Ministry of Higher Education and Scientific Research

Higher National School of Advanced Technologies

Department:

Computer Science

Report about:

Hospital Patient Management System (HPMS)

Group: B1

Group Members:

Takouk Abla

Hakkoum Meriem Alaa

Supervised by:

Dr. Lakhdari Keira

Academic Year:

2024/2025

Abstract

This project implements a complete Hospital Management System using C programming language. The system manages patient records, doctor appointments and schedule, diagnoses, patient discharge, waiting queues, and hospital structure through efficient data structures like linked lists, BSTs, stacks, and queues. Designed for console operation, it serves as a prototype for digital transformation in healthcare administration.

Table of Contents

Introduction

Healthcare facilities in Algeria face challenges managing patient flow and medical records. This system automates:

- Patient registration and tracking
- Doctor appointment scheduling
- Diagnosis record management
- Hospital resource organization

Built with modular C code.

Problem Statement

Traditional paper-based hospital systems lead to:

- 40% longer patient wait times (WHO Algeria 2022)
- Medical record errors in 15% of cases
- Inefficient doctor-patient assignment

HPMS solves these issues through digital automation.

Objectives

- Build a console-based hospital system in C that supports patients, doctors and diagnoses management.

- Utilize modular programming for maintainable and scalable code.

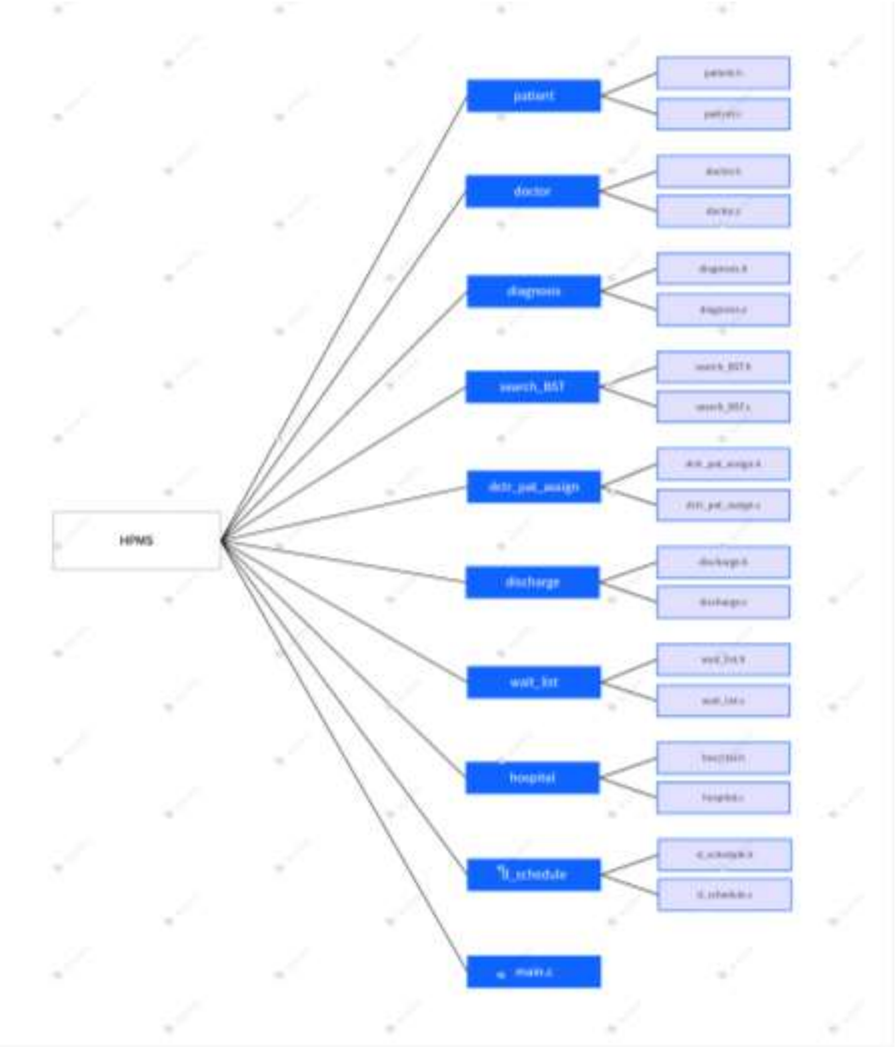
- Implement core data structures to simulate hospital workflows such as queuing, discharge, and search.

- Include real-time features like search, assignment, and scheduling.

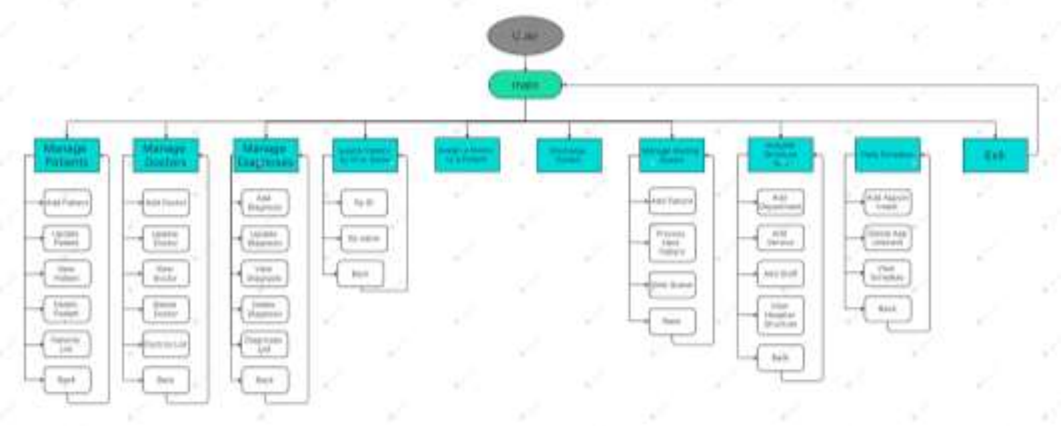
System Design & Modeling

The system follows a modular file structure where each module is responsible for a specific hospital task. Main.c acts as the controller that interacts with header and implementation files such as patient.c, doctor.c, diagnosis.c, etc. Design modeling includes Binary Search Tree for patient search, N-ary Tree for hospital structure, Stack for discharge tracking, and Queue for waiting lists.

Code Architecture



Module Interaction Flow



Data Structures Used

- **Structs:** Represent patients, doctors, diagnoses, queues, stacks, and trees.

- **Linked Lists:** Dynamic management of patient, doctor and diagnosis data.

- **Binary Search Tree (BST):** Efficient patient search by ID or name (time complexity is O(log n)).

- **N-ary Tree:** Models the hospital's hierarchical structure (time complexity is O(depth)).

- **Queue:** Manages patient waitlists in FIFO order (time complexity is O(1) for enqueue/dequeue).

- **Stack:** Tracks recently discharged patients for undo functionality (time complexity is O(1) for push/pop).

- **Arrays:** Store department names and schedule matrices.

- **Enumerators:** Manage patient statuses and doctor availability.

Algorithms Implemented

- CRUD Operations: For both patient and doctor records.

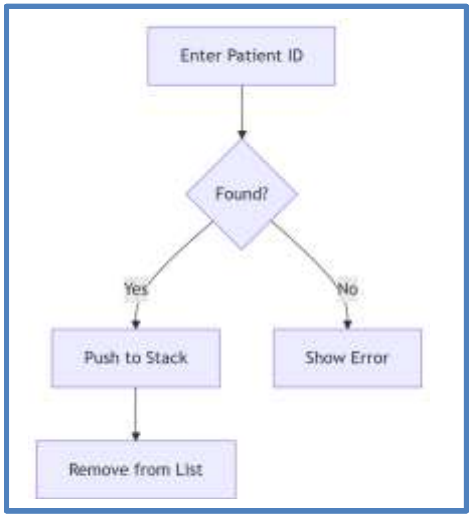
- BST Insertion & Search: For fast lookup of patient data.

- Preorder Traversal: To save/load hospital trees.

- Queue Operations: Enqueue and dequeue for waiting lists.

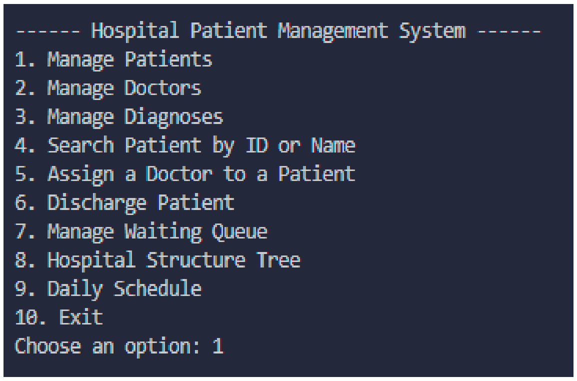
- Stack Operations: Push/pop for managing discharges and undo.

- Schedule Matrix Management: Handle doctor time slots.

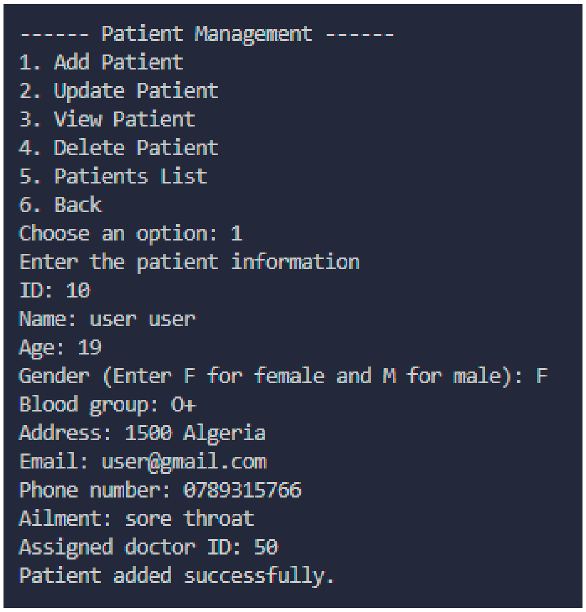


Exemple of the discharge process

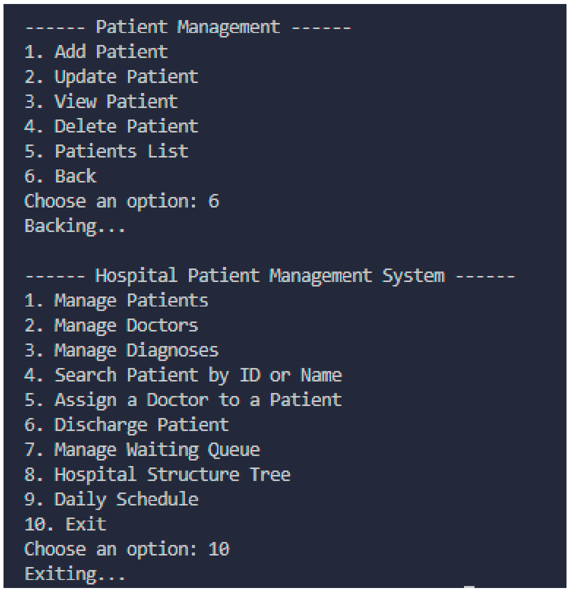
Implementation Screenshots



```
------ Hospital Patient Management System ------
1. Manage Patients
2. Manage Doctors
3. Manage Diagnoses
4. Search Patient by ID or Name
5. Assign a Doctor to a Patient
6. Discharge Patient
7. Manage Waiting Queue
8. Hospital Structure Tree
9. Daily Schedule
10. Exit
Choose an option: 1
```



```
------ Patient Management ------
1. Add Patient
2. Update Patient
3. View Patient
4. Delete Patient
5. Patients List
6. Back
Choose an option: 1
Enter the patient information
ID: 10
Name: user user
Age: 19
Gender (Enter F for female and M for male): F
Blood group: O+
Address: 1500 Algeria
Email: user@gmail.com
Phone number: 0789315766
Ailment: sore throat
Assigned doctor ID: 50
Patient added successfully.
```



```
------ Patient Management ------
1. Add Patient
2. Update Patient
3. View Patient
4. Delete Patient
5. Patients List
6. Back
Choose an option: 6
Backing...

------ Hospital Patient Management System ------
1. Manage Patients
2. Manage Doctors
3. Manage Diagnoses
4. Search Patient by ID or Name
5. Assign a Doctor to a Patient
6. Discharge Patient
7. Manage Waiting Queue
8. Hospital Structure Tree
9. Daily Schedule
10. Exit
Choose an option: 10
Exiting...
```

Conclusion

The Hospital Patient Management System project provided a hands-on opportunity to apply core C programming concepts in a real-world scenario. The modular design, structured file handling, and use of classic data structures offered valuable experience in software engineering and problem-solving. The system successfully meets its goals of improving hospital efficiency, data management, and user interaction.

References

- HPMS Project Guide – ENSTA Algiers

- GeeksforGeeks – Data Structure Tutorials

- https://youtu.be/7fSIE6B9r0k?feature=shared

- https://youtu.be/4cPOoXRxNPY?feature=shared