HEALTH CARE SYSTEM

A Course End Project Report-OBJECT ORIENTED PROGRAMMING LABORATORY(A8602)

Submitted in the partial fulfilment of the

Requirements

For the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Submitted

By

K.S.VADANA SRI	**22881A05G1**
P.BHAVYA	**22881A05J0**
M.ANANYA	**22881A05G9**
B.LAYA	**22881A05D7**
PL.KEERTHANA	**22881A05H9**
B.RUKMINI	**22881A05D4**

Under The Esteemed Guidance Of

Dr.U.Seshadri

Associate professor

Cse

Department Of Computer Science And Engineering

VARDHAMAN COLLEGE OF ENGINEERING,HYDERABAD

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015

Certified

Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India,JANUARY,2024

VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

An autonomous institute affiliated to JNTUH

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Course End Project titled **"HEALTH CARE SYSTEM" is** carried out by Mr./MsKS.VADANA SRI, P.BHAVYA, M.ANANYA, B.LAYA, PL.KEERTHANA, B.RUKMINI towards **A8602 – Object Oriented Programming Laboratory** course and submitted to the **Department of Computer Science and Engineering**, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in the **Department of Computer Science and Engineering** during the Academic year 2023-24.the

Name & Signature of the Instructor

Dr. U.Seshadri
Associate Professor,

Name & Signature of the HOD

Dr. Ramesh Karnati HOD, CSE CSE

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr. U.Seshadri**, Associate Professor, Department of Computer Science and Engineering, Vardhaman College of Engineering, for her able guidance and useful suggestions, which helped us in completing the design part of potential project in time.

We particularly thankful to **Dr. Ramesh Karnati**, Associate Professor & Head, Department of Computer Science and Engineering for his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R.Ravindra**, for having provided all the facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartfelt thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science and Engineering for their valuable support and generous advice. Finally, thanks to all our friends and family members for their continuous support and enthusiastic help.

K.S.VADANA SRI **22881A05G1**
P.BHAVYA **22881A05J0**
M.ANANYA **22881A05G9**

B.LAYA **22881A05D7**

PL.KEERTHANA **22881A05H9**
B.RUKMINI **22881A05D4**

INDEX

1.INTRODUCTION

In the dynamic landscape of healthcare management, technological advancements have become instrumental in improving patient care, enhancing organizational efficiency, and streamlining administrative processes. The Healthcare System presented here leverages the Java programming language to create a versatile and interactive platform. The system is designed to address key aspects of healthcare management, focusing on patient registration and information display. Through a console-based interface, healthcare professionals can seamlessly register new patients, capturing vital details such as patient ID, name, address, and phone number. The system also provides a feature to display the registered patients, offering a quick overview of their information. Utilizing object-oriented principles, the program employs a Patient class to encapsulate patient-related details, fostering code organization and maintainability. The main functionality is encapsulated within the HealthcareSystem class, featuring a user-friendly menu-driven interface. The system's flexibility and ease of use make it adaptable for small-scale healthcare environments. As the program runs continuously until the user chooses to exit, it reflects the persistent nature of healthcare systems. The introduction of a loop structure ensures ongoing interaction, allowing healthcare professionals to register patients and retrieve information at their convenience. This Java Healthcare System serves as a foundational example, illustrating the potential for future expansions and enhancements. The inclusion of a database, graphical user interfaces, and additional features would be essential for a comprehensive healthcare management solution tailored to diverse healthcare settings.

2. OBJECTIVE OF THE PROJECT

1. Preventive Care: Emphasizing health education, vaccinations, screenings, and lifestyle interventions to prevent the onset of diseases.

2. Curative Care: Providing medical treatments and interventions to cure or manage diseases and health conditions.

3. Patient-Centered Care: Focusing on the needs and preferences of individual patients, involving them in decision-making, and respecting their dignity and autonomy.

4.Access to Care: Ensuring equitable access to healthcare services, regardless of

socio-economic status, geographic location, or other factors.

5.Quality of Care: Delivering healthcare services that meet accepted standards of quality, safety, and effectiveness.

6.Cost-Effectiveness: Striving to provide efficient and cost-effective healthcare services to maximize the benefits of available resources.

7.Emergency Response: Being prepared to respond effectively to public health emergencies, epidemics, and disasters.

3.PROBLEM STATEMENT

Rural communities often face a significant disparity in accessing mental health services compared to urban areas. Limited availability of mental health professionals, long travel distances to clinics, and insufficient awareness contribute to the challenge. As a result, individuals in rural areas may experience delayed diagnosis, inadequate treatment, and increased stigma surrounding mental health. This situation not only impacts the well-being of residents but also places an additional burden on primary care providers. Addressing the barriers to mental health service access in rural communities is essential to ensure equitable healthcare delivery and improve overall community mental health.

4.SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements:

1. **Operating System:**

The server-side should run on a stable and secure operating system. Linux distributions like Ubuntu Server, CentOS, or Red Hat are commonly used.

Client-side applications can be designed to be platform-independent, accessible through web browsers. Ensure compatibility with major browsers like Chrome, Firefox, Safari, and Edge.

1. **Web Browser:**

Specify the compatible web browsers if the software is a web application (e.g., Chrome, Firefox, Safari, Edge).

3.Database Management System (DBMS):

Identify the supported database systems if the application relies on a database (e.g., MySQL, PostgreSQL, MongoDB).

4.Framework and Runtimes:

If applicable, list any specific frameworks (e.g., .NET, Django, React) or runtime environments needed.

5.Minimum Software Versions:

Define the minimum versions of required software components to ensure compatibility.

HARDWARE REQUIREMENTS:

1.Processor (CPU):

Specify the minimum and recommended CPU specifications, including the type and speed.

2.Memory (RAM):

Define the minimum and recommended RAM requirements for optimal performance.

3.Storage:

Indicate the minimum available storage space needed for installation and data storage.

4.Graphics and Display:

Specify any graphics card requirements, screen resolution, or color depth specifications.

5.Network Requirements: Outline any specific network requirements, such as minimum internet speed or network protocols.

5. PROJECT DESCRIPTION

The Comprehensive Health Care Management System is an advanced software solution designed for streamlined healthcare delivery. Featuring modules for patient management, appointment scheduling, medical staff coordination, billing, and reporting, it aims to enhance patient care and operational efficiency. Utilizing technologies such as React.js, Node.js, and MongoDB, the system prioritizes data security and compliance

Targeted at hospitals and clinics, its goals include improving patient experiences and optimizing resource allocation. The estimated development time is 6-9 months, and the system strives to revolutionize healthcare administration through technology-driven solutions.

Overview:

The Comprehensive Health Care Management System is a sophisticated software solution designed to streamline and enhance healthcare delivery. This system integrates various modules to manage patient records, facilitate medical staff coordination, and improve overall efficiency in healthcare processes.

Key Features:

Patient Management:

Centralized electronic health records (EHR) to store and manage patient information securely.

Patient registration, history tracking, and appointment scheduling for improved patient care.

Appointment Scheduling:

Intuitive scheduling system to efficiently manage appointments and reduce waiting times.

Automated appointment reminders for patients and healthcare providers.

Medical Staff Coordination:

Role-based access control to ensure data security and privacy compliance.

Communication tools for seamless collaboration among healthcare professionals.

Billing and Insurance:

Automated billing and invoicing system for accurate and timely financial transactions.

Integration with insurance providers to streamline claims processing.

Prescription Management:

Digital prescription capabilities for healthcare providers.

Medication tracking and reminders for patients.

Reporting and Analytics:

Comprehensive reporting tools for healthcare administrators to monitor system performance and analyze patient data.

Data analytics for identifying trends, optimizing resource allocation, and improving decision-making.

Technologies Used:

Frontend: React.js for a responsive and user-friendly interface.

Backend: Node.js with Express for a robust and scalable server.

Database: MongoDB for efficient and secure data storage.

Security: SSL encryption, role-based access control, and regular security audits.

Target Audience:

Hospitals, clinics, and healthcare facilities looking to modernize and optimize their healthcare management processes.

Goals:

Improve patient care and experience.

Enhance operational efficiency for healthcare providers.

Ensure data security and compliance with healthcare regulations.

6.FLOWCHART/ALGORITHM/PROCEDURE

Creating a flowchart, algorithm, or procedure for a comprehensive health care management system involves breaking down the processes and interactions within the system

Start:

Triggered by a patient's request for an appointment.

User Input:

Collect patient details and appointment preferences.

Availability Check:

Check the availability of the preferred healthcare provider and time slot.

Appointment Confirmation:

If available, confirm the appointment and notify the patient.

Notification:

Send automated reminders to patients closer to the appointment date.

Medical Staff Coordination:

Notify the healthcare provider and update their schedule.

Patient Arrival:

Mark the patient as arrived upon check-in.

Consultation:

Document the consultation details and update the patient's electronic health records.

Billing:

Generate bills based on services provided and update financial records.

End:

Conclude the appointment and update the system.

7. CODE

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
public class HealthcareSystem {
 private List<Patient> patients;
 public HealthcareSystem() {
 this.patients = new ArrayList<>();
 }
 public void registerPatient(Patient patient) {
 patients.add(patient);
 System.out.println("Patient registered successfully!");
 }
 public void displayPatients() {
 for (Patient patient : patients) {
 System.out.println(patient);
 System.out.println("----------------");
 }
 }
 public static void main(String[] args) {
 HealthcareSystem healthcareSystem = new HealthcareSystem();
 Scanner scanner = new Scanner(System.in);
6
 System.out.println("Healthcare System");
 System.out.println("------------------");
 while (true) {
```

```
System.out.println("1. Register Patient");
System.out.println("2. Display Patients");
System.out.println("3. Exit");
System.out.print("Enter your choice: ");
int choice = scanner.nextInt();
scanner.nextLine(); // Consume the newline character
switch (choice) {
case 1:
System.out.print("Enter Patient ID: ");
String patientId = scanner.nextLine();
System.out.print("Enter Patient Name: ");
String name = scanner.nextLine();
System.out.print("Enter Patient Address: ");
String address = scanner.nextLine();
System.out.print("Enter Patient Phone Number: ");
String phoneNumber = scanner.nextLine();
Patient patient = new Patient(patientId, name, address,
phoneNumber);
healthcareSystem.registerPatient(patient);
7
break;
case 2:
healthcareSystem.displayPatients();
break;
case 3:
System.out.println("Exiting the Healthcare System. Thankyou!");
System.exit(0);
```

```
break;
 default:
 System.out.println("Invalid choice. Please enter a valid
option.");
 }
 }
 }
}
```

8.RESULTS

OUTPUT:

```
Healthcare System
 ----------
1. Register Patient
2. Display Patients
3. Exit
Enter your choice:
1 Enter Patient ID: 7237
Enter Patient Name: Nanda Padma
Enter Patient Address: Shadhnagar
Enter Patient Phone Number: 7702799789
Patient registered successfully!
1. Register Patient
2. Display Patients
3. Exit
Enter your choice:
1 Enter Patient ID: 7252
```

Enter Patient Name: Bachu Sushmitha

Enter Patient Address: Kalvakurthy

Enter Patient Phone Number: 9345681976

Patient registered successfully!

1. Register Patient

2. Display Patients

3. Exit

Enter your choice:

1 Enter Patient ID: 7255

Enter Name: Gudise Tabitha

Enter Address: Hyderabad

Enter Phone Number: 7354897150

Patient registered successfully!

---

1. Register Patient

2. Display Patients

3. Exit

Enter your choice:

2 Patient ID: 7237

Patient Name: Nanda Padma

Address: Shadhnagar

Patient Phone Number: 7702799789

Patient ID: 7252

Patient Name: Bachu Sushmitha

Patient Address: Kalvakurthy

Patient Phone Number: 9345681976

Patient ID: 7255

Name: Gudise Tabitha

Address: Hyderabad

Phone Number: 7354897150

1. Register Patient 10

1. Register Patient

2.Display Patients

3. Exit

Enter your choice:

1 Exiting the Healthcare System.

Thankyou.

Screenshots

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class HealthcareSystem {
    private List<Patient> patients;

    public HealthcareSystem() {
        this.patients = new ArrayList<>();
    }

    public void registerPatient(Patient patient) {
        patients.add(patient);
        System.out.println("Patient registered successfully!");
    }

    public void displayPatients() {
        for (Patient patient : patients) {
            System.out.println(patient);
            System.out.println("---------------");
```

```
    }
}

public static void main(String[] args) {
    HealthcareSystem healthcareSystem = new HealthcareSystem();
    Scanner scanner = new Scanner(System.in);
    System.out.println("Healthcare System");
    System.out.println("-----------------");
    while (true) {
        System.out.println("1. Register Patient");
        System.out.println("2. Display Patients");
        System.out.println("3. Exit");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character
        switch (choice) {
            case 1:
                System.out.print("Enter Patient ID: ");
                String patientId = scanner.nextLine();
```

```
                System.out.print("Enter Patient Name: ");
                String name = scanner.nextLine();
                System.out.print("Enter Patient Address: ");
                String address = scanner.nextLine();
                System.out.print("Enter Patient Phone Number: "
                    );
                String phoneNumber = scanner.nextLine();
                Patient patient = new Patient(patientId, name,
                    address, phoneNumber);
                healthcareSystem.registerPatient(patient);
                break;
            case 2:
                healthcareSystem.displayPatients();
                break;
            case 3:
                System.out.println("Exiting the Healthcare
                    System. Thankyou!");
                System.exit(0);
                break;
            default:
```

```
                    System.out.println("Invalid choice. Please enter
                        a valid option.");
                }
            }
        }
    }

class Patient {
    private String patientId;
    private String name;
    private String address;
    private String phoneNumber;

    public Patient(String patientId, String name, String address,
        String phoneNumber) {
        this.patientId = patientId;
        this.name = name;
        this.address = address;
        this.phoneNumber = phoneNumber;
    }
```

```
public String getPatientId() {
    return patientId;
}

public String getName() {
    return name;
}

public String getAddress() {
    return address;
}

public String getPhoneNumber() {
    return phoneNumber;
}

@Override
public String toString() {
    return "Patient ID: " + patientId + ", Name: " + name + ",
```

```
        return "Patient ID: " + patientId + ", Name: " + name + ",
                Address: " + address + ", Phone Number: " + phoneNumber;
    }
}
```

```
java -cp /tmp/Or9DObctvM HealthcareSystem
Healthcare System
-----------------
1. Register Patient
2. Display Patients
3. Exit
Enter your choice: 1
Enter Patient ID: 424
Enter Patient Name: vysh
Enter Patient Address: khmanpur
Enter Patient Phone Number: 9848656198
9848656198
Patient registered successfully!
1. Register Patient
2. Display Patients
3. Exit
Enter your choice: 2
Patient ID: 424, Name: vysh, Address: kmanpur, Phone Number: 9848656198
---------------
1. Register Patient
```

```
9848656198
Patient registered successfully!
1. Register Patient
2. Display Patients
3. Exit
Enter your choice: 2
Patient ID: 424, Name: vysh, Address: kmanpur, Phone Number: 9848656198
---------------
1. Register Patient
2. Display Patients
3. Exit
Enter your choice: 3
Exiting the Healthcare System. Thankyou!
```

9.CONCLUSION:

In conclusion, a robust and accessible healthcare system is essential for the well-being of individuals and society as a whole. It requires a balance between preventive care, efficient diagnostics, and comprehensive treatment options. Adequate investment, technological advancements, and a focus on patient-centered care are pivotal for the continuous improvement of healthcare systems globally. Additionally, fostering collaboration among stakeholders and addressing disparities in healthcare access will contribute to creating a more equitable and sustainable healthcare future.

10.REFERENCES:

https://www.javatpoint.com/health-care-system

https://www.geeksforgeeks.org/health-care-system-documentation/