HOSPITAL MANAGEMENT SYSTEM

GROUP 9

GROUP MEMBERS

	NAME	MATRIC NUMBER
1.	IZNURIN FATIHAH BINTI MD FAIZAL	B23CS0041
2.	NUR AIMI AFIQAH BINTI RAWI	B23CS0065
3.	NUR BALQIS BATRISYIA BINTI SAIDIN SYAZLI	B23CS0066
4.	NUR BALQIS BINTI MOHD NASIR	B23CS0067

01

INTRODUCTION

The healthcare sector is important to society, and efficient administration of healthcare facilities is necessary to guarantee patient safety and enhance the efficiency and effectiveness of the operational processes. From patient registration to inventory management, the Hospital Management System (HMS) is a complete system that automates and optimizes many elements of hospital operations. The goal of this project is to improve the healthcare services by creating a stable and user-friendly hospital management system.

02

OBJECTIVES

1. To improve our healthcare system

2. To create a system that is user-friendly and well-manageable

3. To create a system for patient so that they can choose their doctor based on their preferences

03

OO CONCEPTS

ENCAPSULATION & DATA HIDING

```
public class Name {
    private String firstName;
    private String lastName;

    public Name(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }
```

ASSOCIATION : AGGREGATION

Person.java * | Patient.java * | Name.java | MedicalRecord.java | HospitalApp.java | Doctor.java | Bill.java | Appointment.java | Addr

```java
public class Patient extends Person {
    private String patientID;
    private int age;
    private String gender;
    private Address address;
    private MedicalRecord medicalRecord;


    public Patient(String firstName, String lastName, int phoneNum, int age, String gender,
                   Address address, MedicalRecord medicalRecord) {
        super(new Name(firstName, lastName), phoneNum);
        this.patientID = patientID;
        this.age = age;
        this.gender = gender;
        this.address = address;
        this.medicalRecord = medicalRecord;
    }
```

ASSOCIATION : COMPOSITION

```java
public class Person {
    private Name name;
    private String phoneNum;

    public Person(Name name, String phoneNum) {
        this.name = name;
        this.phoneNum = phoneNum;
    }
```

INHERITANCE

```java
public class Doctor extends Person {
    private String doctorID;
    private String specialization;

    public Doctor(Name name, String doctorID, int phoneNum,  String specialization) {
        super(name, phoneNum);
        this.doctorID = doctorID;
        this.specialization = specialization;
    }

    public Name getName(){
        return super.getName();
    }

    public String getDoctorId() {
         for (int i = 1; i <= 100; i++) {
             return "D00" + i;
         }
         return doctorID;
    }

    public String getSpecialization() {
        return specialization;
    }
```

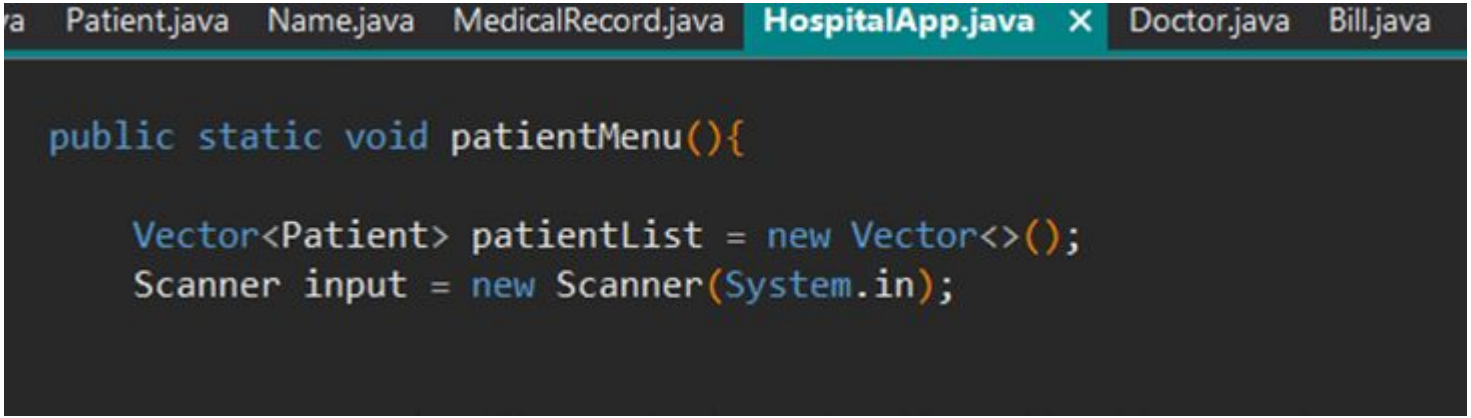
POLYMORPHISM

```
private static Doctor displayDoctorDetails(String doctorId) {
        Doctor d1 = new Doctor(new Name("Dr. Amin", "Afiq"), "D001", 13456789, "General doctor");
        Doctor d2 = new Doctor(new Name("Dr. Maria", "Noor"), "D002", 123456789, "Surgery");
        Doctor d3 = new Doctor(new Name("Dr. Aslam", "Ahmad"), "D003", 1452363887, "Cardiology");

         switch (doctorId) {
                    case "D001":
                        return d1;
                    case "D002":
                        return d2;
                    case "D003":
                        return d3;
                    default:
                        System.out.println("Doctor not found with ID: " + doctorId);
                        return null;
        }

    }
```

EXCEPTION HANDLING

```java
public class HospitalApp {
    public static void main(String[] args) throws IOException{
       Scanner input = new Scanner(System.in);
         try {

                    Scanner inp = new Scanner(new File("Doctors.txt"));

                    String name, phoneNum, specialization;

                    System.out.println("Welcome to Hospital Sultanah Ibrahim management");

                    // Print header
                    System.out.printf("%-18s%-15s%-2s\n", "Name", "Phone Number", "Specialization");

                    while (inp.hasNext()) {
                        name = inp.next();
                        phoneNum = inp.next();
                        specialization = inp.nextLine().trim();

                        // Print data with appropriate spacing
                        System.out.printf("%-9s%-8s%-18s\n", name, phoneNum, specialization);
                    }

                    inp.close(); // close the scanner after use
                } catch (IOException e) {
                    System.err.println("Error reading the file: " + e.getMessage());
                } catch (Exception e) {
                    System.err.println("An unexpected error occurred: " + e.getMessage());
        }
```

VECTOR


Patient.java
Name.java
MedicalRecord.java
HospitalApp.java
Doctor.java
Bill.java
public static void patientMenu(){
Vector<Patient> patientList = new Vector<>();
Scanner input = new Scanner(System.in);

INPUT FILE

```
public class HospitalApp {
    public static void main(String[] args) throws IOException{

        Scanner input = new Scanner(System.in);
        Scanner inp = new Scanner (new File("Doctors.txt"));

        String name, phoneNum,specialization;

        System.out.println("Welcome to Hospital Sultanah Ibrahim management");

        while(inp.hasNext()){
                name = inp.nextLine();
                phoneNum = inp.nextLine();
                specialization = inp.nextLine();

               System.out.println(""+ name+ phoneNum + specialization + "" );

    }
```

OUTPUT FILE

```
public static void appointmentMenu() throws FileNotFoundException{

    Vector<Appointment> appointmentList = new Vector<>();
    Scanner input = new Scanner(System.in);
    PrintWriter outputFile = new PrintWriter("Bill.txt");
```

OUTPUT FILE

```
        outputFile.printf("----------Receipt-----------");
        outputFile.printf("\nAppointment ID: " + a.getAppointmentID());
        outputFile.printf("\nAppointment Date: " + a.getDate());
        outputFile.printf("\nAppointment Time: " + a.getTime());
        outputFile.printf("\nPatient ID in Appointment: " + a.patientID());
        outputFile.printf("\nDoctor ID in Appointment: " + a.getDoctorId());




        outputFile.printf("\nBill ID: " + bill.getBillID());
        outputFile.printf("\nBill Amount: RM" + amount);
        outputFile.println();  // Separate each bill entry with a newline
    }

outputFile.close();
```

THAT’S ALL FROM US

THANK YOU ^^