

HOSPITAL MANAGEMENT

Mini project | Principles of programming |

PROBLEM STATEMENT

Patient information management:

- Difficulty in maintaining and organising patient records.
- Risk of data loss or inconsistency in manual record-keeping.

Billing and Financial Management:

- Challenges inaccurate billing and invoicing for medical services.
- Difficulty in tracking and managing financial transactions.

Communication Issues:

- Ineffective communication between different departments.
- Lack of a centralized communication system for staff.

Patient Flow Management:

- Challenges in managing the flow of patients within the hospital.
- Difficulty in optimizing the use of available facilities.

OBJECTIVES

- **Patient Management:**

- Maintain detailed patient records, including personal information, medical history, and treatment plans.
- Streamline the admission, discharge, and transfer processes.
- Track patient demographics, visits, and appointments.

- **Appointment Scheduling:**

- Efficiently manage and schedule appointments for patients.
- Reduce waiting times and optimize the utilization of healthcare resources

3. Billing and Invoicing:

Automate the billing process for medical services and treatments.

Generate accurate and timely invoices for patients and insurance companies.

4. Staff Management:

Maintain employee records, including qualifications, work schedules, and roles.

Facilitate efficient communication among staff members.

5. Electronic Health Records (EHR):

Implement a secure and centralized system for storing and retrieving electronic health records.

Ensure the confidentiality and integrity of patient data.

6. Inventory Management:

Keep track of medical supplies, drugs, and equipment in real-time.

Ensure proper inventory levels to avoid shortages or excess stock.

7. Emergency Management:

Develop protocols for handling emergencies and disasters.

Ensure the availability of essential resources during critical situations.

8. Mobile Accessibility:

Provide mobile access to key features of the HMS, allowing healthcare providers to access information remotely.

9. Continuous Improvement:

Implement a system for feedback and continuous improvement, allowing the hospital to adapt to changing needs and technologies.

10. Security and Compliance:

Implement robust security measures to safeguard patient data.

Ensure compliance with healthcare regulations and standards.

REQUIREMENTS ANALYSIS

1. Stakeholder Identification:

- Identify all stakeholders, including healthcare professionals (doctors, nurses, administrators), patients, IT staff, and regulatory bodies.
- Conduct interviews and surveys to gather insights into their specific requirements and expectations.

2. Functional Requirements:

- **Patient Management:**
 - Record patient demographics, medical history, and insurance details.
 - Enable patient admission, discharge, and transfer processes.

- **Appointment Scheduling:**

- Allow users to schedule, modify, and cancel appointments.
- Send appointment reminders to patients.

- **Billing and Invoicing:**

- Generate accurate invoices for services rendered.
- Integrate with insurance systems for claims processing.

- **Inventory Management:**

- Track and manage medical supplies, drugs, and equipment.
- Set up alerts for low stock levels.

- **Staff Management:**

- Maintain employee records, including qualifications and schedules.
- Facilitate communication between staff members.

- **Reporting and Analytics:**
 - Generate reports on patient outcomes, resource utilization, and financial performance.
 - Implement analytics for data-driven decision-making.
- **Integration with Laboratory and Imaging Systems:**
 - Integrate with laboratory and imaging systems for test orders and results.
 - Ensure seamless communication between different departments.
- **Security and Compliance:**
 - Implement robust security measures to protect patient data.
 - Ensure compliance with healthcare regulations and standards.
- **Quality of Care:**
 - Facilitate evidence-based practices and clinical decision support.
 - Implement protocols for monitoring and improving patient care.

SOURCE CODE

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
// Structure to store patient details
```

```
struct patient {
```

```
    char name[50];
```

```
    int age;
```

```
    char gender[10];
```

```
    char address[100];
```

```
    char contact[15];
```

```
};
```

// Function to add patient details

```
void addPatient(struct patient* p, int n) {
```

```
    printf("--- Add Patient Details ---\n");
```

```
    printf("Enter Name: ");
```

```
    scanf("%s", p[n].name);
```

```
    printf("Enter Age: ");
```

```
    scanf("%d", &p[n].age);
```

```
    printf("Enter Gender: ");
```

```
    scanf("%s", p[n].gender);
```

```
    printf("Enter Address: ");
```

```
    scanf("%s", p[n].address);
```

```
printf("Enter Contact Number: ");
scanf("%s", p[n].contact);
printf("Patient details added successfully!\n");
}
```

// Function to display patient details

```
void displayPatients(struct patient* p, int n) {
    printf("--- Patient Details ---\n");
    for (int i = 0; i < n; i++) {
        printf("Patient %d:\n", i + 1);
        printf("Name: %s\n", p[i].name);
        printf("Age: %d\n", p[i].age);
        printf("Age: %d\n", p[i].age);
        printf("Gender: %s\n", p[i].gender);
        printf("Address: %s\n", p[i].address);
        printf("Contact Number: %s\n", p[i].contact);
        printf("-----\n");
    }
}
```

```
int main() {  
    int choice, count = 0;  
    struct patient patients[100]; // Array to store patient  
    details  
  
    while (1) {  
        printf("\n--- Hospital Management System ---\n");  
        printf("1. Add Patient\n");  
        printf("2. Display Patients\n");  
        printf("3. Exit\n");  
        printf("Enter your choice: ");  
        scanf("%d", &choice);  
  
        switch (choice) {  
case 1:  
            addPatient(patients, count);  
            count++;  
            break;
```

case 2:

displayPatients(patients, count);

break;

case 3:

printf("Thank you for using the system. Exiting...\n");

exit(0);

default:

printf("Invalid choice!\n");

}

}

return 0;

main.c

```
61
62         case 2:
63             displayPatients(patients, count);
64             break;
65
66         case 3:
67             printf("Thank you for using the system. Exiting...\n");
68             exit(0);
69
70         default:
71             printf("Invalid choice!\n");
72     }
73 }
74
75 return 0;
76 }
77
78
79
```

input

```
-- Hospital Management System ---
. Add Patient
. Display Patients
. Exit
Enter your choice: 
```

```
- Hospital Management System ---
Add Patient
Display Patients
Exit
```

```
Enter your choice: 2
- Patient Details ---
```

```
- Hospital Management System ---
Add Patient
Display Patients
Exit
```

```
Enter your choice: 1
- Add Patient Details ---
Enter Name: Chinmayee
Enter Age: 18
Enter Gender: female
Enter Address: Atria institute of technology
Enter Contact Number: Patient details added successfully!
```

```
- Hospital Management System ---
Add Patient
```

9. CONCLUSION

Since we are entering details of the patients electronically in the "Hospital Management System", data will be secured. Using this application we can retrieve a patient's history with a single click. Thus processing information will be faster. It guarantees accurate maintenance of Patient details. It easily reduces the bookkeeping task and thus reduces the human effort and increases accuracy speed.



Submitted by,
Lakhani S
Haditha M
Chimayo K L