E-HEALTH MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

KARTHIKEYAN D(2303811710421074)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

DECEMBER- 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"E-Health Management System"** is the bonafide work of **KARTHIKEYAN D(2303811710421074)**who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Mrs.A.Delphin Carolina Rani, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

SIGNATURE

Mrs.K.VALLI PRIYADHARSHINI, M.E., Ph.D.,),

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on …3/12/2024

INTERNAL EXAMINER

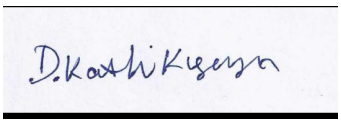
EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **"E-HEALTH MANAGEMENT SYSTEM"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING.**

.

Signature



KARTHIKEYAN D

Place: Samayapuram

Date:3/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **MR. A. MALARMANNAN, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid

conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The E-Health Management System is a comprehensive software solution designed to facilitate the management of healthcare data in a digital format. This system aims to streamline the operations of healthcare providers, improve patient care, and ensure efficient administration of healthcare services. It is developed using Java programming language, which takes advantage of its object-oriented principles, platform independence, and vast library support.

The core functionalities of the E-Health Management System include patient registration, appointment scheduling, medical history tracking, and electronic prescription generation. The system also allows healthcare providers to access patient records, update treatment plans, and manage patient billing information. The application features secure login and role-based access control, ensuring that sensitive patient data is accessible only to authorized personnel.

The e-health Management System provides a scalable, reliable, and secure solution for managing healthcare operations, improving the quality of care, and promoting efficient use of healthcare resources. Using Java ensures portability across platforms and enhances the system's maintainability. The E-Health Management System in Java is a versatile, robust solution that addresses the needs of modern healthcare providers.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
A digital platform that leverages information technology to streamline healthcare processes, enabling secure access to patient medical records, appointment scheduling, telehealth consultations, and data analytics, ultimately improving patient care by facilitating efficient communication between healthcare providers. Electronic health records (EHR), patient portal, appointment scheduling, remote monitoring, prescription management, billing and payment processing.	PO1 -3 PO2 -3 PO3 -3 PO4 -3 PO5 -3 PO6 -3 PO7 -3 PO8 -3 PO9 -3 PO10 -3 PO11-3 PO12 -3	PSO1 -3 PSO2 -3 PSO3 -3

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of the E-Health Management System is to create a comprehensive, user-friendly, and secure platform that revolutionizes the management of healthcare services. The system aims to streamline patient records by maintaining accurate, centralized, and up-to-date data, including demographics, medical history, and treatment details, accessible only to authorized personnel. It facilitates seamless appointment scheduling, allowing patients to book, view, and manage appointments with healthcare providers efficiently, thus reducing waiting times and improving time management. By fostering enhanced communication between patients, healthcare providers, and administrative staff, the system ensures better coordination and improved service delivery.

1.2 Overview

The E-Health Management System is an innovative solution designed to transform healthcare operations by integrating technology into patient care and administrative processes. It serves as a centralized platform that manages patient information, appointment scheduling, and medical services, enabling seamless interaction between patients, healthcare providers, and administrative staff. The system ensures secure and role-based access, allowing patients to view their medical history, book appointments, and access health records while empowering staff and administrators to manage records efficiently

1.3 Java Programming Concepts

1. **Exception Handling**: Java's robust exception handling mechanism was essential to prevent the application from crashing due to unexpected errors. The try-catch blocks were used to handle errors gracefully, such as invalid user input, database connection issues, or incorrect appointment dates. Custom error messages guided the users to fix issues, ensuring a smooth and uninterrupted experience. This concept was crucial for maintaining the integrity and stability of the system.
2. **Object-Oriented Programming (OOP)**: Object-Oriented Programming (OOP) is the foundation of the system's design. Key principles such as Encapsulation, Inheritance, Polymorphism, and Abstraction were employed to structure the application. These principles allowed the creation of reusable, modular, and

maintainable code. Encapsulation ensured that sensitive data, such as patient information, was securely managed, while inheritance allowed for efficient code reuse, particularly in the different user roles like Admin, Staff

3. **Multithreading:**

   Java's multithreading capabilities were employed to allow multiple tasks to run concurrently, ensuring the system remained responsive. Tasks such as appointment scheduling, sending notifications, and background data processing could run in parallel without blocking the main user interface. This was crucial for maintaining a smooth user experience, especially in an environment where real-time updates and responsiveness are essential.

4. **File Handling (I/O)** Java's File I/O functionality was used to read and write data to text files or databases, ensuring that important data such as user profiles and appointment histories were stored persistently. BufferedReader and FileWriter were used for efficient reading and writing, while serialization was employed for saving and retrieving complex objects like patient records. This concept was key for enabling data persistence in the application.

5. **Collections Framework:** The Java Collections Framework was used to manage large datasets efficiently, such as patient records and appointment details. ArrayList, HashMap, and HashSet were used for dynamic data storage and retrieval. The HashMap helped in managing role-based access control, enabling quick lookups of user roles and corresponding permissions, while ArrayList was used to store and manipulate collections of patient or appointment data.

CHAPTER 2
PROJECT METHODOLOGY

2.1 Proposed Work

The proposed E-Health Management System is designed to enhance healthcare management through the use of technology, offering a user-friendly, secure, and efficient platform for managing patient records, appointments, and other healthcare services. The system aims to streamline the administrative and clinical workflows for healthcare providers, ensuring that medical information is organized, accessible, and easily managed.

The system will be developed with several core objectives:

2.1.1 User Management: The platform will include robust user management features to ensure secure access based on different roles, such as Admin, Staff, and Patients. Each user will have customized access to the system depending on their role, ensuring that sensitive medical data is only accessible to authorized personnel.

2.1.2 Patient Information Management: A centralized database will store patient information, including personal details, medical history, and treatment plans. This information will be accessible to both medical staff and the patients themselves, allowing for better collaboration and improved patient care.

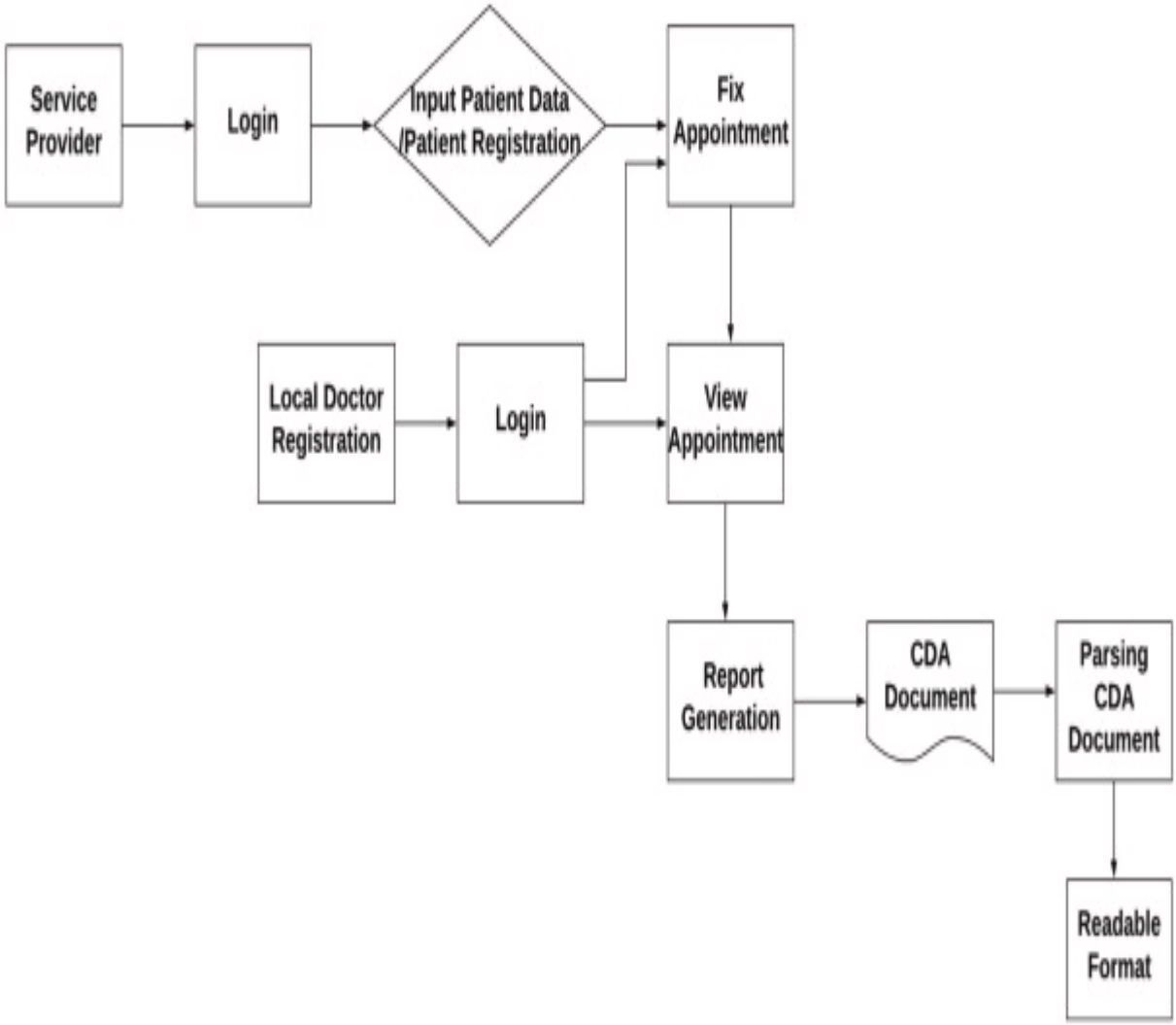
2.2.3 Appointment Scheduling: The system will enable patients to book appointments online, and healthcare providers can view, manage, and update appointments easily. The scheduling system will also include automated reminders to help minimize missed appointments and optimize the clinic's schedule.

2.2.4 Security and Data Privacy: Ensuring the privacy and security of sensitive data will be a top priority. The system will implement encryption protocols, user authentication, and secure data transmission to protect patient and healthcare provider information.

2.2.5 Reporting and Analytics: The platform will feature reporting tools to track patient visits, treatment outcomes, and staff performance. These reports will help healthcare administrators make informed decisions about resource allocation and service improvements.

By integrating these functionalities, the E-Health Management System aims to create a seamless, secure, and efficient environment for managing healthcare services.

2.2 Block Diagram



CHAPTER 3
MODULE DESCRIPTION

The E-Health Management System consists of various modules, each designed to address specific aspects of the application, ensuring efficient healthcare management. Below is an overview of each module, highlighting its unique features.

3.1 User Management Module: This module is the backbone of the system, enabling secure access and efficient role-based management of users. The login functionality ensures the system verifies user credentials securely, utilizing robust password authentication mechanisms. Through role-based access, it differentiates between Admin, Staff, and Patient roles, providing varying levels of system access tailored to their responsibilities. The account creation process allows new users to register by entering necessary details, including username, password, and role, with optional fields for additional information. It incorporates error-handling measures to ensure valid input and maintain unique usernames. Additionally, the module emphasizes reporting and feedback, displaying success messages for actions like account creation and login, while clearly communicating errors such as invalid credentials or duplicate usernames. By integrating role-specific workflows and stringent security protocols, this module guarantees a user-centric experience while maintaining the integrity of the system.

3.2 Patient Management Module : This module handles the efficient management of patient information, ensuring seamless integration of personal and medical details into the system. Through its patient records management, it facilitates the addition of patient data during account creation, including critical details such as age, gender, and medical history. The module enables viewing and searching patient details, allowing authorized users like Admin and Staff to access comprehensive patient information quickly. A search functionality is implemented to locate patient records by name, providing instant retrieval of relevant data. With its intuitive design and dynamic data management, this module ensures that all patientrelated information is well-organized, easily accessible, and secure, fostering effective healthcare delivery.

3.3 Appointment Management Module : This module simplifies the process of booking and managing appointments, creating a seamless experience for both patients and healthcare providers. It supports appointment booking, enabling patients to select a preferred doctor and schedule appointments by specifying a suitable date. The module

ensures proper date validation, using reliable formats to prevent scheduling errors.

3.3.1 Appointment booking is facilitated by selecting a doctor and specifying a date.

3.3.2 Date format validation ensures appointments are scheduled correctly.

3.3.3 Admin and staff can view all appointments, while patients can access their own appointment details.

3.3.4 ArrayList is used to dynamically manage appointment records, with date validation handled through SimpleDateFormat

3. 4 User Interface Module (GUI) This module offers an intuitive and interactive graphical interface for users to engage with the system effectively. Designed using Java Swing, it provides visually appealing and functional interfaces for operations such as login, account creation, and role-specific tasks. Through dynamic input dialogs, users can perform actions like booking appointments and adding patient details seamlessly. Navigation between features is facilitated through menu-driven options tailored to the user's role, ensuring a streamlined workflow. This module prioritizes user experience by presenting clear feedback and ensuring all functionalities are easily accessible, making the system user-friendly and efficient for all roles.

3.5 Validation and Error Handling Module

This module is integral to maintaining the stability, security, and accuracy of the system, addressing errors proactively and ensuring consistent data integrity. Through centralized data validation, all system inputs such as user roles, patient details, and appointment dates are rigorously verified to prevent inaccuracies. It incorporates error prevention mechanisms, restricting invalid actions based on user roles, thereby ensuring that only authorized operations are executed. Logical workflows are integrated seamlessly, enabling users to perform permitted actions while mitigating the risk of unintended errors. By validating data at every step and providing clear feedback through error messages or success confirmations, this module plays a crucial role in enhancing user experience and maintaining the overall reliability of the system. Its integration with core application logic ensures a smooth and secure operation, making it an indispensable part of the E-Health Management System.

3.5.1 Password validation is used during login to ensure security.

3.5.2 Input validation is performed for patient details, appointment dates, and role-specific actions

3.5.3 Confirmation messages are displayed for successful account creation and appointment bookings

. 3.5.4 Error messages are shown for failed operations, such as incorrect login credentials or invalid date formats

CHAPTER 4
CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The system's centralized approach to data validation and error handling enhances its reliability and stability, minimizing human error and ensuring data integrity throughout. These features guarantee that users are limited to performing actions permitted for their roles, thus reducing the risk of invalid data or incorrect actions. As healthcare demands grow, the scalability of this system ensures it remains effective in managing larger datasets and more complex workflows.

The E-Health Management System provides a robust solution to streamline healthcare operations, ensuring secure, efficient, and user-friendly management of essential healthcare processes. By integrating modules like User Management, Patient Record Handling, Appointment Scheduling, and Data Validation, the system offers a seamless experience for administrators, healthcare professionals, and patients alike. With role-based access, each user can securely interact with the system according to their needs, ensuring that sensitive data is protected and workflows are efficiently managed.

4.2 FUTURE SCOPE

The future of the E-Health Management System offers significant opportunities for innovation and growth. Key advancements could include the integration of cloud computing, allowing for real-time synchronization across multiple locations and devices. This would make healthcare more accessible by enabling remote consultations and telemedicine services, reducing the need for in-person visits and improving patient engagement. Another promising avenue for the system’s future development is the incorporation of Artificial Intelligence (AI) and machine learning algorithms. These technologies could support predictive analytics for better health management, such as anticipating patient needs or diagnosing conditions more accurately. AI-powered chatbots could also streamline the appointment scheduling process, enhancing user convenience and reducing administrative burdens.

On the security front, advanced encryption protocols and multi-factor authentication could bolster the protection of patient data, ensuring the system remains resilient against cyber threats. As technology continues to evolve

APPENDIX A
(SOURCE CODE)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;

// Class to represent a patient
class Patient {
    String name;
    int age;
    String contactNumber;
    String ailment;

    public Patient(String name, int age, String contactNumber, String ailment) {
        this.name = name;
        this.age = age;
        this.contactNumber = contactNumber;
        this.ailment = ailment;
    }

    @Override
    public String toString() {
        return "Name: " + name + ", Age: " + age + ", Contact: " + contactNumber + ", Ailment: "
+ ailment;
    }
}

// Main class
public class eHealthManagementSystem extends JFrame {
    private ArrayList<Patient> patientRecords = new ArrayList<>();
    private DefaultListModel<String> patientListModel = new DefaultListModel<>();
    private JList<String> patientList = new JList<>(patientListModel);
```

```
public eHealthManagementSystem() {
    setTitle("eHealth Management System");
    setSize(600, 400);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setLocationRelativeTo(null);

    // Layout
    setLayout(new BorderLayout());

    // Patient Records Panel
    JPanel patientPanel = new JPanel(new BorderLayout());
    patientPanel.setBorder(BorderFactory.createTitledBorder("Patient Records"));
    JScrollPane scrollPane = new JScrollPane(patientList);
    patientList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
    patientPanel.add(scrollPane, BorderLayout.CENTER);

    // Buttons Panel
    JPanel buttonPanel = new JPanel();
    JButton addPatientButton = new JButton("Add Patient");
    JButton viewPatientsButton = new JButton("View Patients");
    JButton bookAppointmentButton = new JButton("Book Appointment");
    JButton exitButton = new JButton("Exit");

    buttonPanel.add(addPatientButton);
    buttonPanel.add(viewPatientsButton);
    buttonPanel.add(bookAppointmentButton);
    buttonPanel.add(exitButton);

    add(patientPanel, BorderLayout.CENTER);
    add(buttonPanel, BorderLayout.SOUTH);

    // Button Actions
    addPatientButton.addActionListener(e -> addPatientRecord());
```

```
        viewPatientsButton.addActionListener(e -> viewPatientRecords());
        bookAppointmentButton.addActionListener(e -> bookAppointment());
        exitButton.addActionListener(e -> System.exit(0));

        setVisible(true);
    }

    // Method to add a patient record
    private void addPatientRecord() {
        JTextField nameField = new JTextField();
        JTextField ageField = new JTextField();
        JTextField contactField = new JTextField();
        JTextField ailmentField = new JTextField();

        JPanel inputPanel = new JPanel(new GridLayout(4, 2));
        inputPanel.add(new JLabel("Name:"));
        inputPanel.add(nameField);
        inputPanel.add(new JLabel("Age:"));
        inputPanel.add(ageField);
        inputPanel.add(new JLabel("Contact Number:"));
        inputPanel.add(contactField);
        inputPanel.add(new JLabel("Ailment:"));
        inputPanel.add(ailmentField);

        int result = JOptionPane.showConfirmDialog(this, inputPanel, "Add Patient Record",
JOptionPane.OK_CANCEL_OPTION);
        if (result == JOptionPane.OK_OPTION) {
            try {
                String name = nameField.getText().trim();
                int age = Integer.parseInt(ageField.getText().trim());
                String contact = contactField.getText().trim();
                String ailment = ailmentField.getText().trim();

                if (!name.isEmpty() && !contact.isEmpty() && !ailment.isEmpty()) {
```

```
                Patient patient = new Patient(name, age, contact, ailment);
                patientRecords.add(patient);
                patientListModel.addElement(patient.toString());
                JOptionPane.showMessageDialog(this, "Patient record added successfully!");
            } else {
                JOptionPane.showMessageDialog(this, "All fields must be filled!", "Error", JOptionPane.ERROR_MESSAGE);
            }
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Invalid age! Please enter a number.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
  }

  // Method to view all patient records
  private void viewPatientRecords() {
    if (patientRecords.isEmpty()) {
      JOptionPane.showMessageDialog(this, "No patient records found.", "Info", JOptionPane.INFORMATION_MESSAGE);
    } else {
      StringBuilder records = new StringBuilder("=== Patient Records ===\n");
      for (int i = 0; i < patientRecords.size(); i++) {
        records.append(i + 1).append(". ").append(patientRecords.get(i)).append("\n");
      }
      JOptionPane.showMessageDialog(this, records.toString(), "Patient Records", JOptionPane.INFORMATION_MESSAGE);
    }
  }

  // Method to book an appointment
  private void bookAppointment() {
    if (patientRecords.isEmpty()) {
      JOptionPane.showMessageDialog(this, "No patients available for appointment
```

```
booking.", "Info", JOptionPane.INFORMATION_MESSAGE);
        } else {
            int selectedIndex = patientList.getSelectedIndex();
            if (selectedIndex != -1) {
                Patient selectedPatient = patientRecords.get(selectedIndex);
                JOptionPane.showMessageDialog(this, "Appointment booked for " +
selectedPatient.name, "Appointment", JOptionPane.INFORMATION_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(this, "Please select a patient to book an
appointment.", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(eHealthManagementSystem::new);
    }
}
```

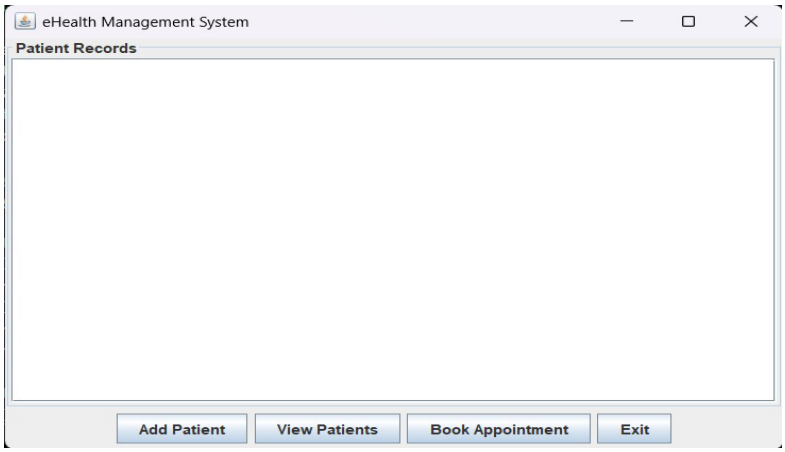
APPENDIX B

(SCREENSHOTS)

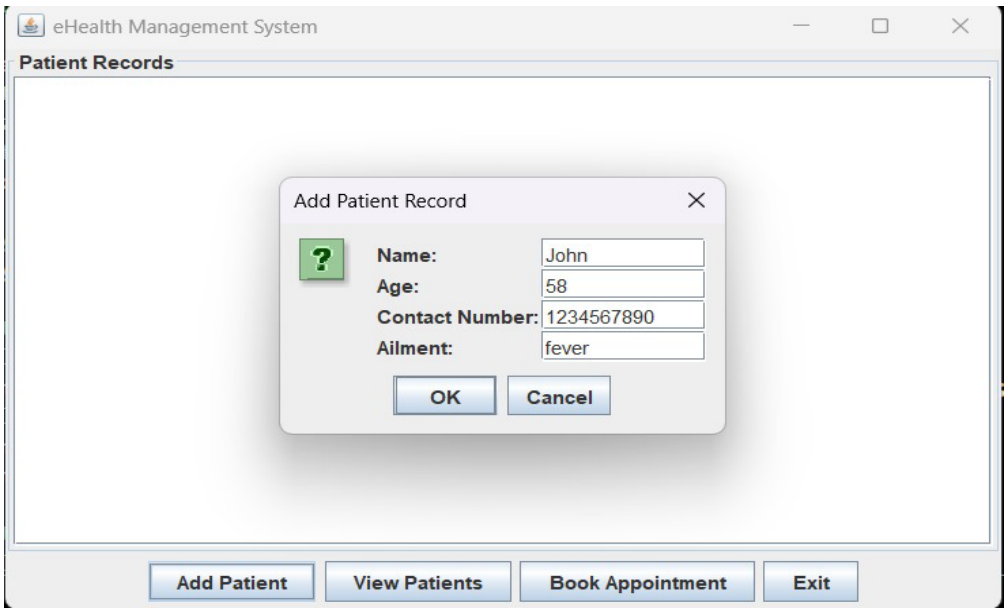
1. E- Health Management System Home page

1.1 patient records

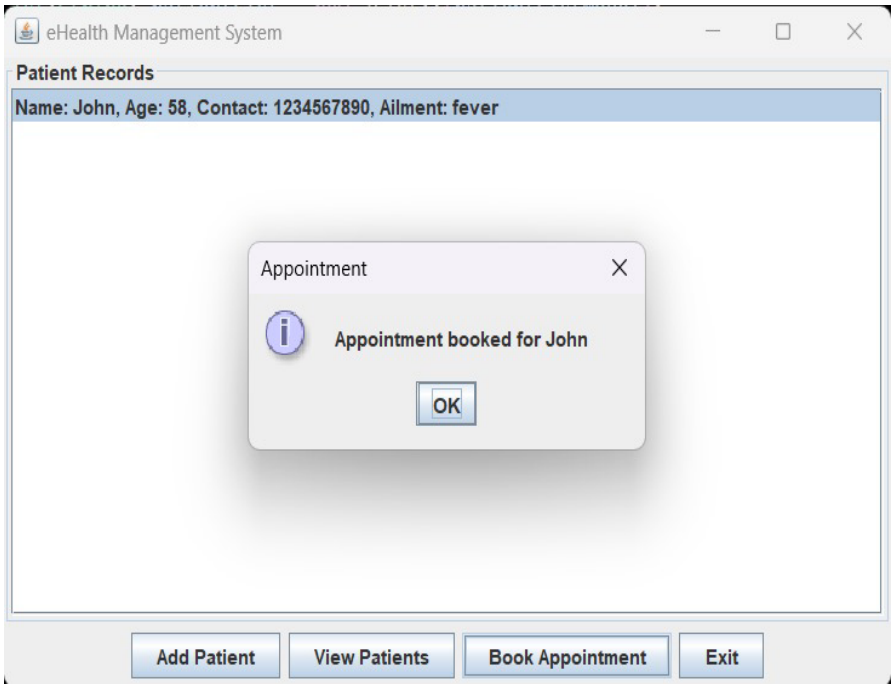
1.2 Create new patient Account



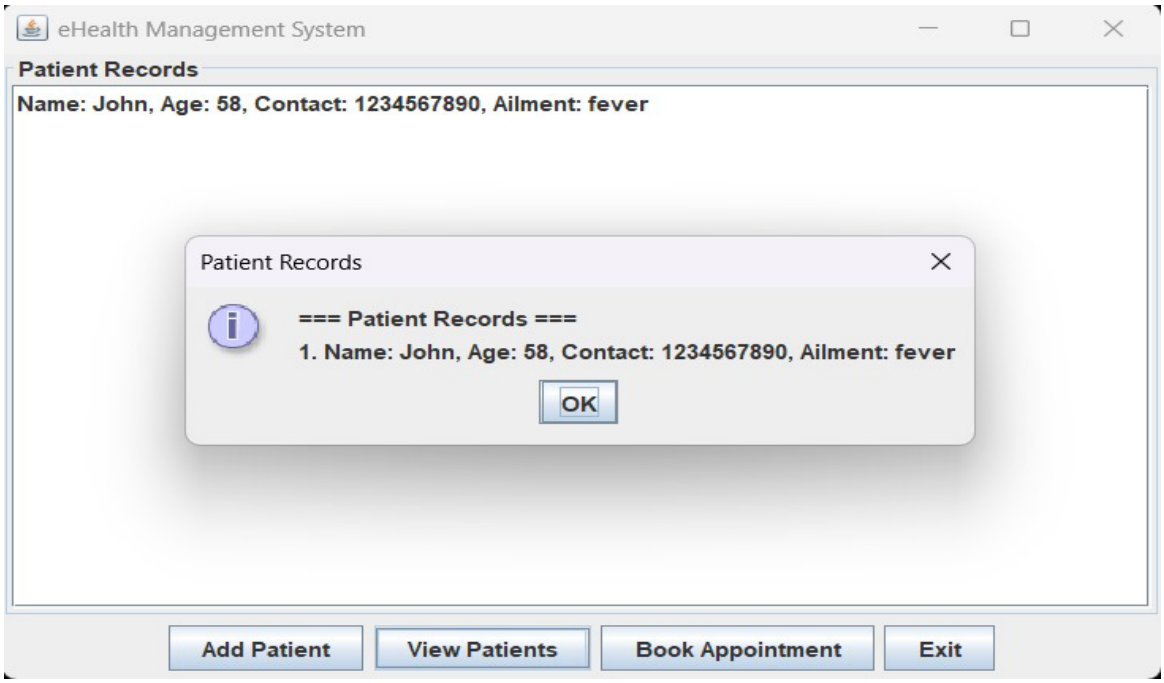
2. add patient record



3. Appointment Booking



4. patient record view



REFERENCES

1. Bashshur, R., Shannon, G., & Smith, B. (2014). "The Empirical Foundations of Telemedicine Interventions for Chronic Disease Management." Telemedicine and e-Health, 20(9), 769–799.
2. Pope, C., & Mays, N. (2021). "Reaching the Limits of eHealth: Critical Reflections on eHealth in Health Care Systems." Health Services Research and Policy, 26(2), 128-135.
3. Boulos, M. N. K., & Zhang, J. (2011). "eHealth Systems and Their Impact on Healthcare Delivery." Journal of the American Medical Informatics Association, 18(5), 537-543.