TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS

PROJECT 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 Institute of Engineering, Tribhuvan University
Kathmandu, Nepal

April 2024

Hospital Management System

COPYRIGHT

ACKNOWLEDGMENT

We were only able to complete this project with a great deal of additional research from various personnels as well as various resources available on the internet. We took help from some experienced people and even from web-pages. This project has been prepared as the project assignment assigned by Mr. Saroj Shakya. This project has definitely helped us gain practical knowledge about System Development and C Programming Language as a whole. We would like to thank the Institute of Engineering, Thapathali Campus for allowing us to broaden our knowledge and concepts on programming, specifically C language by providing us the opportunity to work on this project.

We are very glad to express our deepest sense of gratitude and sincere thanks to our respected supervisor Er. Saroj Shakya, Er. Rajad Shakya and Er. Prajwal Pakka who all have guided us as a mentor and provided us, their valuable supervision, guidance, encouragement, and support for completing this work. Their inspiration and guidance have made this project successful. We were also aided by our colleagues and we would like to extend our warmest regards to everyone who helped us in completing this project and getting it to the level we were satisfied with.

Shaswat Sharma (THA080BCT040)
Raman Shrestha (THA080BCT031)
Biraj Adhikari (THA080BCT013)
Janak Chhatkuli (THA080BCT018)

ABSTRACT

The world is digitised 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 optimised 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 the C programming language, we are able to prepare the Hospital Management System (HMS). This management system can be operated even by the ones with little to no 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 is able to add a patient's details, 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)
CIS (Clinical Information System)
EHR (Electronic Health Records)
IDE *(Integrated Development Environment)*
RFID (Radio Frequency Identification)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES:

1. INTRODUCTION

The Hospital Management System has been 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 approaches. 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.

1.1 Background

We as the students of Computer Engineering are taught about the basic concept of programming in high level language as a way of improving our skills in logical reasoning, application of technical knowledge, and quantitative problem solving. Thus, we were assigned to prepare a project in C language in order to have a basic practical knowledge about C-programming. Knowing the current demand in the medical field and the gap of technology and the whole healthcare industry in general, we decided to develop a management system for the better functioning of a small scale health institution.

We firstly discussed the program flow, necessary logic and user interface. Then we modulated our program into a number of segments. We then divided our workload among the four of us and started working on our individual parts. We allocated about four weeks for the completion of this project. We consulted course and non- course books as a reference and guide in our project preparation.

When it comes to implementing a Hospital Management System (HMS) in C, the initial step involves defining the system's fundamental functionalities and rules. This includes outlining the core features of the HMS, such as patient management, record keeping, and administrative tasks. Following this, we designed data structures to represent key components of the HMS, such as patient records, medical staff details, and administrative information. These structures facilitate the efficient storage and

manipulation of data within the program. Subsequently, algorithms were developed to manage various aspects of the HMS's functionality. This includes procedures for adding, viewing, searching, editing, and deleting patient records. Additionally, algorithms for handling administrative tasks, generating reports, and managing system resources may also be implemented for larger scale implementation.

For an HMS, the primary focus revolves around maintaining accurate patient records, ensuring data security, and streamlining administrative processes. Thus, the implementation involves creating functions to handle tasks such as patient registration, appointment scheduling, medical history retrieval, and billing.

Overall, developing an HMS in C demands a comprehensive understanding of programming concepts, data structures, file handling operations and algorithms. It presents a rewarding challenge for developers seeking to enhance their skills in software development while delivering an efficient and user-friendly solution for managing healthcare facilities.

1.2 Problem Definition

The challenge is to develop a Hospital Management System (HMS) in C that efficiently handles patient data management, administrative tasks, and regulatory compliance in healthcare facilities. This entails defining core functionalities such as patient registration, record keeping, appointment scheduling, and billing while ensuring robust data security measures and regulatory adherence. The system must offer scalability, flexibility, and cross-platform compatibility to accommodate evolving healthcare needs and facilitate seamless integration with existing infrastructure. Ultimately, the goal is to deliver a user-friendly HMS solution that optimises hospital operations, enhances patient care, and ensures compliance with healthcare standards and regulations.

1.3 Objectives

- To reform and replace the traditional way of managing a healthcare organisation with the use of a computer program.
- To reduce the number of misdiagnosis due to carelessness and negligence
- Practice error handling and validation techniques to ensure code reliability.
- Develop testing and debugging proficiency for an error-free code.
- Improve documentation skills for clear communication of project details.

●To make the use as programmers familiar with file handling, functions and various other elements of programming language process.

1.4 Features:

We have given our program essential features to make a proper system.

● The program utilises a console-based interface, employing screen clearing and cursor positioning techniques to maintain clarity and organisation.

● Patient data and administrative tasks are presented in an intuitive manner, enhancing user experience and facilitating smooth navigation.

● Error handling mechanisms provide informative feedback, enabling quick resolution of issues and ensuring a seamless user experience throughout.

2. REVIEW OF RELATED LITERATURES

A Hospital Management System (HMS) plays a crucial role in modern healthcare facilities, encompassing various administrative and operational functions to optimise 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.

2.1 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].

2.2 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 emphasised the

importance of implementing advanced scheduling algorithms and queue management systems to minimise patient waiting times, streamline resource allocation, and enhance overall patient satisfaction. In addition, the utilisation 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).

2.3 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 optimising
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).

2.4 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 standardised 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 revolutionise 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].

2.5 Data Analysis

From 2018 to 2026, the smart hospital market's annual value surged, driven by heightened adoption of Hospital Management Systems (HMS) and cutting-edge healthcare technologies. Early stages saw modest figures, but increasing demand for efficient patient care, operational streamlining, and better outcomes fueled consistent growth. By 2026, HMS implementation, bolstered by features like EHR and AI, revolutionised healthcare management, propelled further by government initiatives and pandemic-driven digital healthcare imperatives.

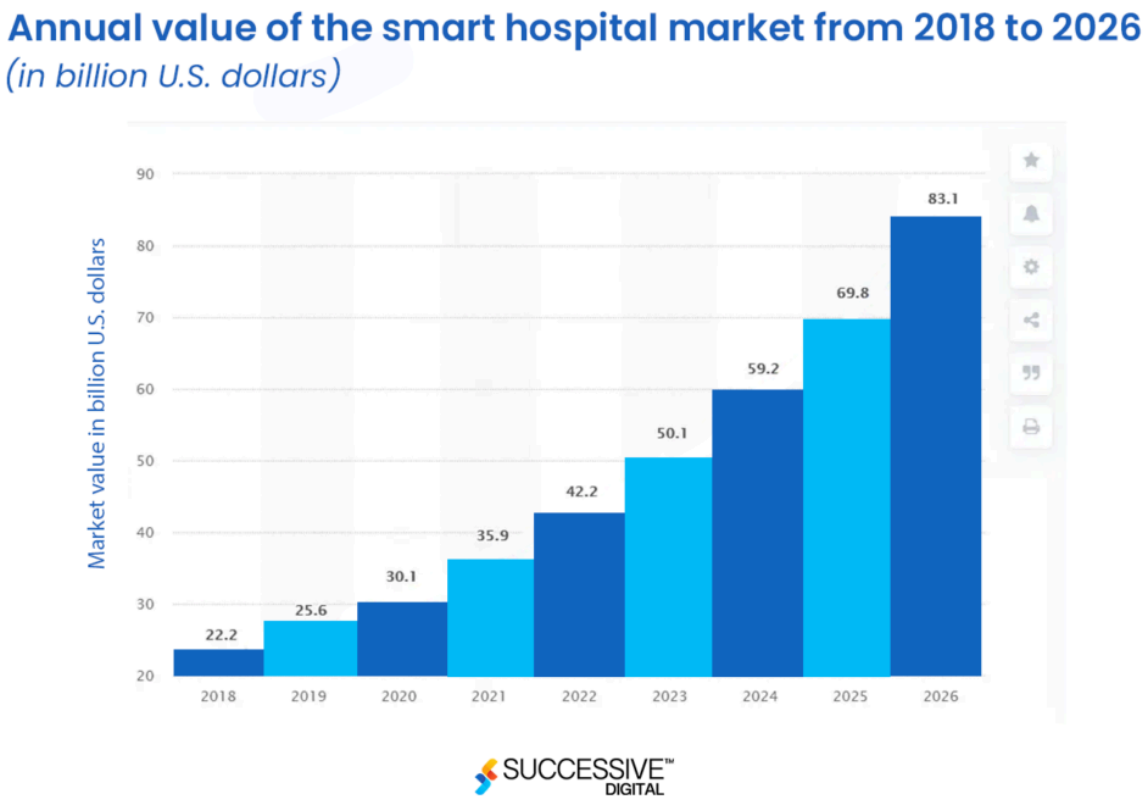

Table 1 – Graph of the implementation of HMS

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 utilisation. However, addressing the existing challenges and embracing

emerging trends will be essential to realise the full potential of these systems and ensure their seamless integration into healthcare ecosystems.

3. METHODOLOGY AND SYSTEM DESCRIPTION

3.1 Methodology

Our HMS system's methodology entails the systematic development of a comprehensive hospital management solution using the C programming language. The approach is segmented into distinct phases, each contributing to the system's functionality and effectiveness.

3.1.1 System Initialization and Setup

The system initialises by defining essential data structures for patient information and hospital records. Initialization of necessary variables, including patient attributes and system configurations, lays the groundwork for seamless operation. Additionally, setup of the console-based interface ensures clarity and organisation for efficient user interaction..

3.1.2 Staff Interaction and Input Handling

Staff interact with the system by providing input for various operations, such as registration, appointment scheduling, and data updates. Input validation mechanisms ensure the integrity of patient data, preventing erroneous entries and maintaining accuracy. The system processes patient input, updating the database and displaying relevant information in response to user actions.

3.1.3 Data Management and Record Keeping

Implementation of algorithms for efficient data management, including storage, retrieval, and manipulation of patient records, forms a cornerstone of the system. Utilisation of file

handling techniques facilitates secure storage of patient data and seamless access to information. Integration of error handling mechanisms addresses data discrepancies, ensuring the reliability of stored records.

3.1.4 Administrative Functions and System Operations

Provision of administrative functionalities for hospital staff enables tasks such as patient management, inventory control, and financial transactions. Development of algorithms to streamline administrative processes enhances operational efficiency and productivity. Implementation of user-friendly interfaces and menu options facilitates intuitive navigation and task execution.

3.1.5 System Maintenance and Optimization

Incorporation of features for system maintenance, including data backup, system updates, and performance monitoring, ensures long-term reliability. Integration of optimization techniques enhances system responsiveness, scalability, and resource utilisation. Continuous refinement and improvement of system functionalities based on user feedback and evolving healthcare requirements drive ongoing optimization efforts.

3.1.6 User Training and Support

Provision of training materials and resources facilitates user adoption and proficiency with the HMS system. Implementation of user support mechanisms, including help documentation, online tutorials, and dedicated support channels, enhances user experience. Ongoing assistance and guidance to users address queries, troubleshoot issues, and maximise the benefits of the HMS system.

3.2 System Description

The Hospital Management System (HMS) project in C initialises key variables and structures to facilitate efficient management of hospital operations. It employs a console-based interface to interact with users and manage patient information. The system allows for various functionalities, including patient registration, viewing information, searching records, editing

data, and deleting entries. Input validation mechanisms ensure the accuracy and integrity of data, while error handling procedures provide informative feedback to users. The system prioritises user convenience and clarity, aiming to streamline hospital administrative tasks and enhance overall efficiency in healthcare management.

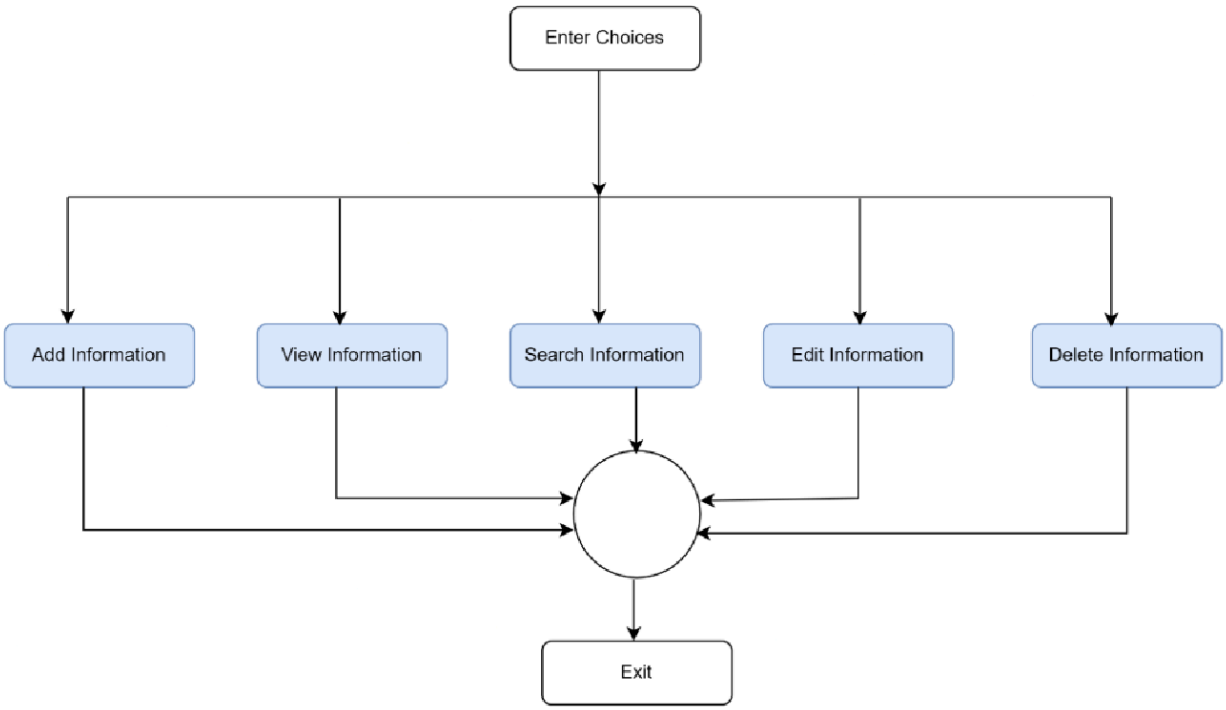

Fig 1 System Flow of HMS

3.2.1 System Initialization and Setup:

Implementation:

The HMS system initialises its core components by defining patient structures and setting up necessary variables. It clears the console screen for a clean interface and positions the cursor appropriately.

Code Implementation:

In the code, patient structures are defined to store patient information, and variables are

initialised to manage data. Console manipulation functions like `system("cls")` and `gotoxy()` ensure a clear and organised interface, enhancing user experience.

3.2.2 Patient Registration and Input Handling:

Implementation:

Patients interact with the system by providing their details through the console interface. Input validation ensures accurate data entry, preventing errors and ensuring the integrity of the patient database.

Code Implementation:

Functions such as add() prompt patients to input their information, including name, age, and address. Input validation within these functions ensures that valid data is entered, enhancing the reliability of the patient registration process.

3.2.3 Record Management and Storage:

Implementation:

Upon successful input, patient records are stored in a file for efficient management and retrieval. This ensures that patient data is securely stored and can be accessed whenever needed for reference or analysis.

Code Implementation:

Functions like `save()` handle the storage of patient records in a file named "patients.txt". The data is formatted appropriately and appended to the file for future access and management, contributing to the organised storage of patient information

3.2.4 Record Retrieval and Viewing:

Implementation:

Patients and healthcare providers can easily retrieve and view patient records from the system. This facilitates quick access to patient information for medical consultations,

treatment planning, and administrative purposes

Code Implementation:
Functions like `view()` enable users to view patient records stored in the file "patients.txt". The system reads the file and displays the patient information in a structured format, enhancing accessibility and usability for healthcare professionals.

3.2.5 Record Editing and Updating:

Implementation:
Healthcare providers can edit and update patient records to ensure accuracy and completeness. This functionality allows for corrections, updates, or additions to patient information as necessary.

Code Implementation:
Functions like `edit()` facilitate record editing by prompting users to select the record to be edited and providing options to modify the relevant information. The system then updates the patient record accordingly, maintaining data accuracy and integrity.

3.2.6 Record Deletion and Management:

Implementation:
The system provides functionality for deleting patient records when necessary, such as when a patient's information is no longer relevant or required. This helps streamline the database and maintain data hygiene.

Code Implementation:
Functions like `delete()` allow users to delete patient records by specifying the criteria for deletion, such as patient ID or name. The system then removes the selected records from the database, ensuring efficient data management.

4. RESULT AND ANALYSIS

The implementation of the HMS system in C programming language has resulted in a functional and user-friendly administrative experience. Below, we provide an analysis of the system's performance, usability:

4.1 Outputs

Fig 2: Main Loading Page

Fig 3: Record Adding Option

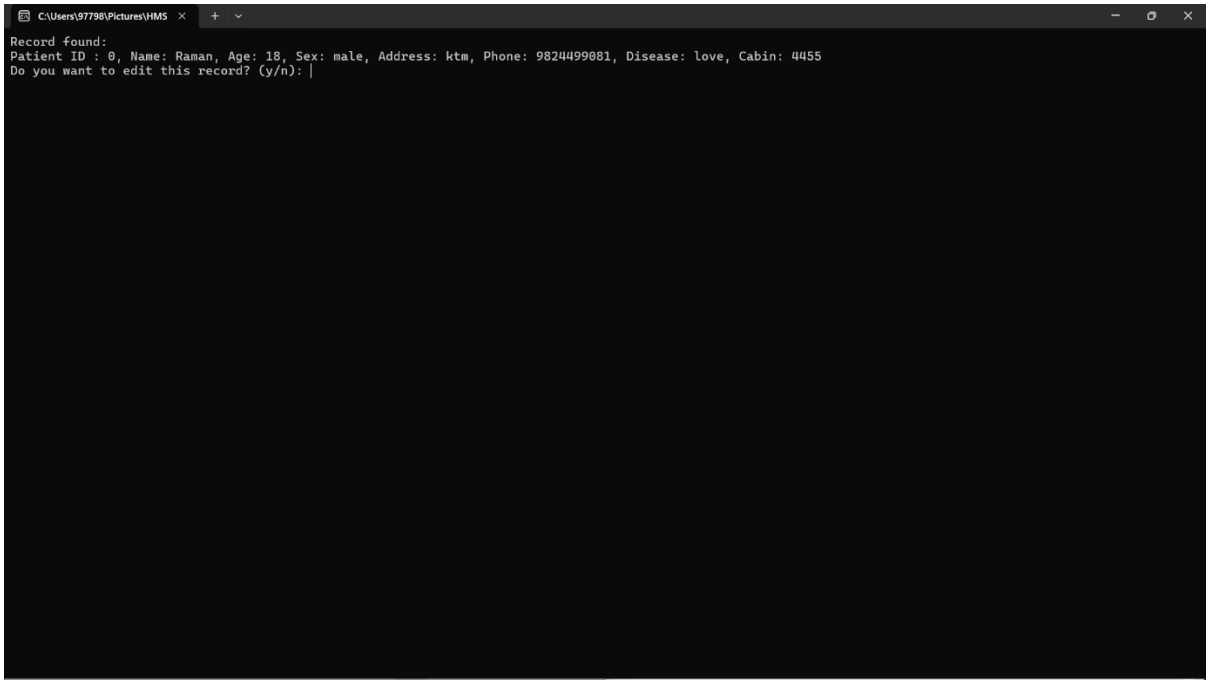

Fig 4 (Record Editing Function)

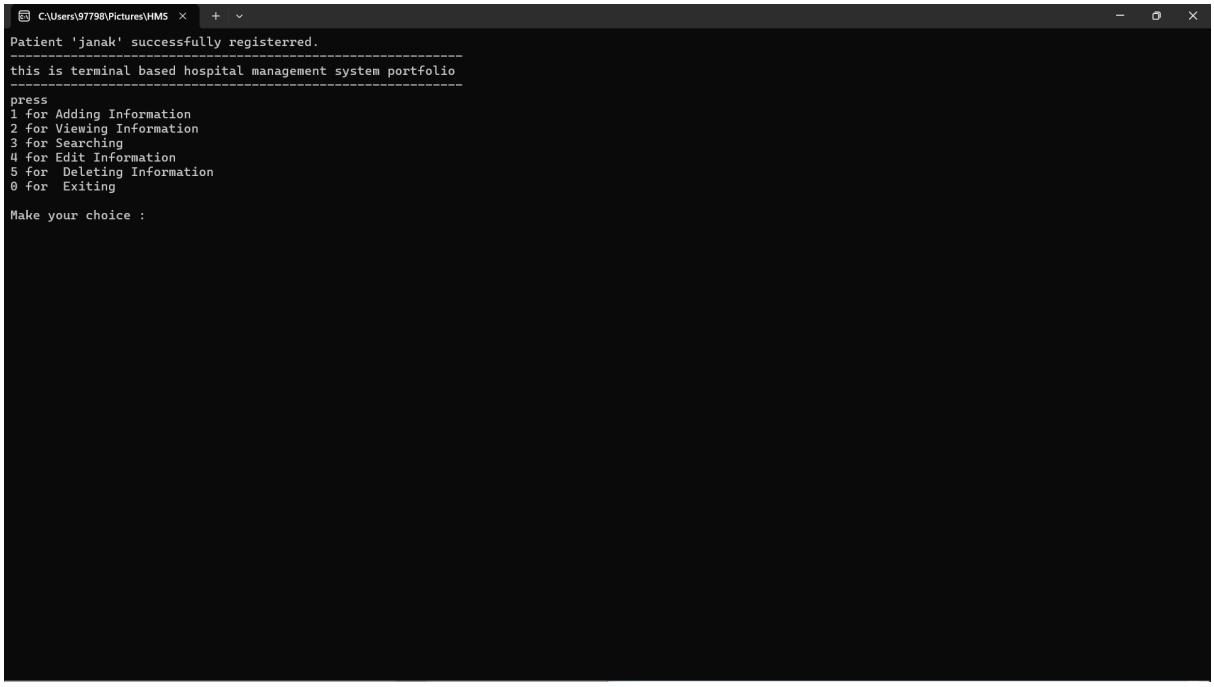

Fig 5: (Successful registration and main panel)

Fig 6 Data viewing operation

Fig 7 (Search Panel)

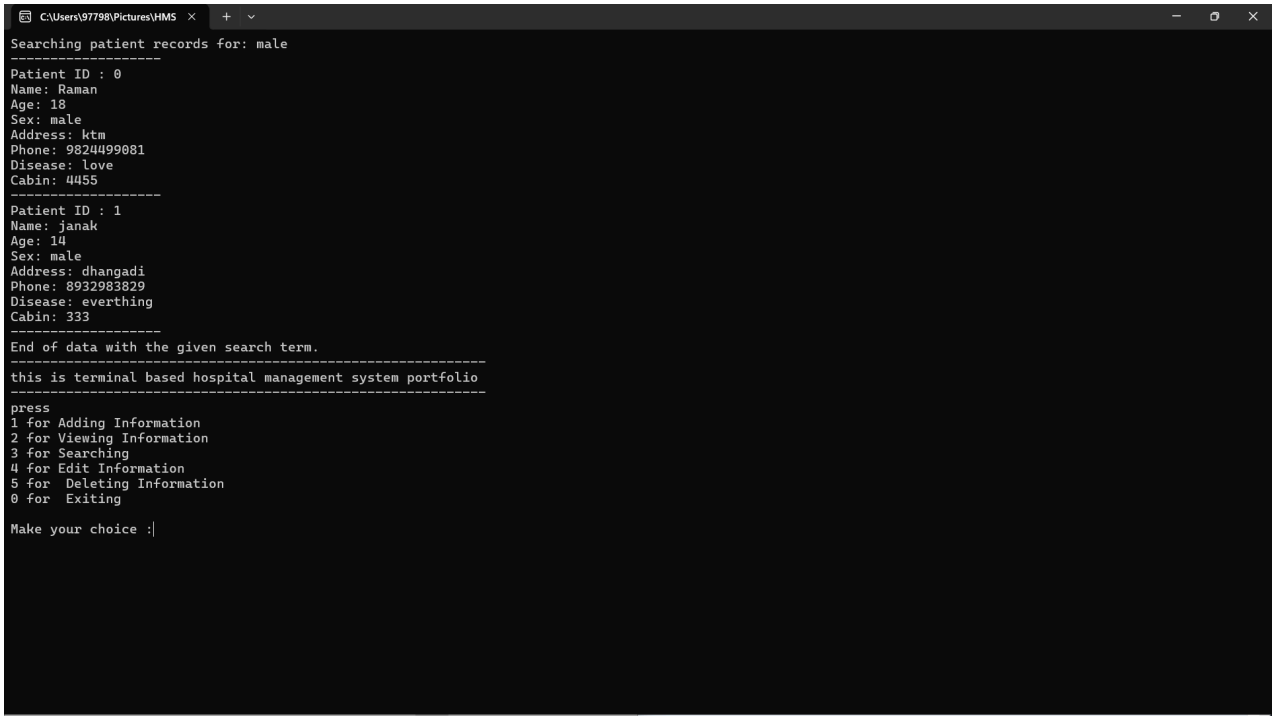

Fig 8 (Search according to field)

Fig 9 Deleting Record

4.2 Conclusion from result and analysis:

The Hospital Management System (HMS), crafted with precision using cutting-edge technology, aims to revolutionise the healthcare administration. With its intuitive interface and robust functionality, it simplifies patient record management, appointment scheduling, and inventory tracking. Rigorous testing ensures seamless performance and reliability, guaranteeing uninterrupted healthcare service delivery. The HMS's graceful interface and efficient algorithms streamline workflows, empowering healthcare professionals to focus on patient care. This implementation exemplifies the transformative power of technology in healthcare, offering a glimpse into the future of medical management systems.

5. CONCLUSION:

In summary, the HMS project demonstrates the effective utilisation of programming fundamentals in creating a robust hospital management system. Through this endeavour, we've gained insights into data structuring, user input validation, and system logic implementation. While the system may lack extensive scalability and advanced functionalities, it serves as a foundational tool for managing hospital operations. Future improvements could focus on enhancing user experience, incorporating additional features like billing and inventory management, and ensuring compatibility with diverse healthcare environments. Despite its current limitations, the HMS project lays a solid foundation for tackling more intricate healthcare management solutions in the future.

5.1 Limitations:

5.1.1 Scalability limitations:

The current HMS implementation may encounter scalability issues when managing a large volume of patient data or when scaling up to accommodate multiple hospitals or healthcare facilities. Increased data load and system complexity could lead to performance degradation

and resource constraints.

5.1.2 User Interface:

The existing user interface of the HMS system relies on a basic text-based console interface, which may not offer the modern, intuitive user experience expected in contemporary healthcare management software. This simplistic interface could hinder adoption by the healthcare institutions and satisfaction, especially among users accustomed to more visually appealing and user-friendly interfaces.

5.2 Future Enhancements

5.2.1 Wider Scalability

Efforts will be directed towards optimising the HMS system's scalability by redesigning database structures, implementing efficient data management strategies, and leveraging scalable cloud-based solutions.

5.2.2 Improved User Interface

Plans include the development of a modern and user-friendly graphical user interface (GUI) for the HMS system, featuring intuitive navigation, visually appealing design elements, and interactive functionalities.

5.2.3 Implementation of Tutorial and Help Features

Future iterations of the HMS system will include comprehensive tutorials, interactive guides, and contextual help features to onboard new users, educate them on system functionalities, and provide assistance when needed.

5.2.4 Expansion of Functionality

Enhancements will be made to expand the HMS system's functionality, incorporating modules for telemedicine support, patient engagement tools, and interoperability with external healthcare systems.

6. APPENDICES

6.1 Appendix A: Time Estimation Chart:

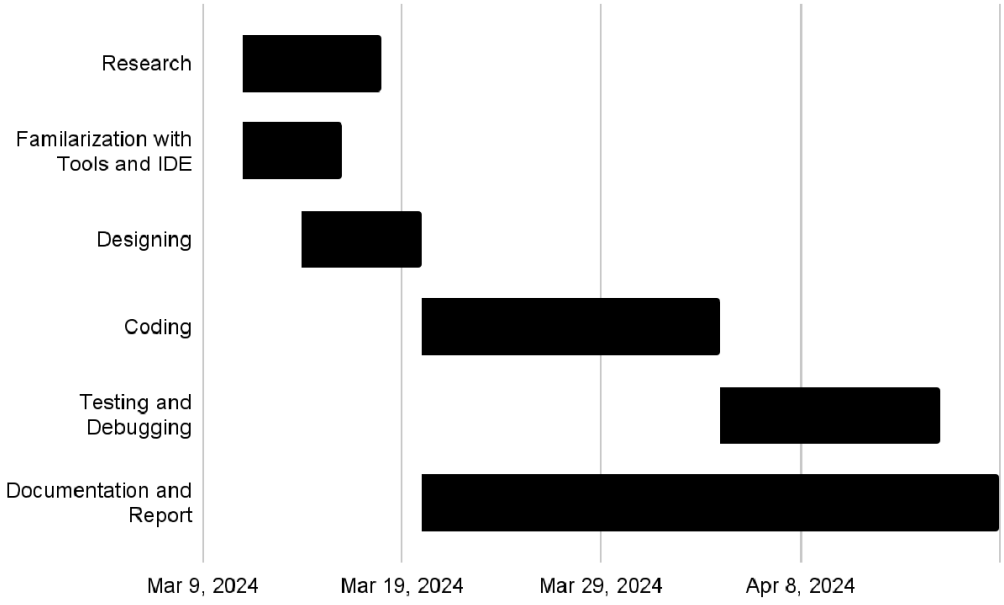

Chart: Time Estimation Chart

7. REFERENCES:

1 (Federation, 2014), A. A. (2014). Improving safety with information technology. New England Journal of Medicine, 370(25), 2418-2420.

2. (Wright, A 2017) ., 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.