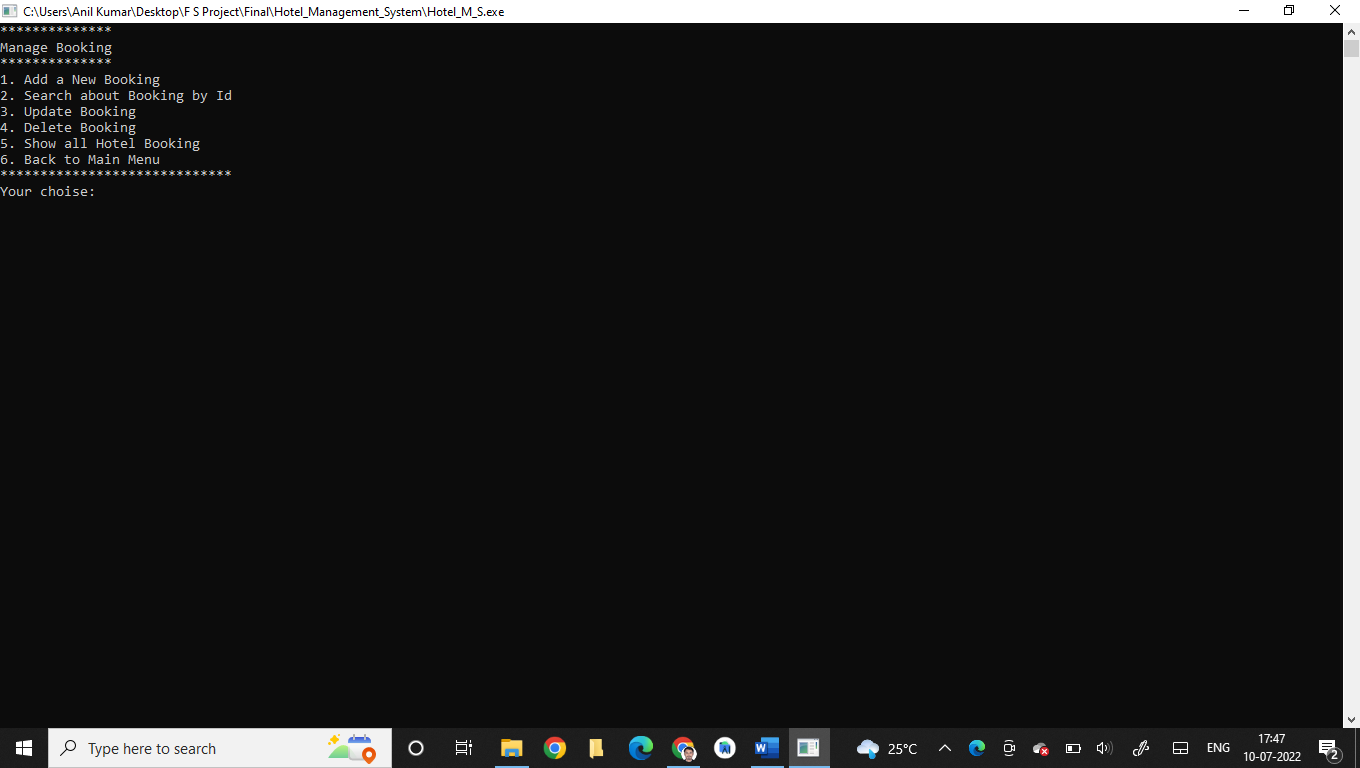
**CHAPTER 5**

**SAMPLE OUTPUT**

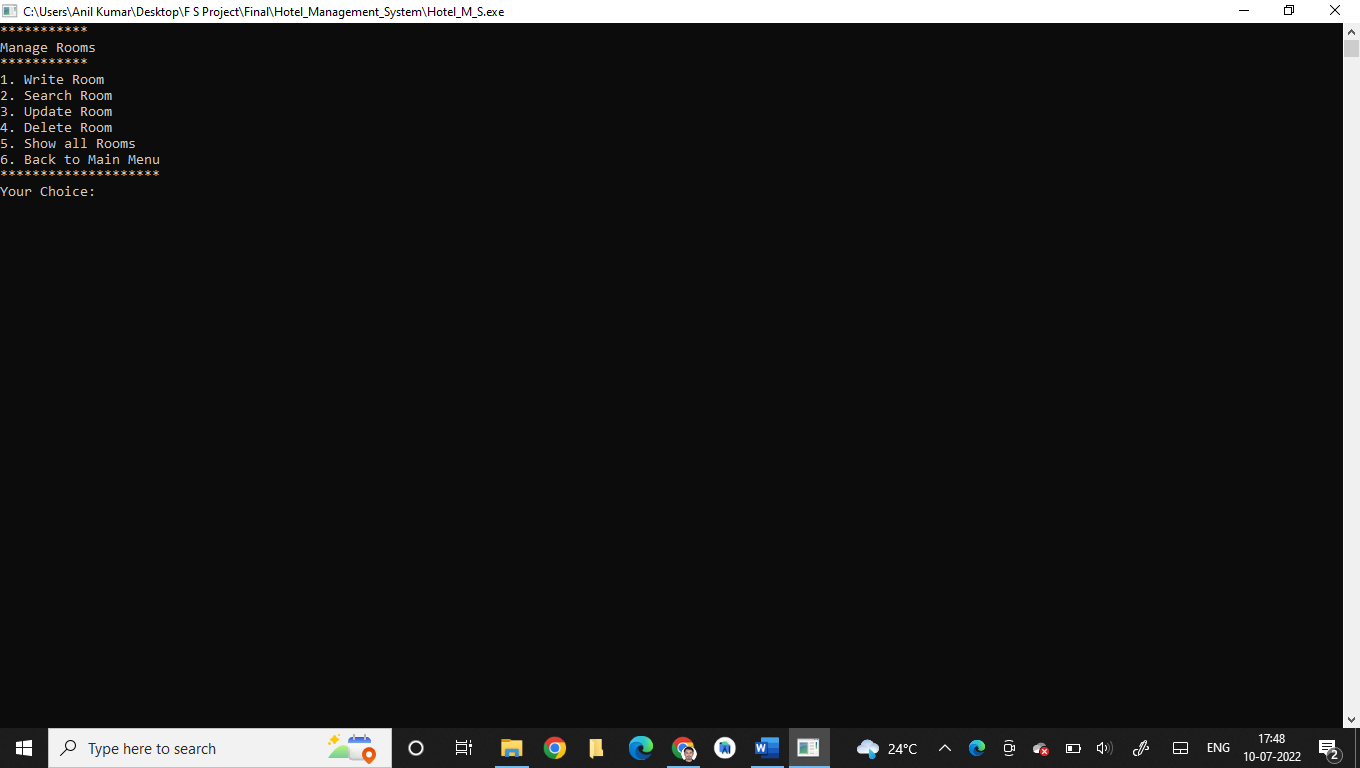
**5.1 Screenshots**



**Figure 5.1 Main Menu**



**Figure 5.2 Manage Booking**



**Figure 5.3 Manage Rooms**

**CHAPTER 6**

**CONCLUSION**

HOTEL MANAGENMENT SYSTEM is a standalone application. This project is designed to meet the requirements of Online Hotel Management. It has been developed in C++ and File structure keeping in mind the specifications of the system. For designing the system, we have used simple user requirements. Overall, the project teaches us the essential skills like: Using system analysis and design techniques. Sometimes it happens that the rooms get booked soon when one visits the place therefore user can make advance booking using this system. It saves user time in search of rooms. The system is useful as it calculates an exact cost for requested number of days. It saves organization resources and expenses. This system is effective and saves time and cost of users.

**BIBLIOGRAPHY**

[1] https://www.codewithc.com/mini-project-in-c-medical-store-management-system/

[2] https://www.codewithc.com

[3] [www.youtube.com](http://www.youtube.com)

[4] [www.quora.com](http://www.quora.com)

[5] [www.wikipedia.org](http://www.wikipedia.org)

[6] <https://www.w3schools.com/cpp/cpp_files.asp>

[7] https://www.geeksforgeeks.org/introduction-to-c-programming-language/