# HOSPITAL MANAGEMENT SYSTEM

## A  C-PROGRAMMING PROJECT

**SUBMITTED BY :**

Aryan Sharma

590026194

Batch – 61

**SUBMITTED TO :**

Dr. Srinivasan Ramachandran

Associate Professor

School Of Computer Science

# ABSTRACT

This project presents a **Hospital Management System** developed in C programming language. The system provides functionalities for patient registration, appointment booking, and doctor search. It uses structured programming concepts such as arrays, structures, and functions to manage hospital data efficiently. The project demonstrates how menu-driven programs can simulate real-world applications, offering a simple yet effective solution for hospital administration.
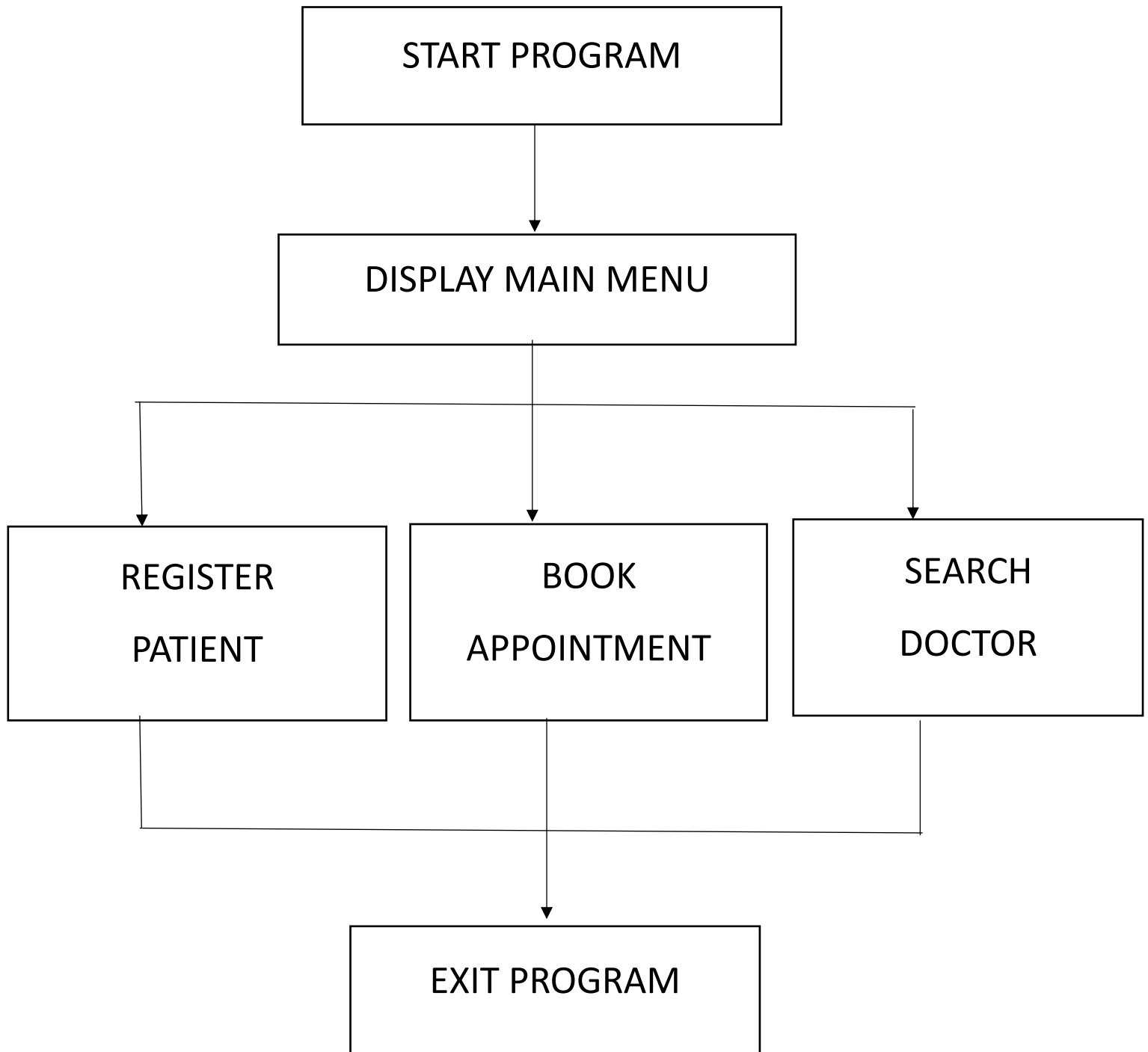
# PROBLEM DEFINITION

Hospitals often face challenges in managing patient records, doctor schedules, and appointments. Manual systems are prone to errors, delays, and inefficiency. The problem is to design a **computerized system** that:

- Registers patients with their details.

- Allows booking of appointments with doctors across multiple departments.

- Enables searching for doctors by name.

The goal is to create a **simple, menu-driven program** that improves hospital workflow and reduces administrative burden.

# SYSTEM DESIGN FLOWCHART

```
┌─────────────────────────┐
│      START PROGRAM      │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│    DISPLAY MAIN MENU    │
└─────────────────────────┘
            │
    ┌───────┼───────────────────┐
    ▼       ▼                   ▼
┌──────────┐ ┌──────────────┐ ┌──────────┐
│ REGISTER │ │    BOOK      │ │  SEARCH  │
│ PATIENT  │ │ APPOINTMENT  │ │  DOCTOR  │
└──────────┘ └──────────────┘ └──────────┘
            │
            ▼
    ┌─────────────────┐
    │  EXIT PROGRAM   │
    └─────────────────┘
```

# ALGORITHM

1. Input patient name.

2. Search patient list to verify registration.

3. If not found → prompt registration.

4. Display list of departments.

5. User selects department.

6. Display doctors in that department.

7. User selects doctor.

8. Store appointment details in appointments[]

9. Confirm booking

# IMPLEMENTATION WITH SNIPPETS

1. Patient Registration

```c
int patientCount = 0;
int appointmentCount = 0;

void registerPatient() {
    printf("\n--- Patient Registration ---\n");
    printf("Enter patient name: ");
    scanf(" %[^\n]", patients[patientCount].name);

    printf("Enter age: ");
    scanf("%d", &patients[patientCount].age);

    printf("Enter gender: ");
    scanf(" %[^\n]", patients[patientCount].gender);

    printf("Enter problem: ");
    scanf(" %[^\n]", patients[patientCount].problem);

    patientCount++;
    printf("Patient registered successfully!\n");
}
```

2. Appointment Booking

```c
void bookAppointment(struct Department dept[], int depCount) {
    char patientName[50];
    int depChoice, docChoice;

    printf("\n--- Appointment Booking ---\n");
    printf("Enter registered patient name: ");
    scanf(" %[^\n]", patientName);

    // Check patient exists
    int found = 0;
    for (int i = 0; i < patientCount; i++) {
        if (strcmp(patientName, patients[i].name) == 0) {
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Patient not found! Register first.\n");
        return;
    }
```

# TEST CASES

1. Register patient → Patient added successfully.

```
===== HOSPITAL MANAGEMENT SYSTEM =====
1. Register Patient
2. Book Appointment
3. Search Doctor
4. Exit
Enter choice: 1

--- Patient Registration ---
Enter patient name: Riya
Enter age: 19
Enter gender: Female
Enter problem: Cough, common cold and fever
Patient registered successfully!
```

1. Book appointment for unregistered patient → Error message displayed.

```
===== HOSPITAL MANAGEMENT SYSTEM =====
1. Register Patient
2. Book Appointment
3. Search Doctor
4. Exit
Enter choice: 2

--- Appointment Booking ---
Enter registered patient name: Keshav
Patient not found! Register first.
```

2. Book appointment for registered patient →
   Appointment stored.

```
===== HOSPITAL MANAGEMENT SYSTEM =====
1. Register Patient
2. Book Appointment
3. Search Doctor
4. Exit
Enter choice: 2

--- Appointment Booking ---
Enter registered patient name: Riya

Select Department:
1. Cardiology
2. Orthopedics
3. Neurology
4. ENT
5. Pediatrics
6. Pulmonology
Enter choice: 5

Doctors in Pediatrics:
1. Dr. Prashant Jain (10 AM - 1 PM) - Fee: Rs.1450
2. Dr. Ketan Parikh (2 PM - 5 PM) - Fee: Rs.1550
3. Dr. baldeep singh (5 PM - 8 PM) - Fee: Rs.1650
Choose doctor: 1

Appointment booked successfully!
```

3. Invalid department/doctor choice → Error
   message displayed.

```
===== HOSPITAL MANAGEMENT SYSTEM =====
1. Register Patient
2. Book Appointment
3. Search Doctor
4. Exit
Enter choice: 3

--- Search Doctor ---
Enter doctor name to search: Dr. Vishwanath Sood
Doctor not found!
```

4. Search doctor by name → Doctor details displayed.

```
===== HOSPITAL MANAGEMENT SYSTEM =====
1. Register Patient
2. Book Appointment
3. Search Doctor
4. Exit
Enter choice: 3

--- Search Doctor ---
Enter doctor name to search: Dr. Baldeep Singh

Doctor Found!
Name: Dr. baldeep singh
Department: Pediatrics
Time Slot: 5 PM - 8 PM
Fee: Rs.1650
```

# RESULT

All test cases executed successfully. The system correctly handled valid and invalid inputs, ensuring robustness.

# CONCLUSION AND FUTURE WORK

## CONCLUSION

The Hospital Management System demonstrates how **C structures and arrays** can be used to manage real-world data. It provides a simple interface for patient registration, appointment booking, and doctor search.

## FUTURE WORK

- Add file handling to store patient and appointment data permanently.
- Implement login system for admin and staff.
- Expand doctor database dynamically instead of fixed size.
- Add billing and prescription modules.

# REFERENCES

- Kernighan, B.W., & Ritchie, D.M. *The C Programming Language*.

- Tutorials Point – C Programming Guide

- GeeksforGeeks – Structures in C.