Hospital Management System

Project Report

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD
OF THE DEGREE OF

Bachelor Of Technology
Computer Science Engineering

Submitted By

Raj Vardhan Singh
(2103490100026)

September 2022

Dr. APJ Abdul Kalam Technical University, Uttar Pradesh
Lucknow, INDIA

CONTENT OF REPORT :

OBJECTIVE:

The hospital management system project is a basic C Programming language program. This project can be used to improve coding skills and logic building abilities. It is meant to store all the data and information of a hospital Including patients data and data related to the availability of the doctors. This project will help in making the older way of storing data more efficient and easy.

This discussion gives you the whole report on the Hospital Management System Project along with its abstract and modules. It enables you to be knowledgeable of the project's core functions and processes by knowing the purposes of the Hospital Management System. The Hospital Management System Project also comes with PDF Documentation reports and the abstract of every source code available.

The purpose of the project entitled as "HOSPITAL MANAGEMENT SYSTEM" is to computerize the Front Office Management of Hospital to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient's information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is to register and store patient details and doctor details and retrieve

The main objectives of this hospital management system project are :-

- To simplify the cumbersome work of storing the data related to the hospital.
- To provide an organized way of storing and managing the information of the hospitals.
- This new system will make the time of admission of patients very less as compared to the older system.
- Managing the work of doctors and will help in getting a full record of the patient.
- Design a system for better patient care.
- Reduce hospital operating costs.
- Provide MIS (Management Information System) report on demand to management for better decision making.

- Better co-ordination among the different departments.
- Provide top management a single point of control.
- Hospital Management System handles activities of major departments in a hospital like:
- 1.Front Office/ OPD Management
- 2. Patient management (scheduling, registration and long-term care)
- 3. Patient care management and departmental modules (radiology, pharmacy and pathology labs)
- 4. Investigative Labs
- 5. Billing
- 6. Medical Stores
- 7. Financial Accounting (billing, insurance processing, materials management, accounts payable/receivable, payroll and general ledger)
- 8. Payroll
- Hospital management system can be developed by using waterfall model which is a popular version of development life cycle model for software engineer. It describes a development method that is linear and sequential. It has distinct goals for each phase of development. In this model once, a phase of development is completed, there is no turning back, the development proceeds to the next phase. The advantage of this model is that it allows for departmentalization and managerial control.

INTRODUCTION:

The project Hospital Management system Is designed to make the cumbersome process of storing the information and data related to a hospital more easier and effective. This will help in reducing the efforts and human power to store and manage the data. It will also make the data available urgently whenever it is required. This project contains the basic words of C programming language like structures, switch statements, calling of functions, functions, conditional statements, while loops and a much detailed use of file handling.

The Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow. Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital and helps you manage your processes.

Human Resources Personnel managing the timekeeping will not be the same again as they will experience relief. It would be fast and easy for them to handle transactions such as report making and monitoring time entries. Further determining or computing the salary-related manually will be eliminated that will lead to faster transactions.

Thus, this mini project on Hospital management system will help a lot in the functioning of the hospital and will help a lot in reducing manwork.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

The Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes

Purpose: Main aim of developing a Hospital management System is to provide an easy way not only to automate all functionalities involved managing leaves and Payroll for the employees of Company, but also to provide full functional reports to management of Hospital with the necessary details. HOSPITAL MANAGEMENT SYSTEM 2016 10 Nowadays large scale organizations (hospitals in this case) are committed to bring the best way of management in the various forms of HMS.

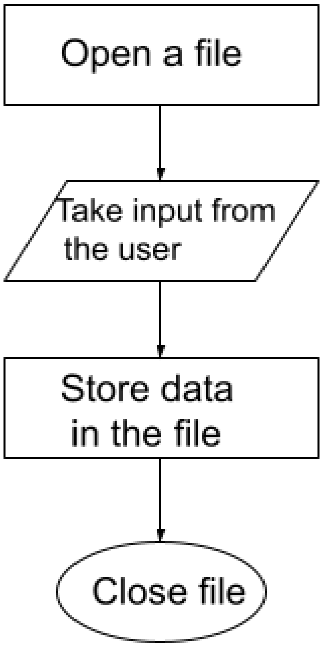
The purpose of the project entitled as "HOSPITAL MANAGEMENT SYSTEM" is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of patient's information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Hospital Management System can be entered using a user name and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

STRUCTURE OF PROGRAM :

The program is mainly Divided into a few modules containing **Functions** of different operations which are ultimately called under a **SWITCH** statement. Each MODULE is so made that it makes the code understandable and easy to Manipulate. The MODULES of the program are:-

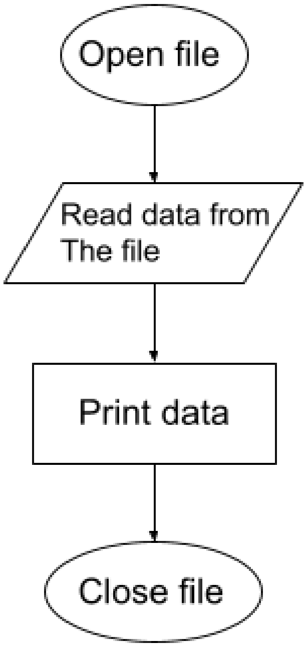
❖ MODULE 1:

This module consists of the function to admit the patient and is called under a SWITCH statement. Which uses the structure of the patient's information Denoted by struct patient. It is primarily a function that is designed to read the data from the user and store it through structure in a file.



❖ MODULE 2:

This module contains a function that is used to check the list of the patients admitted in the hospital along with the progress report in their disease or health condition. It also stores the data in a file named as patient. This module opens the file only for reading mode. Its function is to read the data from the file and print it on the screen.

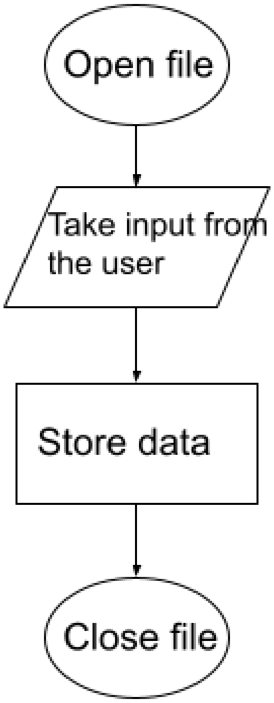


❖ MODULE 3:

Calling of this module provides the user the facility to discharge the patient that has been admitted to the program or the file via MODULE 1. It uses the basic functions of file handling and works by comparing the ID of patients one by one. If the comparison is negative then paste the data to a new file and. If the result is positive then left that data to the previous file. Now rename the new file with the older one.

❖ MODULE 4:

The function available in this module is of insertion type that is used to insert a new doctor to the list of the doctors of the hospital. It uses a structure named as struct doctor. This module is the same as the first one.



❖ MODULE 5:

It is used to see the list of the doctors available in the hospital at the current time. It will help in checking the availability of the doctor at a condition of emergency. It also works on the file handling facility of the C programming language.

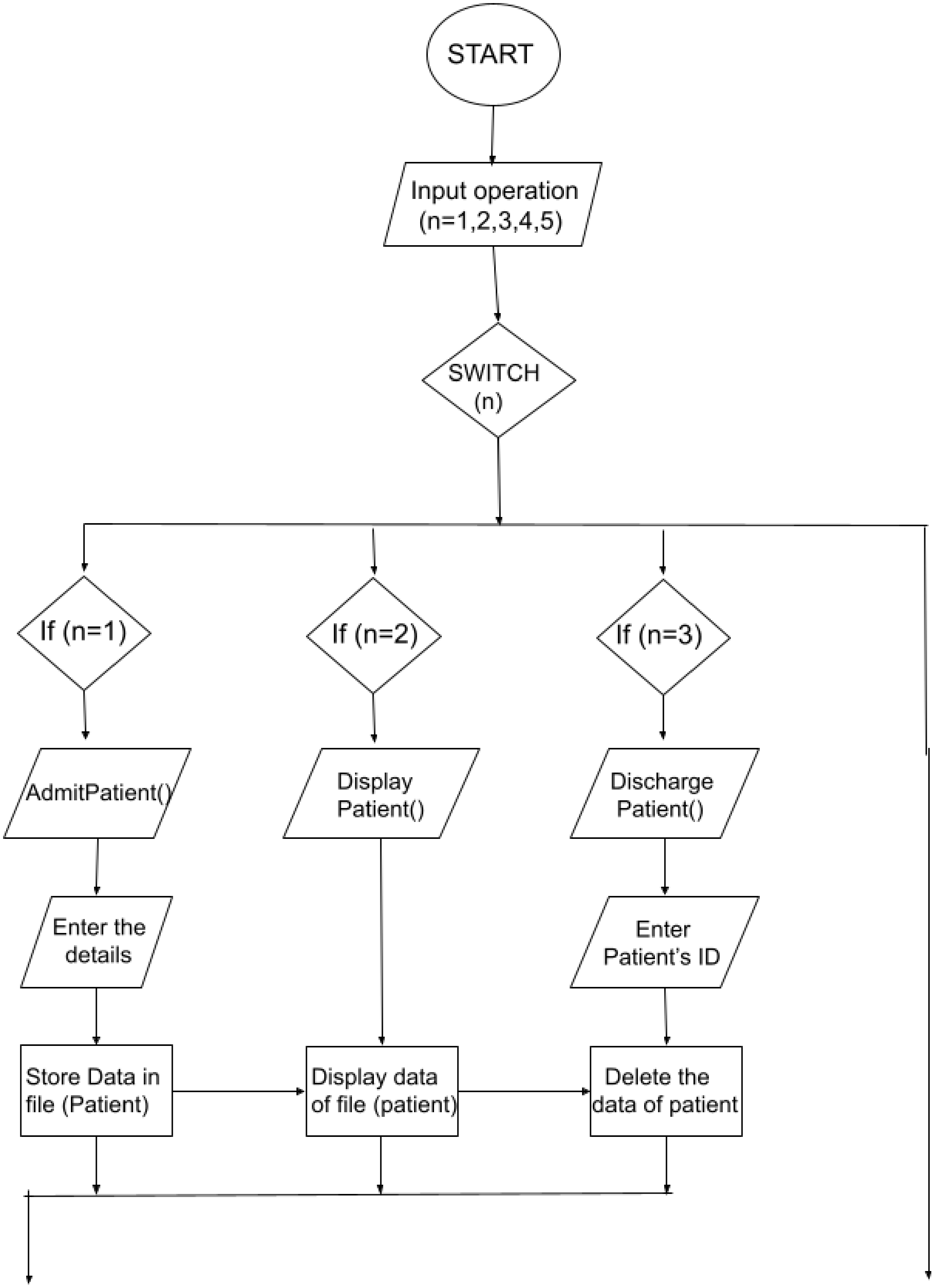
SOFTWARE REQUIRED :

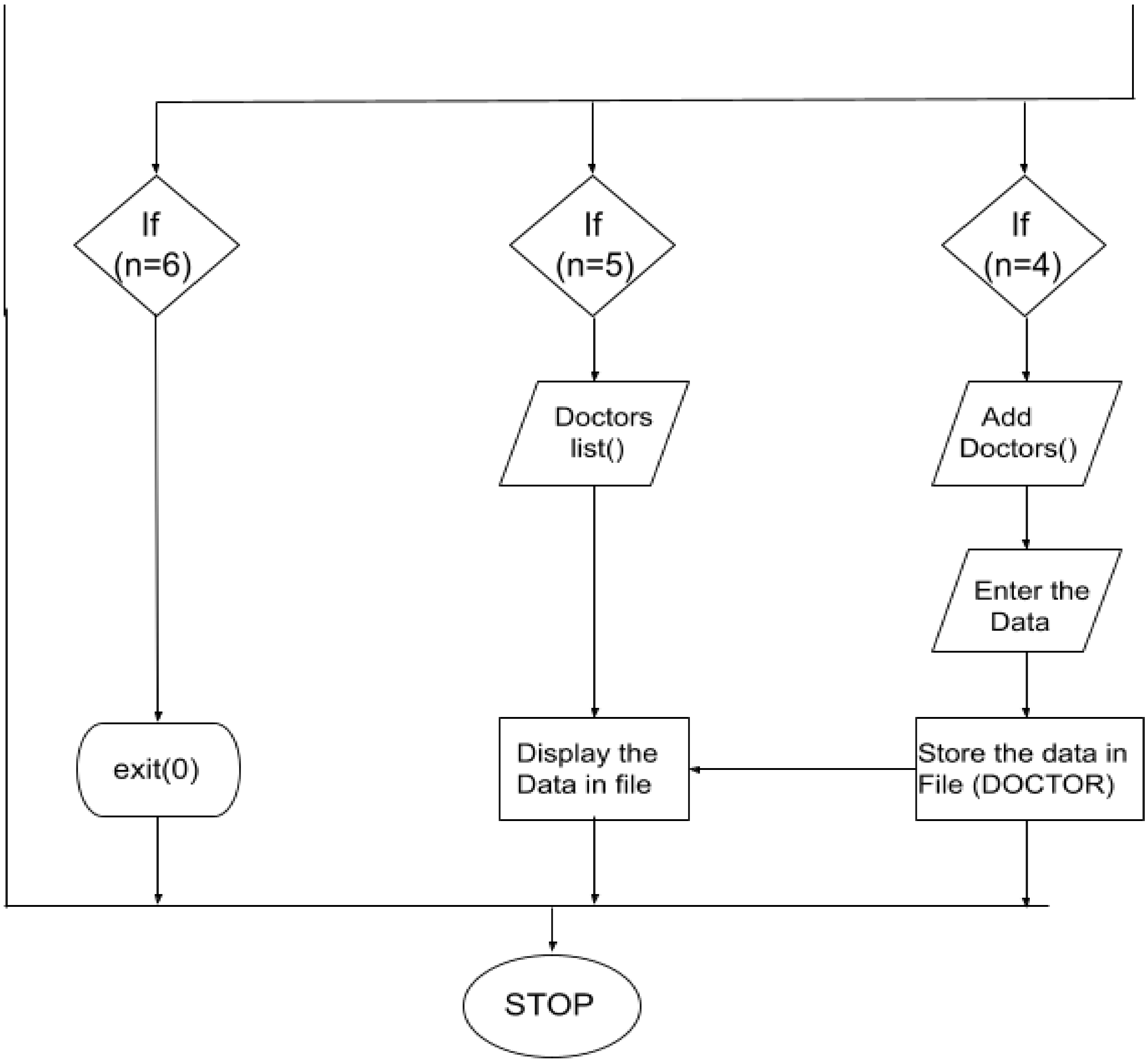
Software	Minimum Recommended Requirement	Maximum Recommended Requirement
System type	Microsoft Windows 7 (32 bits), Windows 8 (32 bits), ubuntu, linux.	Microsoft Windows 10, Windows 11.
Storage	FAT File System	NTFS File System
Programming Language Compiler	Turbo C for windows 7.	Turbo C for windows 10,11, Dev c/c++, and other compilers.

HARDWARE REQUIRED :

Hardware	Minimum recommended requirement	Maximum recommended Requirement
RAM	2 GB	3 GB OR More
Hard Disk Storage	20 GB	30 GB or More
Processor	Intel i3 10th gen, AMD Ryzen 3……	Intel i5 10th gen, i5 11th gen, i7, AMD Ryzen 5, Ryzen 7…..

FLOWCHART :


START
Input operation (n=1,2,3,4,5)
SWITCH (n)
If (n=1)
If (n=2)
If (n=3)
AdmitPatient()
Display Patient()
Discharge Patient()
Enter the details
Enter Patient's ID
Store Data in file (Patient)
Display data of file (patient)
Delete the data of patient



If (n=6)
If (n=5)
If (n=4)
Doctors list()
Add Doctors()
Enter the Data
exit(0)
Display the Data in file
Store the data in File (DOCTOR)
STOP

LITERATURE REVIEW:

As we have many industries turning towards the digital front and it could be a very great move for the legacy and necessary industry such as Hospitals to move towards that direction. The current existing module is efficient but when the time is not a constraint. We can not have this system when each and every second matters. This system should include many features in the online front that include the patients records including his disease history and reports. Above mentioned data can be accessed by the respective doctor from anywhere around the globe. The storage of all these details would be done by setting up a database server. If a patient is admitted to the hospital, all the vital details would be updated for the doctors to check it online. They can even give online prescriptions directly to the pharmacy specific to a particular patient with their patient id. Every person who visits the website can register themselves as a patient and get an unique Patient ID that is referred to in all the future transactions. Apatient can take the appointments online and know the availability of the doctor. Notifications regarding the regular health check-ups and medicine reorders. Tips on regular better lifestyle and good Health are provided. Now-a-days each and every individual is bound to have a smart device that connects him to the world of the internet and that's when the speed of data transfer or data availability comes into picture. This digital approach would help many people who are in need of medical services for small inconveniences and are unable to travel for the necessary medical treatment. Paper is organised as follows, in Section 2, literature survey is discussed. Section 3 presents the methodology, Section 4 presents the proposed system, Section 5 contains the methodology of the project and Section 6 discusses the Results and discussions of the project, Section 7 discusses the conclusions and finally the future enhancements for the project. [9]

There are two modules that are identified, Patient and Doctor modules. The patient module is used to book appointments, make payments, view reports and see their medical as well as payment history. The patient module also contains a disease prediction section where the patients Whereas the doctor module has the User Interface(UI) for the employees to access the database. There are four types of users in the Doctor module. They are Admins, Doctor , Lab staff and Pharmacy staff. They all have hierarchical access to the database. Admin is responsible for adding the users to the database and giving access based on their designation. Users with Doctor access can view his patients details, give medicines and view the tests assigned to him. Lab staff are incharge of the payment section and reports. Pharmacy staff can add or remove the medicine details and dispatch the medicines based on the payment details.[9].

Abstract:

Health care in India as in many other countries is confronted with growing demand for medical treatment and services. The medical records must appropriately have all of the patients' medical history. Physicians must maintain flawless records, because this document serves a number of purposes. This study on hospital management system is design to transform the manual way of searching, sorting, keeping and accessing hospital information (files) into electronic medical record in order to solve the problem associate with manual method. The existing system has been studied and hence computer based software was provided to replace this manual method. These computer based systems generate the patient report as the patient register in and out of the hospital. It also generate the information regarding the doctor, nurse allot to the patient. This paper generally looks for a more accurate, reliable and efficient method of computer to facilitate hospital record's keeping in General Hospitals to ensure efficient outcome that will lessen time consuming. The study proposed that the design of hospital management record will be a solution to the problem being experienced by the current manual method of keeping patient medical record. Registration of patients, storing their details into the system and also computerized billing in the pharmacy, and labs. Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current Status of each room. User can search availability of a doctor and the details of a patient using the id. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast. [10]

BENEFITS:

Conceptually, requirements analysis includes three types of activities:

Eliciting requirements:(e.g. the project charter or definition), business process documentation, and stakeholder interviews. This is sometimes also called requirements gathering or requirements discovery.

Analyzing requirements: determining whether the stated requirements are clear, complete, consistent and unambiguous, and resolving any apparent conflicts.

Recording requirements: Requirements may be documented in various forms, usually including a summary list and may include natural-language documents, use cases, user stories, process specifications and a variety of models including data models.

The system should also embrace the following requirements:

· **User-friendly**: The system must accommodate a clearly understandable user interface as well as documentation help at any stage of the user interaction with the system.

· **Security**: The system shake it impossible for anybody to logon without a valid username and password. Data encryption should be employed to keep the user login name and password secret.

· **Reliability**: The system would be used by the accounting section of any organization. Since this application is subject to process monetary matters, this must be reliable to the users of this application.

· **Ease of Use**: The views and operations should be easy to use and intuitive. Documentation should be provided.

· **Performance:** The system should have a quick response time.

· **System requirements**: This system would be designed to run on a minimum hardware configuration like 500MHz x86 machines. Considering the vast hardware available nowadays, this would not pose any problems.[10]

Management has been defined as the process, comprised of social and technical functions and activities, occurring within organizations for the purpose of accomplishing predetermined objectives through humans and other resources (Longest, Rakich & Darr, 2000). Healthcare quality and patient safety are the common mantra of all primary and secondary health care providers. In hospitals, over the years, a variety of models and schemes for hospital interventions and development have been deployed (Friesner, 2009). Hospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. There are different modules in the process of Hospital Management System. These include: Patient management, Services management, Appointment scheduling, Store management, Pharmacy management3, Hospital Information systems are in high demand to handle increasing population needs and also aids the practicing doctors and hospital service and support staff with timely service and precision. There are varied metrics available to assess the performance of services like hospital industry, and the successful implementation and usage of Hospital information system forms a crucial role. Hospital information systems are available in the soft ware market which in most cases needs to be customized and in some cases HIS needs to be developed as a customized software based on specific hospital requirements (user requirements). The paper looks at assessing and identifying the key components4 . "Advanced Hospital Management System" includes Registration of patients, storing their details into the system and also computerized billing in the pharmacy, and labs. Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current Status of each room. User can search availability of a doctor and the details of a patient using the id5 . The Advanced Hospital Management System can be entered Using a username and password. It is accessible either by an administrator or Receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast6 . Existing system refers to the system that is being followed till now. Presently all the hospital functionalities are done manually. That is if a Patient want to consult a doctor he can visit their till his chance called. This is make the person very difficult. Outpatients and Inpatient tickets are Distributed directly. The main disadvantage is time consuming. [9]

TECHNOLOGIES USED:

C (*pronounced like the letter C* is a general-purpose computer programming language. It was created in the 1970s by Dennis Ritchie, and remains very widely used and influential. By design, C's features cleanly reflect the capabilities of the targeted CPUs. It has found lasting use in operating systems, device drivers, protocol stacks, though decreasingly[7] for application software. C is commonly used on computer architectures that range from the largest supercomputers to the smallest microcontrollers and embedded systems.

A successor to the programming language B, C was originally developed at Bell Labs by Ritchie between 1972 and 1973 to construct utilities running on Unix. It was applied to re-implementing the kernel of the Unix operating system. During the 1980s, C gradually gained popularity. It has become one of the most widely used programming languages, with C compilers available for almost[] all modern computer architectures and operating systems. C has been standardized by ANSI since 1989 (ANSI C) and by the International Organization for Standardization (ISO).

C is an imperative procedural language supporting structured programming, lexical variable scope, and recursion, with a static type system. It was designed to be compiled to provide low-level access to memory and language constructs that map efficiently to machine instructions, all with minimal runtime support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant C program written with portability in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code.

C is an imperative, procedural language in the ALGOL tradition. It has a static type system. In C, all executable code is contained within subroutines (also called "functions", though not in the sense of functional programming). Function parameters are passed by value, although arrays are passed as pointers, i.e. the address of the first item in the array. *Pass-by-reference* is simulated in C by explicitly passing pointers to the thing being referenced.

C program source text is free-format, using the semicolon as a statement separator and curly braces for grouping blocks of statements.

The C language also exhibits the following characteristics:

- The language has a small, fixed number of keywords, including a full set of control flow primitives: if/else, for, do/while, while, and switch. User-defined names are not distinguished from keywords by any kind of sigil.
- It has a large number of arithmetic, bitwise, and logic operators: +,+=,++,&,||, etc.
- More than one assignment may be performed in a single statement.
- Functions:
    - Function return values can be ignored, when not needed.
    - Function and data pointers permit *ad hoc* run-time polymorphism.
    - Functions may not be defined within the lexical scope of other functions.
    - Variables may be defined within a function, with scope.
    - A function may call itself, so recursion is supported.
- Data typing is static, but weakly enforced; all data has a type, but implicit conversions are possible.
- User-defined (typedef) and compound types are possible.
    - Heterogeneous aggregate data types (struct) allow related data elements to be accessed and assigned as a unit.
    - Union is a structure with overlapping members; only the last member stored is valid.
    - Array indexing is a secondary notation, defined in terms of pointer arithmetic. Unlike structs, arrays are not first-class objects: they cannot be assigned or compared using single built-in operators. There is no "array" keyword in use or definition; instead, square brackets indicate arrays syntactically, for example month.

    - Enumerated types are possible with the enum keyword. They are freely interconvertible with integers.
    - Strings are not a distinct data type, but are conventionally implemented as null-terminated character arrays.
- Low-level access to computer memory is possible by converting machine addresses to pointers.
- Procedures (subroutines not returning values) are a special case of function, with an untyped return type void.
- Memory can be allocated to a program with calls to library routines.
- A preprocessor performs macro definition, source code file inclusion, and conditional compilation.
- There is a basic form of modularity: files can be compiled separately and linked together, with control over which functions and data objects are visible to other files via static and external attributes.
- Complex functionality such as I/O, string manipulation, and mathematical functions are consistently delegated to library routines.
- The generated code after compilation has relatively straightforward needs on the underlying platform, which makes it suitable for creating operating systems and for use in embedded systems.

While C does not include certain features found in other languages (such as object orientation and garbage collection), these can be implemented or emulated, often through the use of external libraries (e.g., the GLib Object System or the Boehm garbage collector).

SYNTAX:

C has a formal grammar specified by the C standard.[26] Line endings are generally not significant in C; however, line boundaries do have significance during the preprocessing phase. Comments may appear either between the delimiters /* and */, or (since C99) following // until the end of the line. Comments delimited by /* and */ do not nest, and these sequences of characters are not interpreted as comment delimiters if they appear inside string or character literals.[27]

C source files contain declarations and function definitions. Function definitions, in turn, contain declarations and statements. Declarations either define new types using keywords such as struct, union, and enum, or assign types to and perhaps reserve storage for new variables, usually by writing the type followed by the variable name. Keywords such as char and int specify built-in types. Sections of code are enclosed in braces ({ and }, sometimes called "curly brackets") to limit the scope of declarations and to act as a single statement for control structures.

CHARACTER SET:

The basic C source character set includes the following characters:

- Lowercase and uppercase letters of ISO Basic Latin Alphabet: a–z A–Z
- Decimal digits: 0–9
- Graphic characters: ! " # % & ' ( ) * + , - . / : ; < = > ? [\ ] ^ _ { | } ~
- Whitespace characters: *space*, *horizontal tab*, *vertical tab*, *form feed*, *newline*

Newline indicates the end of a text line; it need not correspond to an actual single character, although for convenience C treats it as one.

Additional multi-byte encoded characters may be used in string literals, but they are not entirely portable. The latest C standard (C11) allows multi-national Unicode characters to be embedded portably within C source text by using \uXXXX or \UXXXXXXXX encoding (where the X denotes a hexadecimal character), although this feature is not yet widely implement.

OPERATORS:

C supports a rich set of operators, which are symbols used within an expression to specify the manipulations to be performed while evaluating that expression. C has operators for:

- arithmetic: +, -, *, /, %
- assignment: =
- augmented assignment: +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=
- bitwise logic: ~, &, |, ^
- bitwise shifts: <<, >>
- boolean logic: !, &&, ||
- conditional evaluation: ? :
- equality testing: ==, !=
- calling functions: ( )
- increment and decrement: ++, --

C uses the operator = (used in mathematics to express equality) to indicate assignment, following the precedent of Fortran and PL/I, but unlike ALGOL and its derivatives. C uses the operator == to test for equality. The similarity between these two operators (assignment and equality) may result in the accidental use of one in place of the other, and in many cases, the mistake does not produce an error message (although some compilers produce warnings). For example, the conditional expression if (a == b + 1) might mistakenly be written as if (a = b + 1), which will be evaluated as true if a is not zero after the assignment.

DATA TYPES:

The type system in C is static and weakly typed, which makes it similar to the type system of ALGOL descendants such as Pascal.[35] There are built-in types for integers of various sizes, both signed and unsigned, floating-point numbers, and enumerated types (enum). Integer type char is often used for single-byte characters. C99 added a boolean datatype. There are also derived types including arrays, pointers, records (struct), and unions (union).

C is often used in low-level systems programming where escapes from the type system may be necessary. The compiler attempts to ensure type correctness of most expressions, but the programmer can override the checks in various ways, either by using a *type cast* to explicitly convert a value from one type to another, or by using pointers or unions to reinterpret the underlying bits of a data object in some other way.

POINTERS:

C supports the use of pointers, a type of reference that records the address or location of an object or function in memory. Pointers can be *dereferenced* to access data stored at the address pointed to, or to invoke a pointed-to function. Pointers can be manipulated using assignment or pointer arithmetic. The run-time representation of a pointer value is typically a raw memory address (perhaps augmented by an offset-within-word field), but since a pointer's type includes the type of the thing pointed to, expressions including pointers can be type-checked at compile time. Pointer arithmetic is automatically scaled by the size of the pointed-to data type.Pointers are used for many purposes in C. Text strings are commonly manipulated using pointers into arrays of characters. Dynamic memory allocation is performed using pointers; the result of a malloc is usually cast to the data type of the data to be stored. Many data types, such as trees, are commonly implemented as dynamically allocated struct objects linked together using pointers. Pointers to other pointers are often used in multi-dimensional arrays and arrays of struct objects. Pointers to functions (*function pointers*) are useful for passing functions as arguments to higher-order functions (such as qsort or bsearch), in dispatch tables, or as callbacks to event handlers .

ARRAY:

Array types in C are traditionally of a fixed, static size specified at compile time. The more recent C99 standard also allows a form of variable-length arrays. However, it is also possible to allocate a block of memory (of arbitrary size) at run-time, using the standard library's malloc function, and treat it as an array.

Since arrays are always accessed (in effect) via pointers, array accesses are typically *not* checked against the underlying array size, although some compilers may provide bounds checking as an option. Array bounds violations are therefore possible and can lead to various repercussions, including illegal memory accesses, corruption of data, buffer overruns, and run-time exceptions.

C does not have a special provision for declaring multi-dimensional arrays, but rather relies on recursion within the type system to declare arrays of arrays, which effectively accomplishes the same thing. The index values of the resulting "multi-dimensional array" can be thought of as increasing in row-major order. Multi-dimensional arrays are commonly used in numerical algorithms (mainly from applied linear algebra) to store matrices. The structure of the C array is well suited to this particular task. However, in early versions of C the bounds of the array must be known fixed values or else explicitly passed to any subroutine that requires them, and dynamically sized arrays of arrays cannot be accessed using double indexing. (A workaround for this was to allocate the array with an additional "row vector" of pointers to the columns.) C99 introduced "variable-length arrays" which address this issue.

MEMORY MANAGEMENT:

One of the most important functions of a programming language is to provide facilities for managing memory and the objects that are stored in memory. C provides three principal ways to allocate memory for objects:

- Static memory allocation: space for the object is provided in the binary at compile-time; these objects have an extent (or lifetime) as long as the binary which contains them is loaded into memory.

- Automatic memory allocation: temporary objects can be stored on the stack, and this space is automatically freed and reusable after the block in which they are declared is exited.
- Dynamic memory allocation: blocks of memory of arbitrary size can be requested at run-time using library functions such as malloc from a region of memory called the heap; these blocks persist until subsequently freed for reuse by calling the library function realloc or free

LIBRARIES:

The C programming language uses libraries as its primary method of extension. In C, a library is a set of functions contained within a single "archive" file. Each library typically has a header file, which contains the prototypes of the functions contained within the library that may be used by a program, and declarations of special data types and macro symbols used with these functions. In order for a program to use a library, it must include the library's header file, and the library must be linked with the program, which in many cases requires compiler flags (e.g., -lm, shorthand for "link the math library").

The most common C library is the C standard library, which is specified by the ISO and ANSI C standards and comes with every C implementation (implementations which target limited environments such as embedded systems may provide only a subset of the standard library). This library supports stream input and output, memory allocation, mathematics, character strings, and time values. Several separate standard headers (for example, stdio.h) specify the interfaces for these and other standard library facilities.

Another common set of C library functions are those used by applications specifically targeted for Unix and Unix-like systems, especially functions which provide an interface to the kernel. These functions are detailed in various standards such as POSIX and the Single UNIX Specification.

SOME LIBRARY FILES IN C:

C Header Files	Description
<assert.h>	Program assertion functions
<ctype.h>	Character type functions
<locale.h>	Localization functions
<math.h>	Mathematics functions
<setjmp.h>	Jump functions
<signal.h>	Signal handling functions
<stdarg.h>	Variable arguments handling functions
<stdio.h>	Standard Input/Output functions
<stdlib.h>	Standard Utility functions
<string.h>	String handling functions
<time.h>	Date time functions

FILE HANDLING:

File input and output (I/O) is not part of the C language itself but instead is handled by libraries (such as the C standard library) and their associated header files (e.g. stdio.h). File handling is generally implemented through high-level I/O which works through streams. A stream is from this perspective a data flow that is independent of devices, while a file is a concrete device. The high-level I/O is done through the association of a stream to a file. In the C standard library, a buffer (a memory area or queue) is temporarily used to store data before it is sent to the final destination. This reduces the time spent waiting for slower devices, for example a hard drive or solid state drive. Low-level I/O functions are not part of the standard C library but are generally part of "bare metal" programming (programming that's independent of any operating system such as most embedded programming). With few exceptions, implementations include low-level I/O.

FILE HANDLING IN C:

So far the operations using C program are done on a prompt / terminal which is not stored anywhere. But in the software industry, most of the programs are written to store the information fetched from the program. One such way is to store the fetched information in a file. Different operations that can be performed on a file are:

1. Creation of a new file (fopen with attributes as “a” or “a+” or “w” or “w+”)
2. Opening an existing file (fopen)
3. Reading from file (fscanf or fgets)
4. Writing to a file (fprintf or fputs)
5. Moving to a specific location in a file (fseek, rewind)
6. Closing a file (fclose)

FUNCTIONS OF FILE HANDLING:

Description of Function	**Function in Use**
used to open an existing file or a new file	fopen()
writing data into an available file	fprintf()
reading the data available in a file	fscanf()
writing any character into the program file	fputc()
reading the character from an available file	fgetc()
used to close the program file	fclose()
used to set the file pointer to the intended file position	fseek()
writing an integer into an available file	fputw()
used to read an integer from the given file	fgetw()
used for reading the current position of a file	ftell()
sets an intended file pointer to the file's beginning itself	rewind()

STRUCTURE:

Structure in c is a user-defined data type that enables us to store the collection of different data types. Each element of a structure is called a member. Structures ca; simulate the use of classes and templates as it can store various information

Structures (also called structs) are a way to group several related variables into one place. Each variable in the structure is known as a member of the structure.

Unlike an array, a structure can contain many different data types (int, float, char, etc.).

You can create a structure by using the struct keyword and declare each of its members inside curly braces:

```
 struct MyStructure {   // Structure declaration

  int myNum;           // Member (int variable)

  char myLetter;       // Member (char variable)

}; // End the structure with a semicolon
```

SWITCH STATEMENT:

The switch statement in C is an alternative to if-else-if ladder statement which allows us to execute multiple operations for the different possible values of a single variable called switch variable. Here, We can define various statements in the multiple cases for the different values of a single variable.

1) The *switch expression* must be of an integer or character type.

2) The *case value* must be an integer or character constant.

3) The *case value* can be used only inside the switch statement.

4) The *break statement* in the switch case is not necessary. It is optional. If there is no break statement found in the case, all the cases will be executed present after the matched case. It is known as *fall through* the state of C switch statement.

First, the integer expression specified in the switch statement is evaluated. This value is then matched one by one with the constant values given in the different cases. If a match is found, then all the statements specified in that case are executed along with the all the cases present after that case including the default statement. No two cases can have similar values. If the matched case contains a break statement, then all the cases present after that will be skipped, and the control comes out of the switch. Otherwise, all the cases following the matched case will be executed.

FUNCTIONS:

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. Functions are used to perform certain actions, and they are important for reusing code: Define the code once, and use it many times.

A function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional functions.

You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division is such that each function performs a specific task.

A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function.

The C standard library provides numerous built-in functions that your program can call. For example, strcat() to concatenate two strings, memcpy() to copy one memory location to another location, and many more functions.

A function can also be referred to as a method or a subroutine or a procedure, etc.

A function definition in C programming consists of a *function header* and a *function body*. Here are all the parts of a function −

- Return Type − A function may return a value. The return_type is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword void.
- Function Name − This is the actual name of the function. The function name and the parameter list together constitute the function signature.
- Parameters − A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as an actual parameter or argument. The parameter list refers to the type, order, and number of the
- parameters of a function. Parameters are optional; that is, a function may contain no parameters.
- Function Body − The function body contains a collection of statements that define what the function does.

COMPILERS:

In computing, a compiler is a computer program that translates computer code written in one programming language (the *source* language) into another language (the *target* language). The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g. assembly language, object code, or machine code) to create an executable program.

There are many different types of compilers which produce output in different useful forms. A *cross-compiler* produces code for a different CPU or operating system than the one on which the cross-compiler itself runs. A *bootstrap compiler* is often a temporary compiler, used for compiling a more permanent or better optimized compiler for a language.

Related software include, a program that translates from a low-level language to a higher level one is a *decompiler* ; a program that translates between high-level languages, usually called a *source-to-source compiler* or *transpiler*. A language *rewriter* is usually a program that translates the form of expressions without a change of language. A *compiler-compiler* is a compiler that produces a compiler (or part of one), often in a generic and reusable way so as to be able to produce many different compilers.

FEASIBILITY STUDY :

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

7.1 Economic Feasibility

This study is carried out to check the economic impact the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products have to be purchased.

7.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is,the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for implementing this system.

7.3 Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

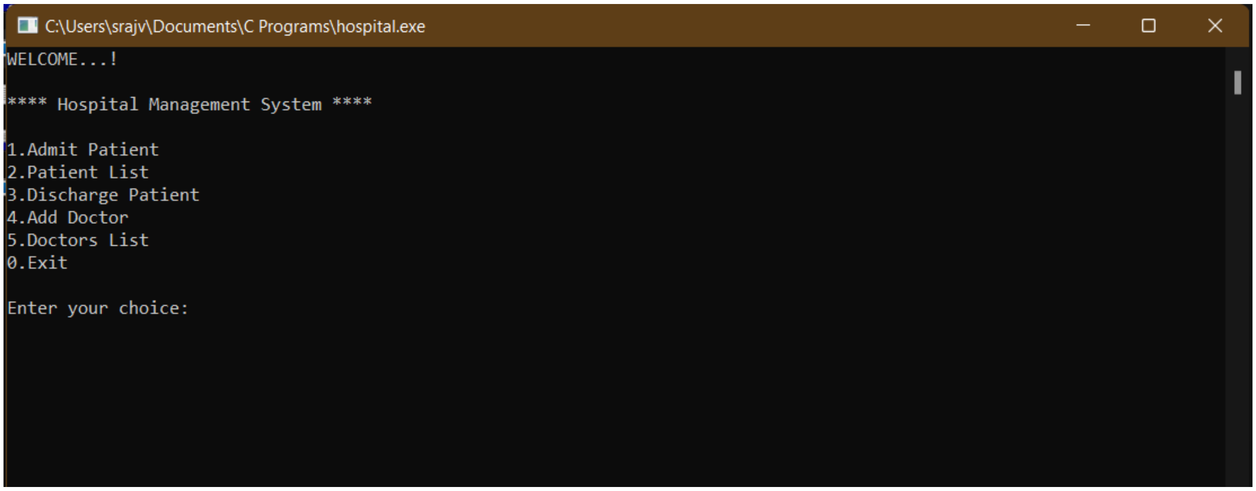
PROPOSED WORK:

This HOSPITAL MANAGEMENT SYSTEM PROJECT is oriented towards the electrification of the old school process of managing the data and information related to hospitals. It will make the process effective and Handy. So the system is proposed to store the data of patients and doctors along with all other important information. This project is designed in special modules to make the code easily re-editable and easy to transverse.Different modules perform different operations like admission and discharge of patients and the list of patients along with a list of doctors.

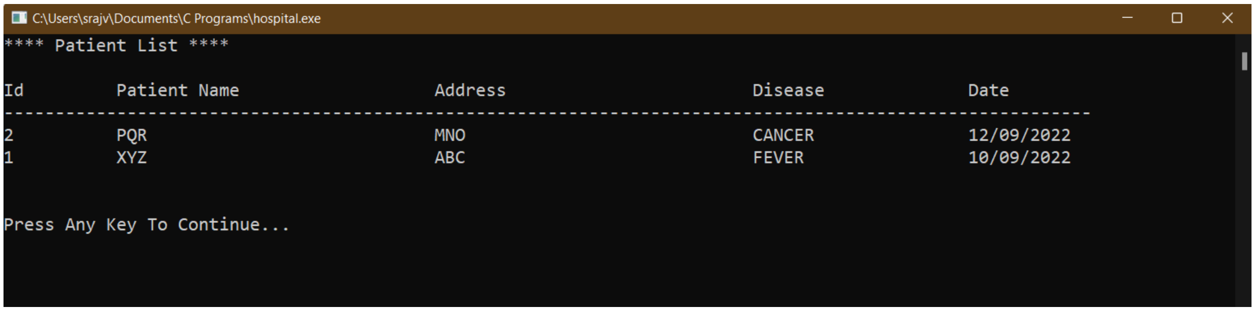
Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

The Hospital Management System is designed for any hospital to replace their existing manual paper based system. The new system is to control the information of patients. Room availability, staff and operating room schedules and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks .

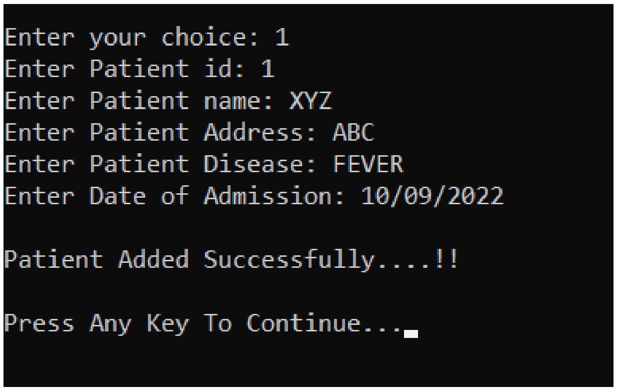
OUTPUT ANALYSIS :



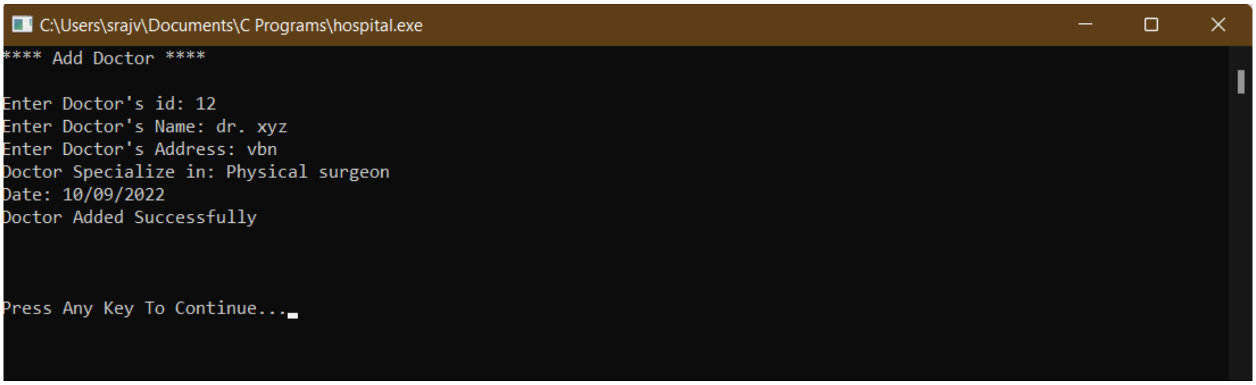
```
WELCOME...!

**** Hospital Management System ****

1.Admit Patient
2.Patient List
3.Discharge Patient
4.Add Doctor
5.Doctors List
0.Exit

Enter your choice:
```



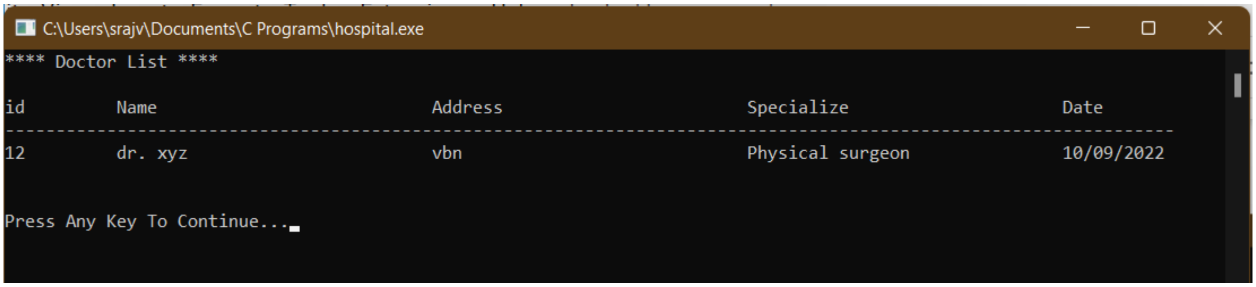
```
**** Patient List ****

Id         Patient Name                   Address                        Disease             Date
-----------------------------------------------------------------------------------------------------------
2          PQR                            MNO                            CANCER              12/09/2022
1          XYZ                            ABC                            FEVER               10/09/2022


Press Any Key To Continue...
```



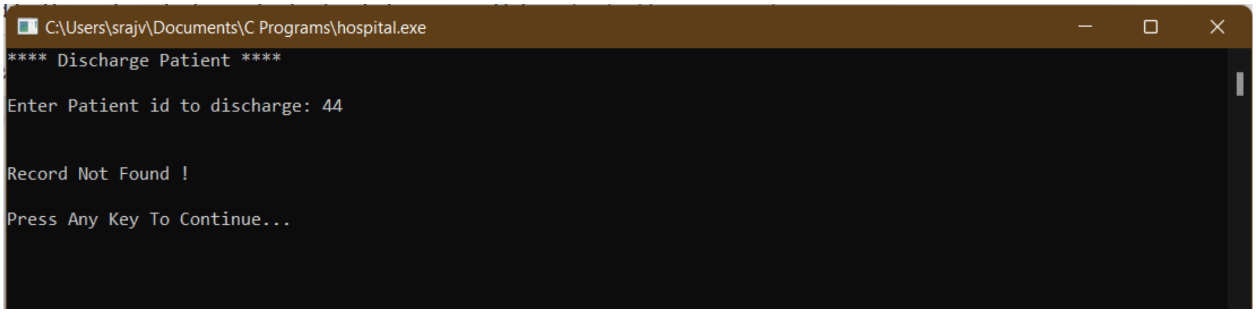
```
Enter your choice: 1
Enter Patient id: 1
Enter Patient name: XYZ
Enter Patient Address: ABC
Enter Patient Disease: FEVER
Enter Date of Admission: 10/09/2022

Patient Added Successfully....!!

Press Any Key To Continue...
```



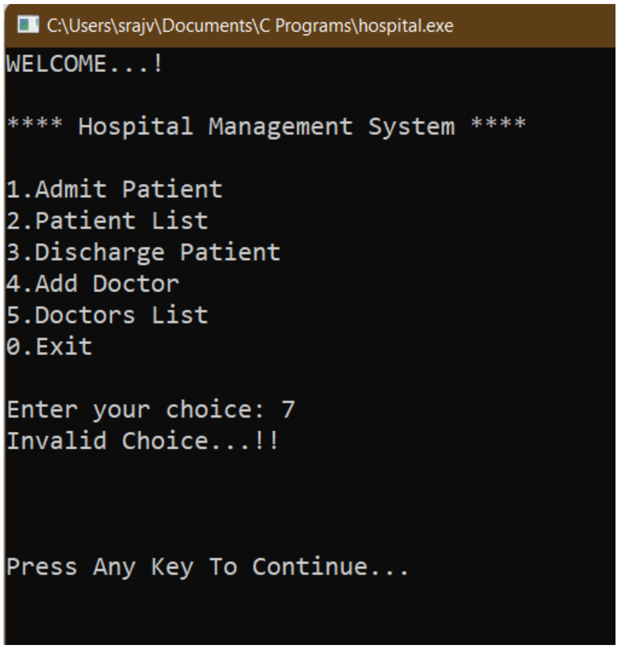
```
**** Add Doctor ****

Enter Doctor's id: 12
Enter Doctor's Name: dr. xyz
Enter Doctor's Address: vbn
Doctor Specialize in: Physical surgeon
Date: 10/09/2022
Doctor Added Successfully



Press Any Key To Continue...
```



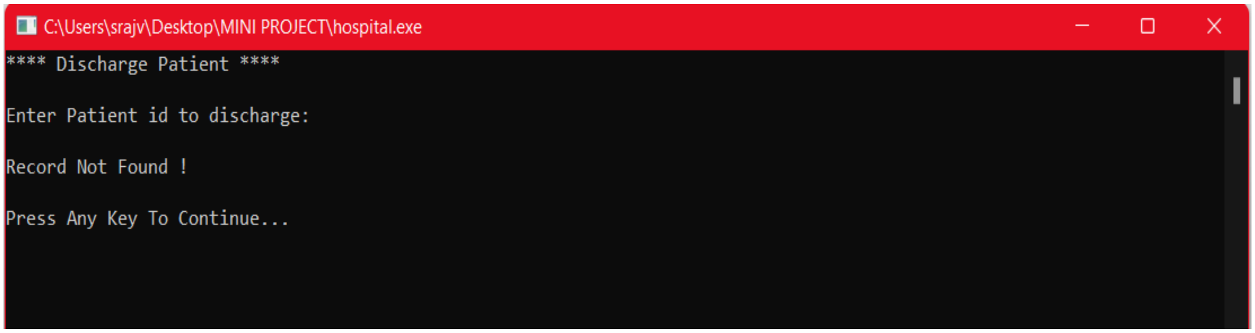
```
**** Doctor List ****

id         Name                           Address                        Specialize                     Date
-------------------------------------------------------------------------------------------------------------------
12         dr. xyz                        vbn                            Physical surgeon               10/09/2022


Press Any Key To Continue...
```



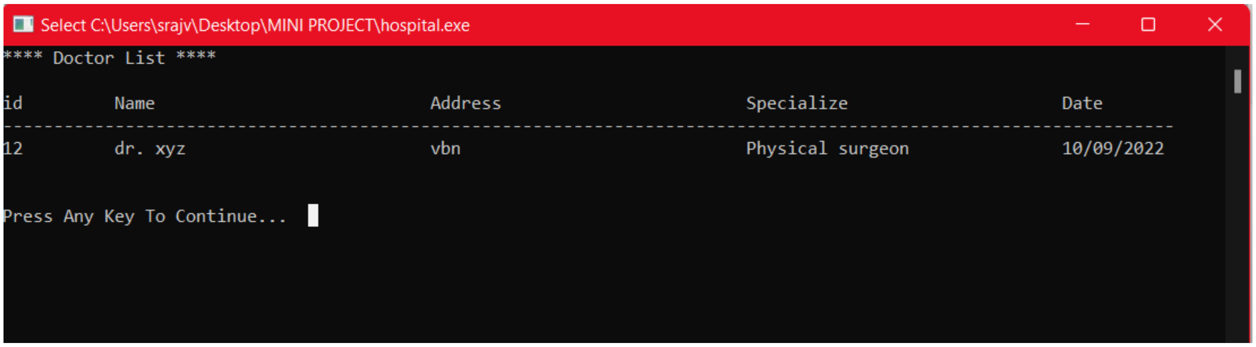
```
**** Discharge Patient ****

Enter Patient id to discharge: 44


Record Not Found !

Press Any Key To Continue...
```



```
WELCOME...!

**** Hospital Management System ****

1.Admit Patient
2.Patient List
3.Discharge Patient
4.Add Doctor
5.Doctors List
0.Exit

Enter your choice: 7
Invalid Choice...!!



Press Any Key To Continue...
```



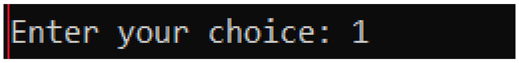
```
**** Discharge Patient ****

Enter Patient id to discharge:

Record Not Found !

Press Any Key To Continue...
```



```
**** Doctor List ****

id         Name                           Address                        Specialize                     Date
-------------------------------------------------------------------------------------------------------------------
12         dr. xyz                        vbn                            Physical surgeon               10/09/2022


Press Any Key To Continue...
```

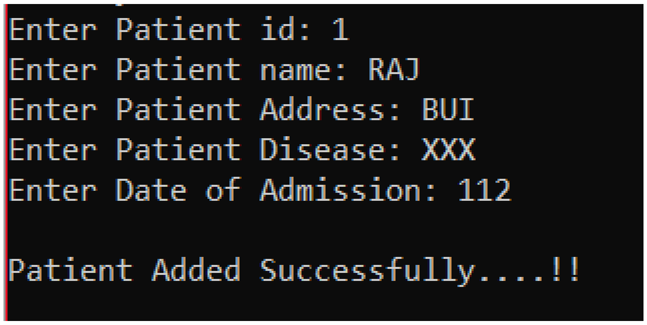
TEST/ RESULT ANALYSIS:

INPUT 01:



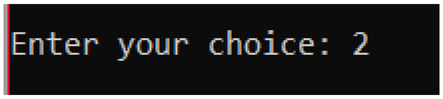
```
Enter your choice: 1
```

OUTPUT 01:



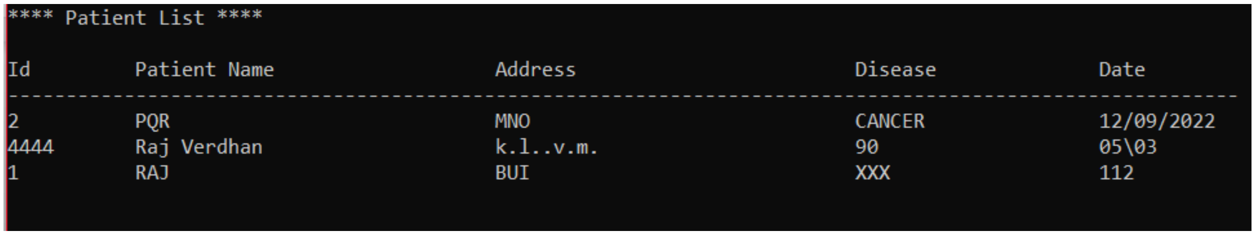
```
Enter Patient id: 1
Enter Patient name: RAJ
Enter Patient Address: BUI
Enter Patient Disease: XXX
Enter Date of Admission: 112

Patient Added Successfully....!!
```

INPUT 02:



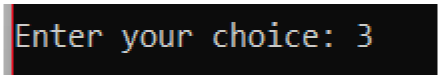
```
Enter your choice: 2
```

OUTPUT 02:



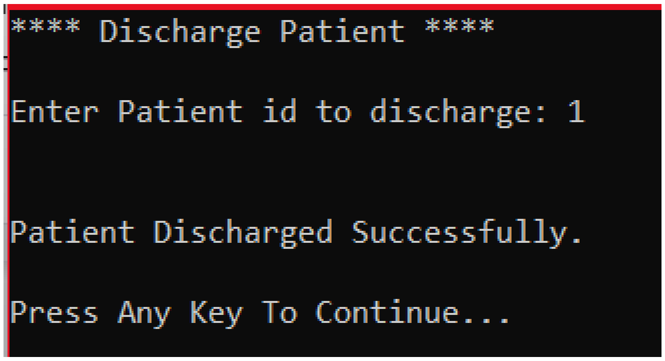
```
**** Patient List ****

Id         Patient Name                   Address                        Disease             Date
----------------------------------------------------------------------------------------------------------
2          PQR                            MNO                            CANCER              12/09/2022
4444       Raj Verdhan                    k.l..v.m.                      90                  05\03
1          RAJ                            BUI                            XXX                 112
```

INPUT 03:



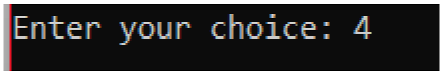
```
Enter your choice: 3
```

OUTPUT 03:



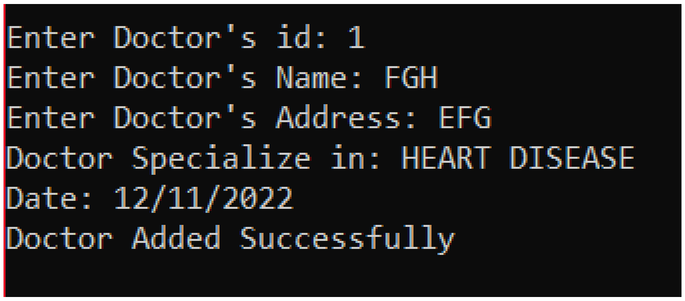
```
**** Discharge Patient ****

Enter Patient id to discharge: 1


Patient Discharged Successfully.

Press Any Key To Continue...
```

INPUT 04:



```
Enter your choice: 4
```

OUTPUT 04:



```
Enter Doctor's id: 1
Enter Doctor's Name: FGH
Enter Doctor's Address: EFG
Doctor Specialize in: HEART DISEASE
Date: 12/11/2022
Doctor Added Successfully
```

CONCLUSION :

Taking into account all the mentioned details, we can make the conclusion that the hospital management system is the inevitable part of the lifecycle of the modern medical institution. It automates numerous daily operations and enables smooth interactions of the users. Developing hospital system software is a great opportunity to create the distinct, efficient and fast delivering healthcare model. Implementation of hospital management system projects helps to store all kinds of records, provide coordination and user communication, implement policies, improve day-to-day operations, arrange the supply chain, manage financial and human resources, and market hospital services. This beneficial decision covers the needs of the patients, staff and hospital authorities and simplifies their interactions. It has become the usual approach to manage the hospital. Many clinics have already experienced its advantages and continue developing new hospital management system project modules.

Need to improve management in your hospital or plan to innovate healthcare with new software products? The hospital system software is a great solution for any medical institution. If you have some ideas and need assistance, you are welcome to contact Existek. We are an outsourcing company with great expertise in software development and always ready to find out the best approaches for your company. If you have any questions, feel free to contact us via the contact form or start the discussion in the website chat widget. Existek will be glad to provide professional consultation and find the solution for you.

Hospital management system is all about modernizing a hospital through use of technology. Computers help in it and take over the manual system for quick and easy functioning. This hospital management system is quite reliable and is proven on many stages. All the basic requirements of the hospital are provided in order to manage it perfectly and a large amount of data can also be stored . It gives many facilities like searching for the details of patients , billing facilities as well as the creation of test reports. So it;s a important system for modern days

REFERENCES:

WEBSITES:-

1. http://wikipedia.org
2. https://brainly.in
3. https://www.geeksforgeeks.org
4. https://www.programiz.com
5. https://github.com

BOOKS:-

6. C Programming Absolute Beginner's Guide,Greg Perry
7. C: The Complete Reference,Herbert Schildt
8. File Handling Using "C": Programming Concepts, Chandan Kumar

RESEARCH PAPERS:-

9. HOSPITAL MANAGEMENT SYSTEM, Grandhi poojitha, May - June 2020, ISSN: 0193-4120
10. A Study of advanced HOSPITAL MANAGEMENT SYSTEM, Dr. Pushphagaran, IOSR Journal of Dental and Medical Sciences (IOSR-JDMS) e-ISSN: 2279-0853, p-ISSN: 2279-0861.Volume 16, Issue 2 Ver. III (February. 2020), PP 127-134

FUTURE SCOPE :

1) Information about Patients is done by just writing the Patient's name, age and gender. Whenever the Patient comes up his information is stored freshly.
2) Bills are generated by recording the price for each facility provided to Patient on a separate sheet and at last they all are summed up.
3) Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.
4) Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.
5) Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and a lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can’t remember them at that time.

Significance :

The doctor/staff/nurse will benefit in the system. They will find it easier to transact about their records since searching in the system is faster than tracking in the record book or log book. The software will give them an easier time with their time log and they don’t have to worry about losing their time cards because it is not necessary.

Human Resources Personnel managing the timekeeping will not be the same again as they will experience relief. It would be fast and easy for them to handle transactions such as report making and monitoring time entries. Further determining or computing the salary-related manually will be eliminated that will lead to faster transactions.

APPENDIX :

13.1 Code of the solution:

```
#include<stdio.h>
#include<stdlib.h>


struct patient{
   int id;
   char patientName[50];
   char patientAddress[50];
   char disease[50];
   char date[12];
}p;


struct doctor{
   int id;
   char name[50];
   char address[50];
   char specialize[50];
   char date[12];
}d;


FILE *fp;


void admitPatient();
void patientList();
void dischargePatient();
void addDoctor();
void doctorList();


int main(){

   int ch;
   while(1){
      system("cls");
      printf("WELCOME...!\n");
```

```
    printf("\n");
    printf("**** Hospital Management System ****\n\n");
    printf("1.Admit Patient\n");
    printf("2.Patient List\n");
    printf("3.Discharge Patient\n");
    printf("4.Add Doctor\n");
    printf("5.Doctors List\n");
    printf("0.Exit\n\n");
    printf("Enter your choice: ");
    scanf("%d", &ch);


    switch(ch){

    case 0:
        exit(0);

    case 1:
        admitPatient(); //CALLING MODULE 1
        break;

    case 2:
        patientList();  //CALLING MODULE 2
        break;

    case 3:
        dischargePatient();  //CALLING MODULE 3
        break;

    case 4:
        addDoctor();  //CALLING MODULE 4
        break;

    case 5:
        doctorList();  //CALLING MODULE 5
        break;

    default:
        printf("Invalid Choice...!!\n\n");

    }
    printf("\n\nPress Any Key To Continue...");
    getch();
}
```

```
   return 0;
}


void admitPatient()  //MODULE 1
{

   fp = fopen("patient.txt", "ab");

   printf("Enter Patient id: ");
   scanf("%d", &p.id);

   printf("Enter Patient name: ");
   fflush(stdin);
   gets(p.patientName);

   printf("Enter Patient Address: ");
   fflush(stdin);
   gets(p.patientAddress);

   printf("Enter Patient Disease: ");
   fflush(stdin);
   gets(p.disease);

   printf("Enter Date of Admission: ");
   fflush(stdin);
   gets(p.date);

   printf("\nPatient Added Successfully....!!");

   fwrite(&p, sizeof(p), 1, fp);
   fclose(fp);
}


void patientList() //MODULE 2
{

   system("cls");
   printf("**** Patient List ****\n\n");
   printf("%-10s %-30s %-30s %-20s %s\n", "Id", "Patient Name", "Address", "Disease",
"Date");

printf("--------------------------------------------------------------------------------------------------------------\
n");
```

```
fp = fopen("patient.txt", "rb");
    while(fread(&p, sizeof(p), 1, fp) == 1){
        printf("%-10d %-30s %-30s %-20s %s\n", p.id, p.patientName, p.patientAddress, p.disease,
p.date);
    }

    fclose(fp);
}



void dischargePatient() //MODULE 3
{
                int id, f=0;
    system("cls");
    printf("**** Discharge Patient ****\n\n");
    printf("Enter Patient id to discharge: ");
    scanf("%d", &id);

    FILE *ft;

    fp = fopen("patient.txt", "rb");
    ft = fopen("temp.txt", "wb");


    while(fread(&p, sizeof(p), 1, fp) == 1){

        if(id == p.id){
            f=1;
        }else{
            fwrite(&p, sizeof(p), 1, ft);
        }
    }


    if(f==1){
        printf("\n\nPatient Discharged Successfully.");
    }else{
        printf("\n\nRecord Not Found !");
    }

    fclose(fp);
    fclose(ft);

    remove("patient.txt");
    rename("temp.txt", "patient.txt");

}



void addDoctor()  //MODULE 4
{
    int f=0;

    system("cls");
    printf("**** Add Doctor ****\n\n");

    fp = fopen("doctor.txt", "ab");

    printf("Enter Doctor's id: ");
    scanf("%d", &d.id);

    printf("Enter Doctor's Name: ");
    fflush(stdin);
    gets(d.name);

    printf("Enter Doctor's Address: ");
    fflush(stdin);
    gets(d.address);

    printf("Doctor Specialize in: ");
    fflush(stdin);
    gets(d.specialize);

    printf("Date: ");
    fflush(stdin);
    gets(d.date);

    printf("Doctor Added Successfully\n\n");

    fwrite(&d, sizeof(d), 1, fp);
    fclose(fp);
}



void doctorList()  //MODULE 5
{
    system("cls");
```

```
   printf("**** Doctor List ****\n\n");

   printf("%-10s %-30s %-30s %-30s %s\n", "id", "Name", "Address", "Specialize","Date");

printf("-------------------------------------------------------------------------------------------------------
-------\n");

   fp = fopen("doctor.txt", "rb");
   while(fread(&d, sizeof(d), 1, fp) == 1){
      printf("%-10d %-30s %-30s %-30s %s\n", d.id, d.name, d.address, d.specialize, d.date);
   }

   fclose(fp);
}
```