BAS**AVARAJESWARI GROUP OF INSTITUTIONS**

# Ballari Institute of Technology & Management, Ballari

# AN AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITY JNANA SANGAMA, BELAGAVI-590018

****

**Software Engineering**

**Mini-Project Report**

**On**

**“HOSPITAL MANAGEMENT SYSTEM”**

Submitted in partial fulfillment of the requirements for the award of degree

**Bachelor of Engineering**

**In**

**CSE-DATA SCIENCE ENGINEERING**

**Submitted by**

**Names : USN**

**1. CHETANA HK 3BR22CD007**

**2. MOUNIKA M 3BR22CD039**

**Project Coordinator Head of the Department**

**Prof. Dr . Jagadeesh R M Dr. Aradhana D**

**Associate Professor Professor & Head of the**

**Department of CSE-DS , BITM, Ballari Department of CSE-DS , BITM, Ballari**

BASAVARAJESWARI GROUP OF INSTITUTIONS

# BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT BALLARI.

# An Autonomous institute under

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY JNANA SANGAMA, BELAGAVI-590018

NACC Accredited Institution\*

**(RecognizedbyGovt.ofKarnataka, approvedbyAICTE, NewDelhi&AffiliatedtoVisvesvarayaTechnological University, Belagavi)**

"**Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road,Allipur, Ballari-583104 (Karnataka)(India)**

**Ph:08392–237100/237190,Fax:08392–237197**

# DEPARTMENT

# OF

**CSE-DATA SCIENCE ENGINEERING**

**CERTIFICATE**

This is to certify that the Software Engineering Project entitled **“Hospital management system”** has been successfully completed by **Chetana HK, Mounika M**USN **3BR22CD007, 3BR22CD039** a Bonafede students of Ballari Institute of Technology and Management, Ballari for the partial fulfillment of the requirements for the **Bachelor of Engineering Degree in CSE-Data science** under VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belagavi during the academic year 2023-2024.

**Project Coordinator Head of the Department**

**Prof. Dr . Jagadeesh R M Dr. Aradhana D**

**Associate Professor Professor & Head of the**

**Department of CSE-DS , BITM, Ballari Department of CSE-DS , BITM, Ballari**

**TABLE OF CONTENT**

1. Abstract
2. Introduction
3. About the project
4. Problem statement
5. Objectives
6. Software requirements
7. Hardware requirements
8. Applications
9. implementation of code
10. Result and Analysis
11. Conclusion

**Abstract:**

The Hospital Management System (HMS) is a comprehensive software solution designed to streamline and optimize the daily operations of a healthcare facility. In the dynamic landscape of the medical field, efficient management of resources, accurate record-keeping, and enhanced patient care are imperative. The HMS serves as a pivotal tool to address these needs, offering a sophisticated yet user-friendly platform for healthcare professionals to manage various aspects of hospital administration.

This system employs advanced data structures, such as structures, linked lists, and arrays, to organize and manage diverse information related to patients, doctors, and other crucial aspects of hospital functioning. The seamless integration of these data structures facilitates efficient storage, retrieval, and manipulation of data, contributing to a more organized and responsive healthcare environment.

Key functionalities of the Hospital Management System include doctor and patient information management, appointment scheduling, and the provision of a centralized database for medical records. The system aims to minimize manual tasks, reduce paperwork, and enhance the overall accuracy and accessibility of healthcare information.

Additionally, the HMS is designed to operate on standard hardware configurations, making it adaptable to various healthcare settings. The software is compatible with commonly used operating systems and can be implemented in both small clinics and large medical institutions.

**INTRODUCTION**

The healthcare industry is undergoing a transformative evolution, necessitating advanced solutions for efficient management and organization of hospital operations. The Hospital Management System (HMS) emerges as a pivotal response to the complex challenges faced by healthcare institutions in delivering quality patient care while maintaining seamless administrative processes.

In the contemporary healthcare landscape, the demand for streamlined and automated systems has become more pronounced. The traditional paper-based approaches to record-keeping and appointment scheduling have proven to be inadequate in managing the ever-expanding volume of patient data and the dynamic nature of medical services.

The Hospital Management System is designed to address these challenges by leveraging cutting-edge technology to create an integrated and user-friendly platform. This system not only ensures the smooth flow of administrative tasks but also enhances the overall quality of patient care. It acts as a centralized repository for a myriad of healthcare information, ranging from patient records to doctor details and appointment schedules

The core objective of the HMS is to optimize and automate various aspects of hospital administration. This includes efficient management of patient information, systematic scheduling of appointments, and comprehensive recording of medical histories. By doing so, the system contributes to the reduction of manual errors, minimizes paperwork, and ultimately facilitates more informed decision-making by healthcare professionals.

Moreover, the introduction of the Hospital Management System aligns with the broader technological shift in healthcare towards digitization. It enables healthcare providers to adapt to the evolving landscape, promoting enhanced efficiency, accuracy, and accessibility of critical information.

As we delve deeper into the functionalities and features of the Hospital Management System, it becomes evident that this software solution is poised to revolutionize how healthcare institutions manage their operations. Its implementation promises not only a more organized and responsive healthcare environment but also a significant improvement in the overall healthcare delivery process.

**ABOUT PROJECT:**

A hospital management system (HMS) is a software solution designed to streamline and automate various administrative and operational tasks within a healthcare facility. Key features typically include:

**Patient Registration**: Capture and manage patient information, including personal details, medical history, and contact information.

**Appointment Scheduling**: Allow staff to schedule appointments for patients efficiently, helping to manage healthcare provider availability.

**Billing and Invoicing**: Handle financial transactions, generate bills, and manage invoicing for services rendered.

**Electronic Health Records (EHR**): Maintain digital health records, making it easier to access and update patient information securely.

**Inventory Management**: Track and manage medical supplies, equipment, and pharmaceuticals within the hospital.

**Staff Management**: Monitor and manage staff schedules, roles, and responsibilities, ensuring proper allocation of resources.

**Laboratory and Imaging Integration**: Facilitate the integration of laboratory and imaging results with patient records for comprehensive healthcare management.

**Pharmacy Management**: Streamline medication dispensing, manage drug inventory, and ensure accurate prescription tracking. Reporting and Analytics: Generate reports for performance analysis, financial insights, and other key metrics to aid decision-making.

**Security and Compliance**: Implement measures to ensure the security and confidentiality of patient data, while adhering to regulatory compliance standards.

When implementing a hospital management system project, considerations should include system scalability, user training, data migration, and integration with existing healthcare infrastructure. The goal is to enhance overall efficiency, reduce errors, and improve the quality of patient care.

**PROBLEM STATEMENT:**

**PROJECT TITLE :**

“HOSPITAL MANAGEMENT SYSTEM”

**PROJECT OVERVIEW**:

Develop a comprehensive Hospital Management System that automates patient registration, appointment scheduling, medical records management, billing, and inventory tracking, ensuring seamless communication between staff, improving overall efficiency, and enhancing the quality of healthcare services provided.

**SOFTWARE REQUIRMENT:**

The software used for a Hospital Management System (HMS) project includes a combination of programming languages, databases, and other tools to develop, deploy, and maintain the system. Here's a list of typical software components for an HMS project:

**1.Programming language** -> c , c++

**2. Operating system** = Window XP/10/8.

**3.Complier**=c complier.

**4** **.C Library** =stdio.h ,conio.h and stdlib.h.

**HARDWARE REQUIREMENTS:**

The hardware requirements for a Hospital Management System (HMS) project are relatively standard and depend on the scale and complexity of the system. Here's a basic outline of the hardware components typically used:

**1.Micoprocessor**:1.0GHz and above CPU based on either AMD or INTEL microprocessor architecture.

**2.Main memory** :2 GB RAM.

**3.Hard Disk** :40 GB.

**5.Hard disk speed in RPM**:5400 RPM.

**6.Keyboard** :QWERTY keyboard.

**7.Monitor** : 1024\*768 display resolution.

**APPLICATIONS OF HMS:**

The application of a Hospital Management System (HMS) project is extensive and contributes significantly to the efficiency and effectiveness of healthcare operations. Here are key applications of an HMS:

**1.Patient Management:**

* Streamlining patient registration processes.
* Managing and updating patient demographics, medical history, and contact information.

**2.Appointment Scheduling:**

* Facilitating easy and efficient scheduling of patient appointments with doctors.
* Sending reminders to patients about upcoming appointments.

**3.Doctor Information and Management:**

* Maintaining a centralized database of doctors, including their qualifications, specializations, and availability.
* Assigning patients to specific doctors based on their needs.

**4.Billing and Invoicing:**

* Automating the billing process for medical services, consultations, and procedures.
* Generating invoices and tracking payments.

**5.Medical Records Management:**

* Digitizing and centralizing patient medical records for easy access and retrieval.
* Storing diagnostic reports, prescriptions, and treatment plans.

**6.Pharmacy Management:**

* Integrating with pharmacy systems to manage medication prescriptions and dispensing.
* Maintaining an inventory of available medications and managing restocking.

**7.Laboratory Information System (LIS):**

* Integrating with laboratory systems to manage and track diagnostic tests.
* Recording and retrieving laboratory results efficiently.

**8.Radiology Information System (RIS):**

* Managing radiology appointments and schedules.
* Storing and retrieving radiology images and reports.

**9.Inventory Management:**

* Monitoring and managing hospital inventory, including medical supplies and equipment.
* Automating reordering processes to ensure sufficient stock levels.

**10.Reporting and Analytics:**

* Generating reports on patient demographics, hospital admissions, and financial data.
* Providing analytics for hospital administrators to make informed decisions.

**11.Security and Compliance:**

* Ensuring the security of sensitive patient information through access controls and encryption.
* Adhering to healthcare regulations and standards to maintain compliance.

**12.Telemedicine Integration:**

* Integrating telemedicine features for remote consultations and follow-ups.
* Facilitating virtual appointments and improving accessibility for patients.

**13.Mobile Access:**

* Providing a mobile-friendly interface for healthcare professionals to access information on the go.
* Allowing patients to view appointments, test results, and other relevant data through a mobile app.

**14.Emergency Management:**

* Streamlining emergency patient admissions and coordinating responses.
* Ensuring quick access to critical patient information during emergencies

**.**

**15.Collaboration and Communication:**

* Facilitating communication between hospital staff through messaging or notification systems.
* Enhancing collaboration among different departments for improved patient CARE.

**IMPLEMENTATION OF HMS;**

Implementing a Hospital Management System (HMS) project involves several steps, including designing the system, coding, testing, and deployment. Here's a high-level overview of the implementation process:

**1.Requirement Analysis:**

* Understand the specific requirements of the hospital or healthcare facility.
* Identify key features, user roles, and integration points.

**2.System Design:**

* Design the database schema for storing patient records, doctor information, appointments, etc.
* Create a user interface design for different modules
* Plan the overall system architecture.

**3.Database Setup:**

* Choose a suitable database management system (e.g., MySQL, PostgreSQL).
* Create the necessary tables for storing data (patients, doctors, appointments, etc.).

**4.Backend Development:**

* Implement server-side logic using a programming language (e.g., C++, Java, Python).
* Develop functionalities such as patient registration, doctor management, and appointment scheduling.

**5.Frontend Development:**

* Create the user interface using HTML, CSS, and JavaScript.
* Implement pages for patient registration, appointment booking, and other relevant features.

**6.Integration:**

* Connect the frontend and backend components to enable seamless communication.
* Ensure data flows accurately between the user interface and the database.

**7.Authentication and Authorization**:

* Implement secure login mechanisms for different user roles (admin, doctor, receptionist).
* Set up access control to restrict unauthorized access to sensitive information.

**8.Testing:**

* Conduct unit testing for individual components.
* Perform integration testing to ensure proper functioning of the entire system.
* Test for security vulnerabilities and address any issues.

**9.User Acceptance Testing (UAT):**

* Collaborate with end-users to validate that the system meets their requirements.
* Address any feedback or issues identified during the testing phase.

**10.Deployment:**

* Deploy the HMS on the chosen server or cloud platform.
* Configure the system to ensure optimal performance.

**11.Training:**

* Train hospital staff on using the new system.
* Provide documentation for administrators, doctors, and other users

**12.Monitoring and Maintenance:**

* Implement monitoring tools to track system performance and detect issues.
* Regularly update and maintain the system, addressing any bugs or adding new features as needed.

**13.User Support:**

* Provide ongoing support for users encountering issues with the system.
* Address feedback and make improvements based on user experience.

**14.Security Measures:**

* Regularly update security measures to protect against potential threats.
* Keep abreast of industry best practices for healthcare data security.

**15.Scale and Optimize:**

* Evaluate system performance and scalability as the hospital grows.
* Optimize database queries and system architecture for efforts.

**IMPLEMENTATION OF CODE (HMS):**

**//PROGRAM FOR HOSPITAL MANEGEMENT SYSTEM:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_PATIENTS 100

struct Patient {

int id;

char name[50];

int age;

char gender;

char diagnosis[100];

int admitted;

};

struct Patient patients[MAX\_PATIENTS];

int patient\_count = 0;

void admitPatient() {

if (patient\_count >= MAX\_PATIENTS) {

printf("Hospital capacity reached. Cannot admit more patients.\n");

return;

}

printf("Enter patient name: ");

scanf("%s", patients[patient\_count].name);

printf("Enter patient age: ");

scanf("%d", &patients[patient\_count].age);

printf("Enter patient gender (M/F): ");

scanf(" %c", &patients[patient\_count].gender);

printf("Enter diagnosis: ");

scanf("%s", patients[patient\_count].diagnosis);

patients[patient\_count].id = patient\_count + 1;

patients[patient\_count].admitted = 1; // Mark patient as admitted

patient\_count++;

printf("Patient admitted successfully.\n");

}

void dischargePatient() {

int id;

printf("Enter patient ID to discharge: ");

scanf("%d", &id);

if (id >= 1 && id <= patient\_count && patients[id - 1].admitted) {

patients[id - 1].admitted = 0; // Mark patient as discharged

printf("Patient %s discharged successfully.\n", patients[id - 1].name);

} else {

printf("Invalid patient ID or patient already discharged.\n");

}

}

void displayPatientInfo() {

int id;

printf("Enter patient ID: ");

scanf("%d", &id);

if (id >= 1 && id <= patient\_count && patients[id - 1].admitted) {

printf("Patient ID: %d\n", patients[id - 1].id);

printf("Name: %s\n", patients[id - 1].name);

printf("Age: %d\n", patients[id - 1].age);

printf("Gender: %c\n", patients[id - 1].gender);

printf("Diagnosis: %s\n", patients[id - 1].diagnosis);

printf("Admission Status: Admitted\n");

} else {

printf("Patient not found or discharged.\n");

}

}

void displayAdmittedPatients() {

printf("Admitted Patients:\n");

for (int i = 0; i < patient\_count; i++) {

if (patients[i].admitted) {

printf("ID: %d, Name: %s, Age: %d, Gender: %c, Diagnosis: %s\n",

patients[i].id, patients[i].name, patients[i].age,

patients[i].gender, patients[i].diagnosis);

}

}

}

int main() {

int choice;

do {

printf("\nHospital Management System\n");

printf("1. Admit Patient\n");

printf("2. Discharge Patient\n");

printf("3. Display Patient Information\n");

printf("4. Display Admitted Patients\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

admitPatient();

break;

case 2:

dischargePatient();

break;

case 3:

displayPatientInfo();

break;

case 4:

displayAdmittedPatients();

break;

case 5:

printf("Exiting...\n");

break;

default:

printf("Invalid choice. Please enter a number between 1 and 5.\n");

}

} while (choice != 5);

return 0;

}

**Algorithm :**

Include necessary header files (stdio.h, stdlib.h, string.h).

Define the maximum number of patients (MAX\_PATIENTS) and a structure for patient information (struct Patient).

Declare an array patients to store patient records and initialize a variable patient\_count to keep track of the number of patients.

Implement functions for admitting a patient, discharging a patient, displaying individual patient information, and displaying the list of admitted patients.

In the main function, use a do-while loop to display a menu of options to the user and perform the corresponding action based on their choice.

Exit the loop when the user chooses option 5 (Exit).

Algorithum:

Start

Display menu options

User input: Choice (1-5)

1: Admit Patient

Prompt for patient details

Add patient to the array

2: Discharge Patient

Prompt for patient ID

Mark patient as discharged

3: Display Patient Information

Prompt for patient ID

Display individual patient details

4: Display Admitted Patients

Display details of all admitted patients

5: Exit

Exit the loop and end the program

Default: Invalid choice message

Repeat steps 2-3 until the user chooses to exit (option 5).

End

**CODE FLOWCHART :**



Hospital Management

System (HMS)

Start Program

Display Menu

User Inputs

Switch Case

Invalid Choice

Admit Display error

Discharge

Display

Exit

Display Result

Repeat loop

User exits

End Program

**RESULT AND ANYALISIS(OUTPUT):**

**OUTPUT:**

Hospital Management system

1. Admit Patient

2. Discharge Patient

3. Display Patient Information

4. Display Admitted Patients

5. Exit

Enter your choice: 1

Enter patient name: Rohan

Enter patient age: 16

Enter patient gender (M/F): M

Enter diagnosis: fever

Patient admitted successfully.

Hospital Management System

1. Admit Patient

2. Discharge Patient

3. Display Patient Information

4. Display Admitted Patients

5. Exit

Enter your choice: 3

Enter patient ID: 123

Patient not found or discharged.

Hospital Management System

1. Admit Patient

2. Discharge Patient

3. Display Patient Information

4. Display Admitted Patients

5. Exit

Enter your choice: 4

Admitted Patients:

ID: 1, Name: Rohan, Age: 16, Gender: M, Diagnosis: fever

Hospital Management System

1. Admit Patient

2. Discharge Patient

3. Display Patient Information

4. Display Admitted Patients

5. Exit

Enter your choice:3

Enter patient ID: 1

Patient is found or discharged.

Hospital Management system

1. Admit Patient

2. Discharge Patient

3. Display Patient Information

4. Display Admitted Patients

5. Exit

Enter your choice: 1

Enter patient name: Radha

Enter patient age: 20

Enter patient gender (M/F): F

Enter diagnosis: Diabitis

Patient admitted successfully.

Hospital Management System

1. Admit Patient

2. Discharge Patient

3. Display Patient Information

4. Display Admitted Patients

5. Exit

Enter your choice: 4

Admitted Patients:

ID: 2, Name:Radha , Age: 20, Gender: F, Diagnosis: Diabites

Hospital Management System

1. Admit Patient

2. Discharge Patient

3. Display Patient Information

4. Display Admitted Patient

5. Exit

Enter your choice: 2

Enter patient ID to discharge: 1

Patient Rohan discharged successfully.

Hospital Management System

1. Admit Patient

2. Discharge Patient

3. Display Patient Information

4. Display Admitted Patients

5. Exit

Enter your choice:2

Enter patient ID to discharge: 2

Patient Radha discharged successfully.

Patient admitted successfully.

Hospital Management System

1. Admit Patient

2. Discharge Patient

3. Display Patient Information

4. Display Admitted Patients

5. Exit

Enter your choice: 5 EXIT!!!!

**THE CONCLUSION:**

In conclusion, the Hospital Management System project aims to enhance healthcare efficiency by automating various administrative and clinical processes. Through features like patient record management, appointment scheduling, and inventory tracking, the system optimizes workflow, reduces errors, and ultimately improves patient care. Implementing this technology can lead to streamlined operations, better resource utilization, and a more organized healthcare environment. In conclusion, the Hospital Management System project aims to enhance healthcare efficiency by automating various administrative and clinical processes. Through features like patient record management, appointment scheduling, and inventory tracking, the system optimizes workflow, reduces errors, and ultimately improves patient care. Implementing this technology can lead to streamlined operations, better resource utilization, and a more organized healthcare environment.