

PROJECT REPORT

HOSPITAL MANAGEMENT SYSTEM

Course: - Bachelors In Computer Science

Submitted By:

Ishika Singh 590028031

Abhi Omar 590028239

Submitted To: - Mr. Vinod Kumar

Date: November 30, 2025

1. ABSTRACT

The Hospital Management System is a C-based, console application that efficiently manages the records of patients, doctors, and nurses. It offers a simple interface for adding records, displaying them, editing, and erasing them. It allows data persistency by storing all the information in a file; hence, it is very useful for small-scale health centers.

2. PROBLEM DEFINITION

Manual hospital record-keeping can lead to:

- Errors in patient and staff data
- Time-consuming updates
- Difficulty in generating reports
- Lack of scalability

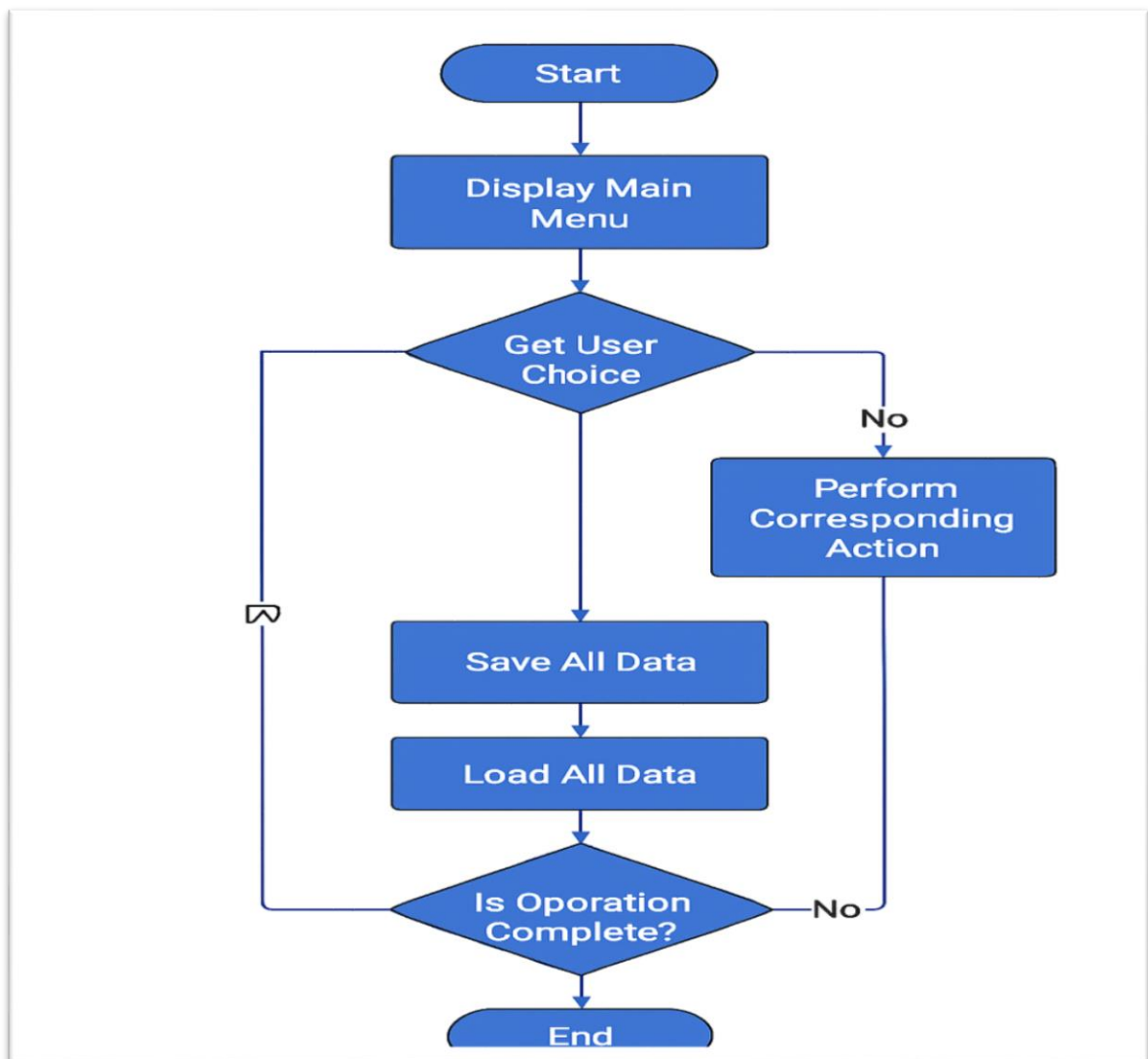
This project provides a single system that:

- Manages patient, doctor, and nurse records
 - Supports editing, searching, and reporting
 - Stores data persistently
 - Offers structured output and user-friendly interface
-

3. SYSTEM DESIGN

3.1 FLOWCHARTS WITH SHAPES

Start → Load Data → Display Menu → Get User Choice → Perform Operation → Save Data → Loop → Exit



Operations:

- Patient Management: Add, View, Edit, Delete, Search, Report
- Doctor Management: Add, View
- Nurse Management: Add, View
- Reports: Age Group Summary

3.2 ALGORITHMS:

Algorithm 1: Add Patient

1. Start
2. Prompt for name, age, problem
3. Store in patient array
4. Save to file
5. Stop

Algorithm 2: Edit Patient

1. Start
2. Ask for patient number
3. Modify fields
4. Save to file
5. Stop

Algorithm 3: Search Patient

1. Start
2. Ask for name query
3. Compare with each record
4. Display matches
5. Stop

Algorithm 4: Add Doctor

1. Start
2. Prompt for name, specialization, experience

3. Store in doctor array
4. Save to file
5. Stop

Algorithm 5: Add Nurse

1. Start
 2. Prompt for name, department, shift
 3. Store in nurse array
 4. Save to file
 5. Stop
-

4. IMPLEMENTATION DETAILS

4.1 Structures

```
struct Patient {  
    char name [50];  
    int age;  
    char problem [50];  
};  
  
struct Doctor {  
    char name [50];  
    char specialization [50];  
    int experience;  
};  
  
struct Nurse {  
    char name [50];  
    char department [50];  
    char shift [20];  
};
```

4.2 Functions

- addPatient()
- showAllPatients()
- editPatient()

- deletePatient()
 - searchPatientByName()
 - addDoctor(), showAllDoctors()
 - addNurse(), showAllNurses()
-

5. TESTING & RESULTS

Test Case 1 - Add Patient:

Input: Name=Ravi, Age=30, Problem=Fever

Expected Output: Patient added successfully

Test Case 2 - Search Patient:

Input: Name=Ravi

Expected Output: Matching record displayed

Test Case 3 - Edit Patient:

Input: Change age to 31

Expected Output: Patient updated

Test Case 4 - Delete Patient:

Input: Patient No=1

Expected Output: Patient deleted

Test Case 5 - Add Doctor:

Input: Name=Dr. Mehta, Specialization=Cardiology, Experience=10

Expected Output: Doctor added

Test Case 6 - Add Nurse:

Input: Name=Anita, Department=ICU, Shift=Night

Expected Output: Nurse added

6. CONCLUSION & FUTURE WORK

The Hospital Management System in C provides a foundational platform for managing hospital records efficiently. It demonstrates structured programming, file I/O, and modular design. Future improvements include:

- Dynamic memory allocation
 - Advanced reporting
 - GUI interface
 - Database integration
-

7. REFERENCES

- ANSI C Language Standards
 - "Programming in C" by E. Balagurusamy
 - Online C Programming Resources
-

8. APPENDIX

- Screenshots of program output

```
Welcome to Enhanced Hospital Management System!

=====
HOSPITAL MANAGEMENT SYSTEM (Patients: 0, Doctors: 0, Nurses: 0)
=====
1. Add Patient
2. Show All Patients
3. View Patient by Number
4. Search Patient by Name
5. Edit Patient
6. Delete Patient
7. Clear All Patients
8. Age Group Report
9. Add Doctor
10. Show All Doctors
11. Add Nurse
12. Show All Nurses
13. Exit
Choice (1-13): 1

Enter Patient Name: Abhi
Enter Age (0-120): 13
Enter Problem: Fever
Patient added successfully!

=====
HOSPITAL MANAGEMENT SYSTEM (Patients: 1, Doctors: 0, Nurses: 0)
=====
1. Add Patient
2. Show All Patients
3. View Patient by Number
4. Search Patient by Name
5. Edit Patient
6. Delete Patient
7. Clear All Patients
8. Age Group Report
9. Add Doctor
10. Show All Doctors
11. Add Nurse
12. Show All Nurses
13. Exit
Choice (1-13): 2

=== All Patients (1 total) ===
No.  Name      Age  Problem
-----
1    Abhi      13    Fever
```

