

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**Proposal**

**On**

**Hospital Management System**

**Submitted by:**

Shaswat Sharma (THA080BCT040)

Raman Shrestha (THA080BCT031)

Biraj Adhikari (THA080BCT013)

Janak Chhatkuli (THA080BCT018)

**Submitted to:**

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

March, 2024

# ABSTRACT

The world is digitized and everything we do is aided by the use of technology and the internet. Even though the use of computers, softwares and the internet is prevalent in most domains and industries, it is yet to be optimized to its maximum potential in the healthcare industry. The hospitals in the city areas are well equipped with computers and management system softwares however the ones in villages and rural places still operate by the use of huge registers and ledgers for storing and retrieving the patient data. Our aim from this project is to streamline that process and make it feasible for the use of a software in managing the hospitals and healthcare offices of the rural areas. Using file handling features in C programming language, we will be able to prepare the Hospital Management System (HMS). This management system can be operated even by the ones with little knowledge of computers and programming. The use of this software will replace the need of bulky ledgers and mismatching of a person's diagnosis with that of another person with the same name. The project aims to reduce the errors that naturally occur with unplanned management and makes the management system accessible to every health institution. The HMS will be able to add a patient's detail, search and view it, edit the details and also delete them which will make it convenient as a replacement for the normal ledgers and papers.

*Keywords:*

*HMS (Hospital Management System)*

*EHR (Electronic Health Records)*

*CIS (Clinical Information System)*

*RFID (Radio Frequency Identification)*

# TABLE OF CONTENTS

[**ABSTRACT 1**](#_heading=h.gjdgxs)

[**TABLE OF CONTENTS 2**](#_heading=h.30j0zll)

[**LIST OF FIGURES 4**](#_heading=h.3znysh7)

[**LIST OF TABLES 5**](#_heading=h.qsh70q)

[**LIST OF ABBREVIATIONS 6**](#_heading=h.3l18frh)

[**1. INTRODUCTION 1**](#_heading=h.4k668n3)

[1.1 Background 1](#_heading=h.2zbgiuw)

[1.2 Motivation 1](#_heading=h.1egqt2p)

[1.3 Objectives 2](#_heading=h.3ygebqi)

[1.4 Scope and Applications 2](#_heading=h.2dlolyb)

[**2. LITERATURE REVIEW 2**](#_heading=h.sqyw64)

[**3. PROPOSED SYSTEM ARCHITECTURE 5**](#_heading=h.3cqmetx)

[3.1 System Architecture 5](#_heading=h.1rvwp1q)

[3.2 Parts of the Program 5](#_heading=h.4bvk7pj)

[3.2.1 Entry Choices 5](#_heading=h.2r0uhxc)

[3.2.1.1 Adding Information 5](#_heading=h.1664s55)

[3.2.1.2 Viewing Information 7](#_heading=h.3q5sasy)

[3.2.1.3 Searching Information 8](#_heading=h.25b2l0r)

[3.2.1.4 Editing Information 9](#_heading=h.kgcv8k)

[3.2.1.5 Deleting Information 10](#_heading=h.34g0dwd)

[3.2.1.6 Exit 11](#_heading=h.1jlao46)

[3.3 Tools and IDE 11](#_heading=h.43ky6rz)

[4. METHODOLOGY 11](#_heading=h.2iq8gzs)

[4.1 Header Files 11](#_heading=h.xvir7l)

[4.2 Formatted and Unformatted Input / Output functions 12](#_heading=h.3hv69ve)

[4.2.1 Formatted Input/Output functions 12](#_heading=h.1x0gk37)

[4.2.2 Unformatted Input/Output functions 13](#_heading=h.4h042r0)

[**4.3 Control Statements 14**](#_heading=)

[4.4 Function 14](#_heading=h.1baon6m)

[4.5 Array 15](#_heading=h.3vac5uf)

[4.6 Structure 16](#_heading=h.2afmg28)

[4.7 File Handling 17](#_heading=h.pkwqa1)

[**5. TIME ESTIMATION 18**](#_heading=h.39kk8xu)

[**6. FEASIBILITY ANALYSIS 19**](#_heading=h.48pi1tg)

[6.1 Technical Feasibility 19](#_heading=h.2nusc19)

[6.2 Operational Feasibility 19](#_heading=h.1302m92)

[6.3 Economic Feasibility 19](#_heading=h.3mzq4wv)

[6.4 Hardware Requirements 19](#_heading=h.2250f4o)

[6.5 Software Requirements 20](#_heading=h.haapch)

[**REFERENCES 21**](#_heading=)

# 

# LIST OF FIGURES

Figure 3.1: System Architecture of the proposed system

Figure 3.2.1.1: Adding Information

Figure 3.2.1.2: Viewing Information

Figure 3.2.1.3: Searching Information

Figure 3.2.1.4: Editing Information

Figure 3.2.1.5: Deleting Information

# LIST OF TABLES

Table 4.2.1: Datatypes in C

Table 5.1: Time Estimation Gantt Chart

# LIST OF ABBREVIATIONS

HMS Hospital Management System

EHR Electronic Health Records

CIS Clinical Information System

RFID Radio Frequency Identification

IoT Internet of Things

IT Information Technology

IDE Integrated Development Environment

AI Artificial Intelligence

# 1. INTRODUCTION

## 1.1 Background

The Hospital Management System will be created in C programming language. The project mostly focuses on using the basic tools of C to prepare a cohesive program. The use of data structures and similar features of C provides the program with necessary components for the program to run. The program is created in order to aid the gap in technology in the medical field and replace traditional methods with modern approach. Using a program to add the information about a patient, their personal details and their condition not only ensures that the added data is safe, it also makes sure that the said information could be retrieved any time and this will remove the hassle of finding the data through traditional means. Not only that, using this system to edit the data will come to the rescue when there are certain types of errors in the input data. The program is also efficient in the sense that the uploaded data will be really easy to delete, making it possible for the program to operate even with the devices with low storage.

## 1.2 Motivation

Human life is very valuable and anything that creates or tends to create a negative impact on human health is pretty detrimental. While still being aware of this fact, knowingly or unknowingly the health of people have been compromised due to sheer negligence many have been misdiagnosed due to this same reason. The traditional approach of managing a patient's data used to work when the number of people were very low, however since the population has increased and so has the number of ill people at any given time, the cases of misdiagnosis has skyrocketed and we've identified that replacing the traditional ways with the use of computer and a software will have a very positive impact in this aspect. With the same noble intention of reducing the number of misdiagnosis of ill persons due to carelessness and hopefully having a role in ensuring an overall better healthcare system, we intend to do this project.

## 1.3 Objectives

The project aims to fulfill the following objectives:

1. To reform and replace the traditional way of managing a healthcare organization with the use of a computer program
2. To reduce the number of misdiagnosis due to carelessness and negligence
3. To make us as programmers familiar with file handling, functions and various other elements of C programming language

## 1.4 Scope and Applications

The program is able to store the data of a patient in a well organized manner. The list of patients can also be viewed as well as the details of the patient. The program will also have a search feature which will help to find the required data through the countless numbers of data stored. Similarly, the data can be edited as well. The patients data can also be deleted when it is not required. Thus the program will be able to store a huge number of data of the patients, make changes to the stored data incase of an error or mishandling. It will also be able to provide the exact patients data when searched and viewed and finally will also be able to discard and delete the useless data entries. It will be applicable in healthcare field.

# 2. LITERATURE REVIEW

A Hospital Management System (HMS) plays a crucial role in modern healthcare facilities, encompassing various administrative and operational functions to optimize patient care, resource management, and overall efficiency. This literature review aims to explore existing research and developments in hospital management systems to provide a comprehensive understanding of the current state of the art, identifying key challenges, and highlighting opportunities for improvement.

1. Electronic Health Records(EHR) and Clinical Information System (CIS)

EHR and CIS have been a focal point of research and development in the healthcare domain. Studies by Bates and others (2014) and Wright and others (2017)[1] have demonstrated the significant impact of EHR systems on improving patient safety, care coordination, and clinical decision-making. Furthermore, the integration of clinical decision support systems within EHR has shown promising results in enhancing diagnostic accuracy and treatment outcomes (Kawamoto, 2005)[2].

1. Patient Management and Workflow Optimization

Efficient patient management and workflow optimization are vital components of hospital operations. Research by Hingorani (2018) and Li and others (2019)[3] has emphasized the importance of implementing advanced scheduling algorithms and queue management systems to minimize patient waiting times, streamline resource allocation, and enhance overall patient satisfaction. In addition , the utilization of mobile health applications for patient management and self-management has gained traction, offering opportunities to improve patient-provider communication and adherence to treatment plans (Bashi and others, 2013).

1. Resource Allocation and Inventory Management

Effective resource allocation and inventory management are critical for ensuring the availability of medical supplies, equipment, and pharmaceuticals. Studies by Van Vlaenderen and others (2016) and Li and others (2018)[4] have highlighted the potential data-driven approaches, such as predictive analytics and machine learning, in optimizing inventory levels, reducing wastage, and forecasting demand for medical resources. Furthermore, the implementation of RFID and IoT enabled systems has shown promise in real-time asset tracking and inventory control within healthcare facilities (Kumar and others, 2019).

Challenges and Future Development:

Despite the advancements in HMS, challenges persist in areas such as interoperability, data security and user adoption. Interoperability issues between different healthcare IT systems continue to hinder seamless data exchange and care coordination (Jha and others, 2014). Addressing these challenges requires a concerted effort to develop standardized data exchange protocols and interoperable platforms. Additionally, ensuring robust data security and privacy safeguards is imperative to maintain patient confidentiality and compliance research and innovation in the areas of AI, blockchain, and telemedicine are poised to revolutionize hospital management systems, offering opportunities to automate routine tasks, enhance data integrity, and extend healthcare services to remote populations (Gordon and others, 2020; Kuo and others, 2017)[5].

The literature reviewed underscores the multifaceted nature of hospital management systems and their pivotal role in modern healthcare delivery. By leveraging advanced technologies and best practices, hospital management systems can drive improvements in patient care, operational efficiency, and resource utilization. However, addressing the existing challenges and embracing emerging trends will be essential to realize the full potential of these systems and ensure their seamless integration into healthcare ecosystems.

# 3. PROPOSED SYSTEM ARCHITECTURE

The different blocks of system architecture are explained below:

## 3.1 System Architecture

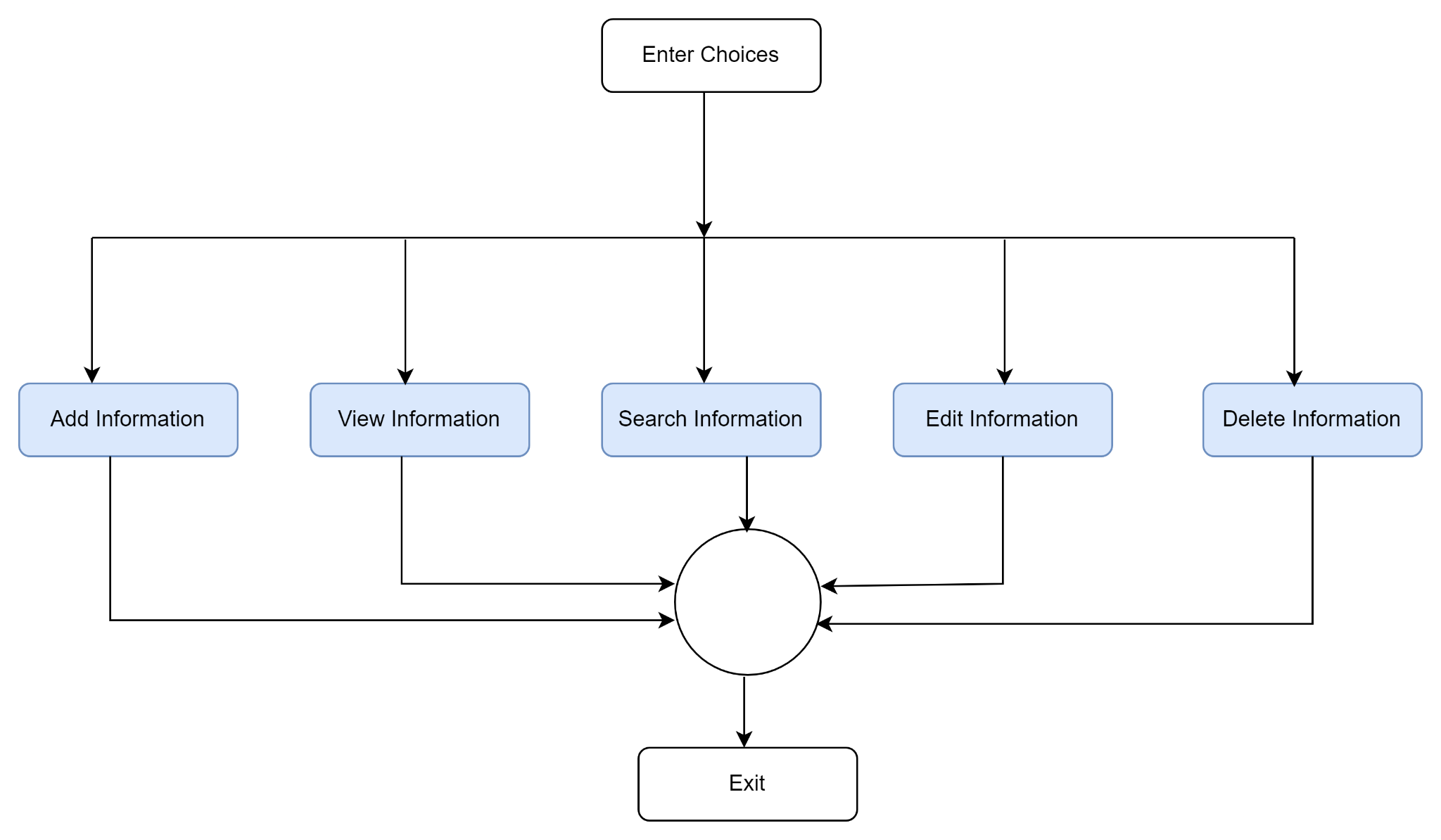


Figure 3.1 : System Architecture of the Proposed System

## 3.2 Parts of the Program

### 3.2.1 Entry Choices

The first main interaction of the user on the console will be the selection of the instruction that has to be operated. First the user has to add new data of the patients, as the file will be empty at the beginning. Then other operations like viewing the information, searching the information, editing the information, deleting the data will be possible.

#### 3.2.1.1 Adding Information

Now the user has to add the number of data that has to be entered. Then information has to be entered by the user. In the first iteration the information of the first patient has to be entered and it continues to ‘n’ number of patients and stored in respective variables.

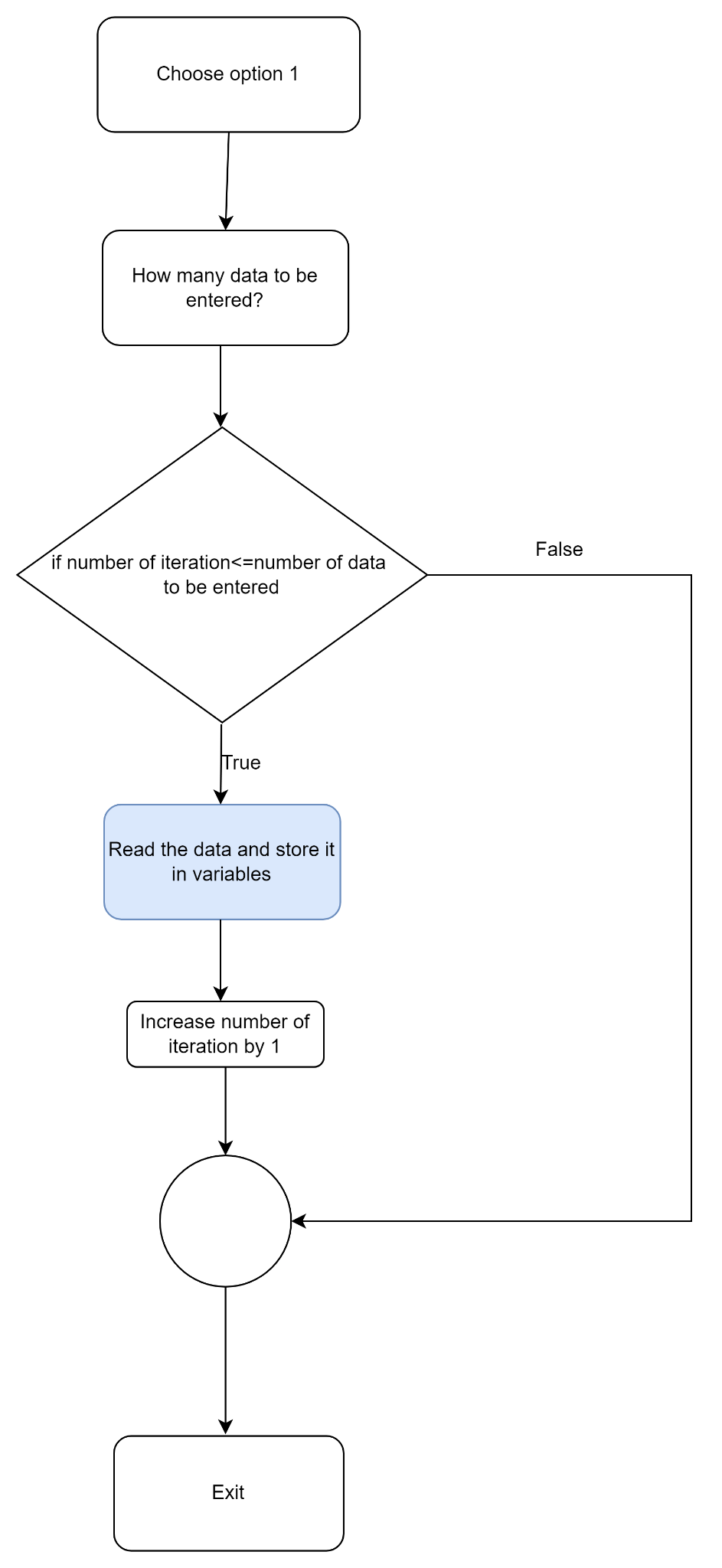


Figure 3.2.1.1: Adding Information

#### 3.2.1.2 Viewing Information

After adding the data of the patients, users can easily see every patient's information by just selecting the second option. But in order to view the patients information it is necessary to have the data of the patient existing in the file.

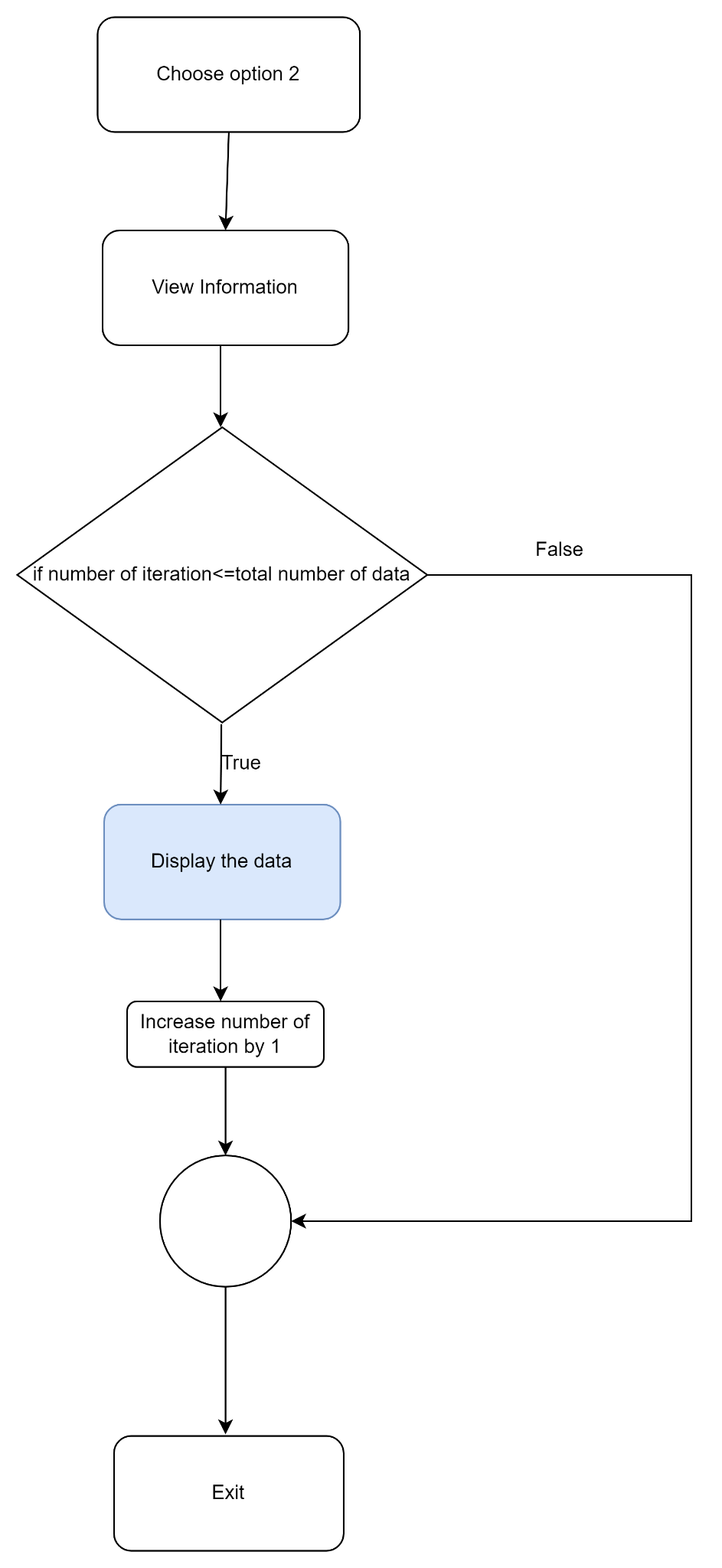


Figure 3.2.1.2: Viewing Information

#### 3.2.1.3 Searching Information

Users can search the required information of the patient by using the variable name as a primary key. Before that, the patient whose information has to be displayed is selected by using serial number assigned to the patient through a loop.

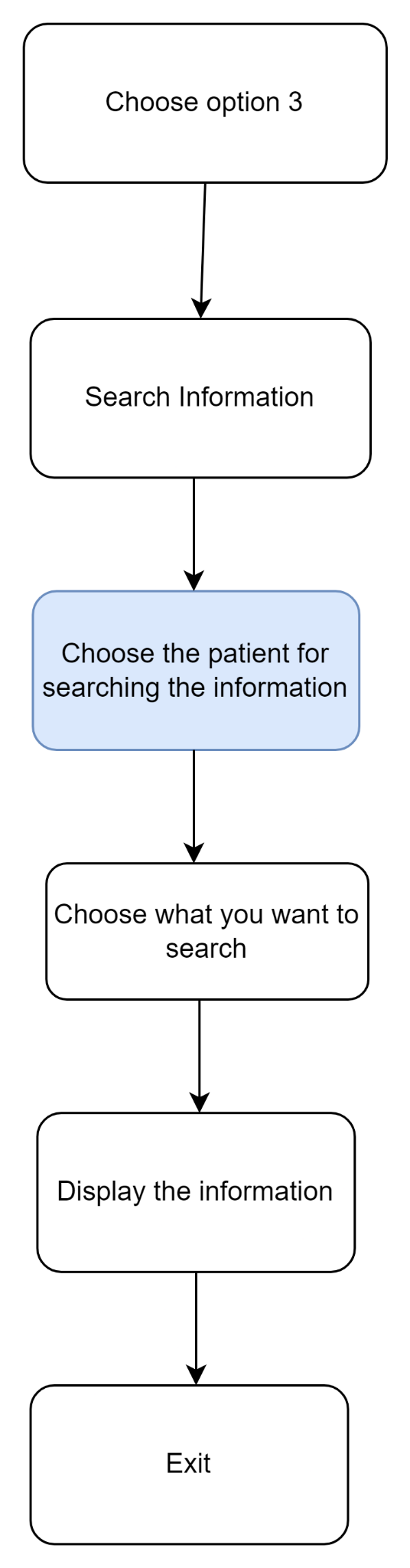


Figure 3.2.1.3: Searching Information

#### 3.2.1.4 Editing Information

In case a user has to update the information of the patient, there will be a user defined library function for editing the selected information of the selected patient.

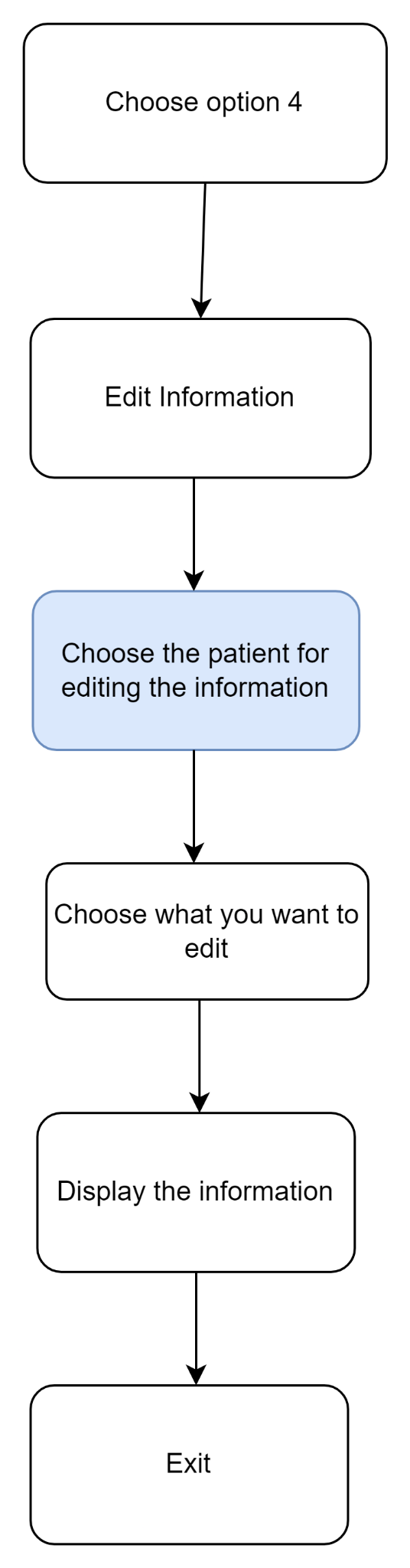


Figure 3.2.1.4: Editing Information

#### 3.2.1.5 Deleting Information

For the patients whose information has to be deleted, a user defined function is defined, where the user can either completely delete the information or selectively delete the information of the user.

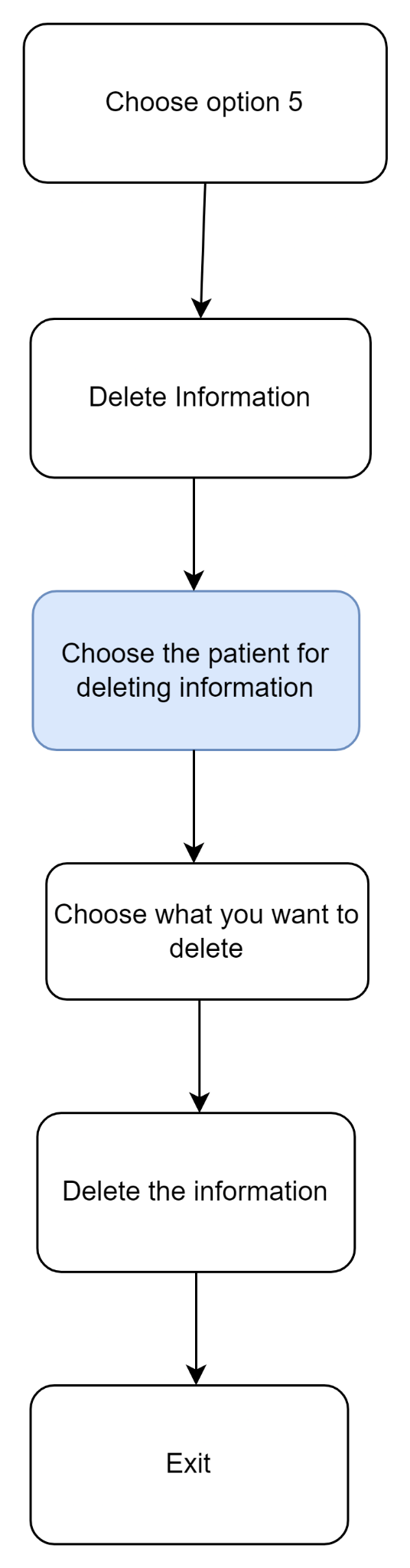


Figure 3.2.1.5: Deleting Information

#### 3.2.1.6 Exit

It exits the console.

## 3.3 Tools and IDE

Code Blocks is used to write the code.

# 4. METHODOLOGY

Hospital Management system is made with C language. C is a high level programming language which can be used for system programming. C is a robust machine independent, general purpose, concise programming language. It features an economy of expressions, modern control flow and data structure, and a rich set of operators.

The methodology used in this program are:

## 4.1 Header Files

In C language, header files contain a set of predefined standard library functions. The .h is the extension of the header files in C and we include the header file in the program by using preprocessor directive called ‘#include’.

Syntax:

#include<header\_file\_name.h>

There are two types of header file in C. They are:

1. Standard header files

These header files are stored in the default directory of the compiler and can be easily accessed using the preprocessor directive.

Some of the standard header files in C are:

1. stdio.h

It is used to perform input and output operations using functions like scanf(), printf() etc.

1. string.h

It is used to perform various functionalities related to string manipulation like strlen(), strcmp(), strcpy(),size(), etc.

1. math.h

It is used to perform mathematical operations like sqrt(), pow(), log2(), etc.

1. User-defined header files

These header files are user defined header files which contain functions for performing a certain task. To include a user defined header file it can be done as:

#include”header\_file.h”

## 4.2 Formatted and Unformatted Input / Output functions

### 4.2.1 Formatted Input/Output functions

Formatted Input/Output functions allow input or output data to format according to the user requirement. While displaying a certain data on the screen, we can specify the number of digits after the decimal point, number of spaces before the data item, the position where the output is to be displayed using a formatted function.

Some of the formatted input/output functions are: printf(), scanf(), etc.

Format Specifiers are used to tell the compiler about the type of data to be printed or scanned in input and output operations. In formatted I/O we use various format specifiers which helps to read and display the data of desired data type.

The chart shows the various datatypes in C.

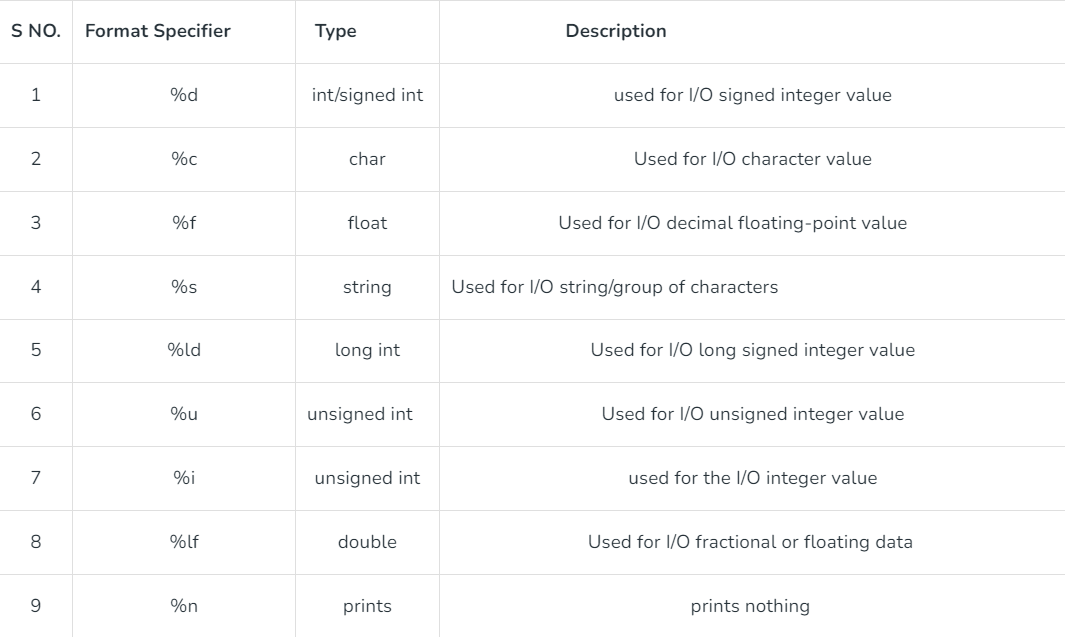


Table 4.2.1: Datatypes in C

### 4.2.2 Unformatted Input/Output functions

Unformatted Input/Output does not allow the user to read and display data in the desired format. These functions are used only for the character data type of character array/string and cannot be used for any other datatype. These functions are used to read single input form the user at the console and it allows to display the value at the console.

Some of the unformatted Input/Output functions are:

1. Unformatted Input functions
2. getch()
3. getche()
4. getchar()
5. gets()

2. Unformatted Output functions

1. putch()
2. putchar()
3. puts()

## 4.3 Control Statements

The control statements used in C language helps the user to specify a program's control flow. In simpler words, the control statement helps the user specify the order of execution of the instruction present in the program. Control statements make it possible for the program to make certain decisions, perform various tasks repeatedly or even jump from one section of the code to different sections. Its types are:

1. if statements
2. Simple if
3. Nested if
4. if else
5. Nested if else
6. if else ladder

2. Switch statement

3. Goto statements

4. Loop statements

1. For loop
2. While loop
3. Do while loop

## 4.4 Function

A function in C is a set of statements that when called performs a specific task. It is the basic building block of a C program that provides modularity and code reusability. The programming statements of a function are enclosed within {} braces, having certain meanings and performing certain operations.

Syntax of Function in C

The syntax of function can be divided into 3 aspects:

1. Function Declaration

In a function declaration, we must provide the function name, its return type, and the type of its parameters. Function declaration tells the compiler that there is a function with the given name defined somewhere in the program.

Syntax: return\_type name\_of\_function(parameters);

1. Function Definition

The function definition consists of actual statements which are executed when the function is called.

Syntax: return\_type func\_name(parameters arguments)

{

// statements to be executed

return 0;

}

1. Function Call

A function call is a statement that instructs the compiler to execute the function. We use the function name and the parameters in the function call. Function is necessary, if not called the function statements will not be executed.

## 4.5 Array

Array is the collection of similar data items that can be represented by a single variable name with different subscripts or index. The individual data item in the array is called an element. Array uses subscripted variables and makes the representation of a large number of homogenous values possible.

The arrays whose elements are specified by one subscript are called one dimension arrays. Analogous arrays whose elements are specified by two or three subscripts are called two dimensional and three dimension arrays.

Declaration of Array:

For 1-D array:

datatype identifier[size of array];

For 2-D array:

datatype identifier[number\_of\_row][number\_of\_column];

## 4.6 Structure

A structure is a collection of logically related data items grouped together under a single name called structure tag. In structure teh individual data type may differ, so we can regard structure as a heterogenous user defined data type. The data items enclosed within a structure are known as members.

Syntax of structure:

Struct struct\_name{

data\_type mem\_1;

data\_type mem\_2;

.

.

.

};

After the structure has been specified the structure variable can be declared as :

struct struct\_name var1,var2,.........var\_n;

## 4.7 File Handling

File Handling in C is the process in which we create, open, read, and close operations on a file. C language provides different functions such as fopen(), fwrite(), fread(), fseek(), fprintf(), etc. to perform input, output, and many different C file operations in our program.

C File operations

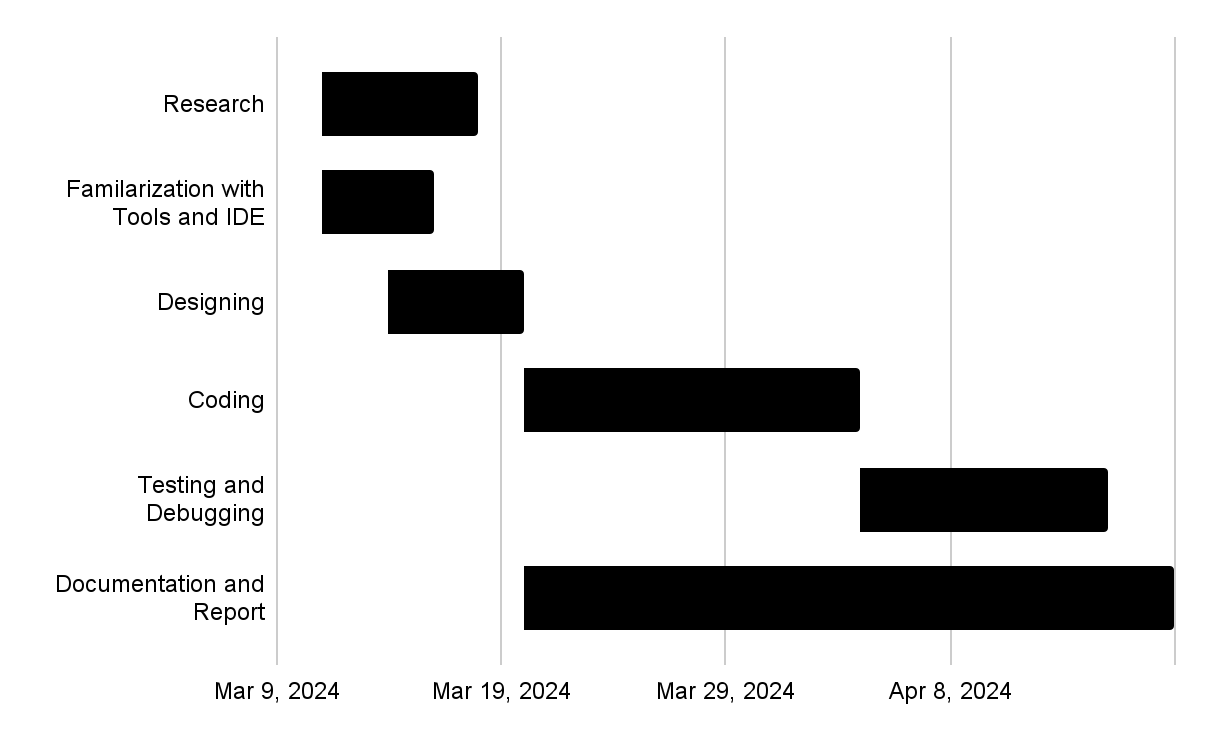
C file operations refers to the different possible operations that we can perform on a file in C such as:

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

These contents are densely used while making the Hospital Management system.

# 5. TIME ESTIMATION

Table 5.1: Time Estimation Gantt Chart



# 6. FEASIBILITY ANALYSIS

The entire purpose of feasibility analysis is to measure how beneficial is the development of this HMS. It is done to test the Technical, Operational and Economical as well as Time feasibility.

## 6.1 Technical Feasibility

In this feasibility, it is decided whether the system can be developed and implemented using the existing technology or not after studying technical factors like hardware and software. Things that were analyzed and ensured after conducting technical feasibility are as follows :

i. Using the current technology, it is possible to develop the proposed system.

ii. It can be made according to the system architecture.

iii. We do have the possession of necessary technical manpower for its operation.

## 6.2 Operational Feasibility

It is done to make sure the developed system will be able to perform designated tasks with the existing environment and the existing human resources. Some of the important issues raised to test the operational feasibility of this project include the following :

i. Anyone with basic practical knowledge of using a computer will be able to use this system.

ii. The system will work properly after its development and implementation.

iii. There is adequate support for the management from the users.

## 6.3 Economic Feasibility

Economic feasibility is done to determine how beneficial the system will be with regards to time, money and other resources required. Generally, financial benefits to the development team exceeds the total cost required to build the system if the development and implantation is successful. As for this HMS, since it is based on C, no development costs is to be allocated.

## 6.4 Hardware Requirements

Even the most basic computer with less storage will also be able to access this program as it is intended to a the hospitals and healthcare organizations of the rural areas where the accessibility of the newest generation PCs are less, so the minimum hardware requirements that will be able to run this program are:

i. Intel i3 2.8 Ghz Processor and above

ii. RAM 4GB and above

iii. HDD 5GB Hard disc and above

iv. An input device (preferably a keyboard)

v. A monitor

## 6.5 Software Requirements

Since the project is created in C, any version of the operating system above Windows 7 will be able to run the program however it is preferred if the operating system is the latest Windows, Linux or macOS.

# REFERENCES

1. Bates, D. W., & Gawande, A. A. (2014). Improving safety with information technology. New England Journal of Medicine, 370(25), 2418-2420.

2. Wright, A., Hickman, T. T., McEvoy, D., Kim, J., & Goldberg, H. J. (2017). Analysis of clinical decision support system malfunctions: a case series and survey. Journal of the American Medical Informatics Association, 24(1), 81-87.

3. Kawamoto, K., Houlihan, C. A., Balas, E. A., & Lobach, D. F. (2005). Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. British Medical Journal, 330(7494), 765.

4. Li, L., Fu, F., Wu, J., Wang, J., & Yan, L. (2019). A survey on healthcare management systems: Issues, challenges, and opportunities. Journal of Industrial Information Integration, 15, 21-38.

5. Li, L., Zheng, L., Wu, J., & Wang, J. (2018). A survey on the application of data mining techniques in healthcare. Computer Methods and Programs in Biomedicine, 104-131.