HOSPITAL MANAGEMENT SYSTEM

GROUP 9
SECTION 01
PN. LIZAWATI BINTI MI YUSOF

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

PROJECT DESCRIPTION

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

To improve our healthcare system

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

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

03

PURPOSES

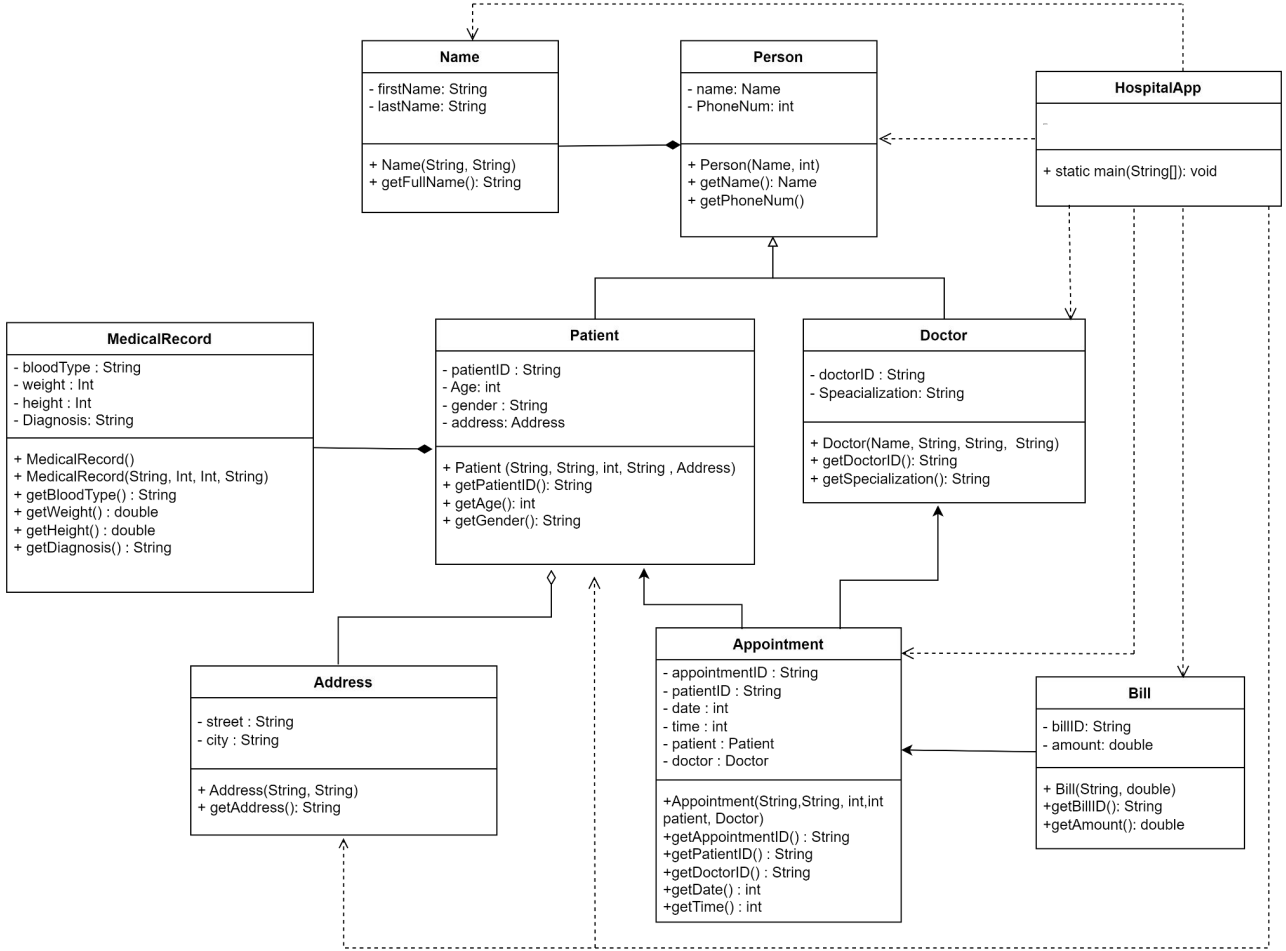
The purpose of creating a hospital management system is to transform the healthcare experience for both patients and providers, leading to improved care, efficiency and overall organizational success.

04

UML DIAGRAM

Link for for UML Diagram :

https://drive.google.com/file/d/1-Lzk2vX6afmUd7ya97RMJ1244n93caOZ/view?usp=sharing


Name
- firstName: String
- lastName: String
+ Name(String, String)
+ getFullName(): String
Person
- name: Name
- PhoneNum: int
+ Person(Name, int)
+ getName(): Name
+ getPhoneNum()
HospitalApp
+ static main(String[]): void
MedicalRecord
- bloodType : String
- weight : Int
- height : Int
- Diagnosis: String
+ MedicalRecord()
+ MedicalRecord(String, Int, Int, String)
+ getBloodType() : String
+ getWeight() : double
+ getHeight() : double
+ getDiagnosis() : String
Patient
- patientID : String
- Age: int
- gender : String
- address: Address
+ Patient (String, String, int, String , Address)
+ getPatientID(): String
+ getAge(): int
+ getGender(): String
Doctor
- doctorID : String
- Speacialization: String
+ Doctor(Name, String, String, String)
+ getDoctorID(): String
+ getSpecialization(): String
Address
- street : String
- city : String
+ Address(String, String)
+ getAddress(): String
Appointment
- appointmentID : String
- patientID : String
- date : int
- time : int
- patient : Patient
- doctor : Doctor
+Appointment(String,String, int,int patient, Doctor)
+getAppointmentID() : String
+getPatientID() : String
+getDoctorID() : String
+getDate() : int
+getTime() : int
Bill
- billID: String
- amount: double
+ Bill(String, double)
+getBillID(): String
+getAmount(): double

05

OO CONCEPTS

ENCAPSULATION & DATA HIDING

```
public class Patient extends Person {
    private static int lastPatientID = 0;
    private String patientID;
    private int age;
    private String gender;
    private Address address;
    private MedicalRecord medicalRecord;


    public Patient(String firstName, String lastName,  int age,String phoneNum, String gender, Address address,
        super(new Name(firstName, lastName), phoneNum);
          lastPatientID++;
        this.patientID = "P00" + lastPatientID;
        this.age = age;
        this.gender = gender;
        this.address = address;
        this.medicalRecord = medicalRecord;
    }


     public String getPatientID() {
           return patientID;
       }
```

AGGREGATION

```java
public class Patient extends Person {
    private static int lastPatientID = 0;
    private String patientID;
    private int age;
    private String gender;
    private Address address; //Aggregartion
    private MedicalRecord medicalRecord;


    public Patient(String firstName, String lastName,  int age,String phoneNum, String gender, Address address,
        super(new Name(firstName, lastName), phoneNum);
          lastPatientID++;
        this.patientID = "P00" + lastPatientID;
        this.age = age;
        this.gender = gender;
        this.address = address;
        this.medicalRecord = medicalRecord;
    }
```

COMPOSITION

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

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

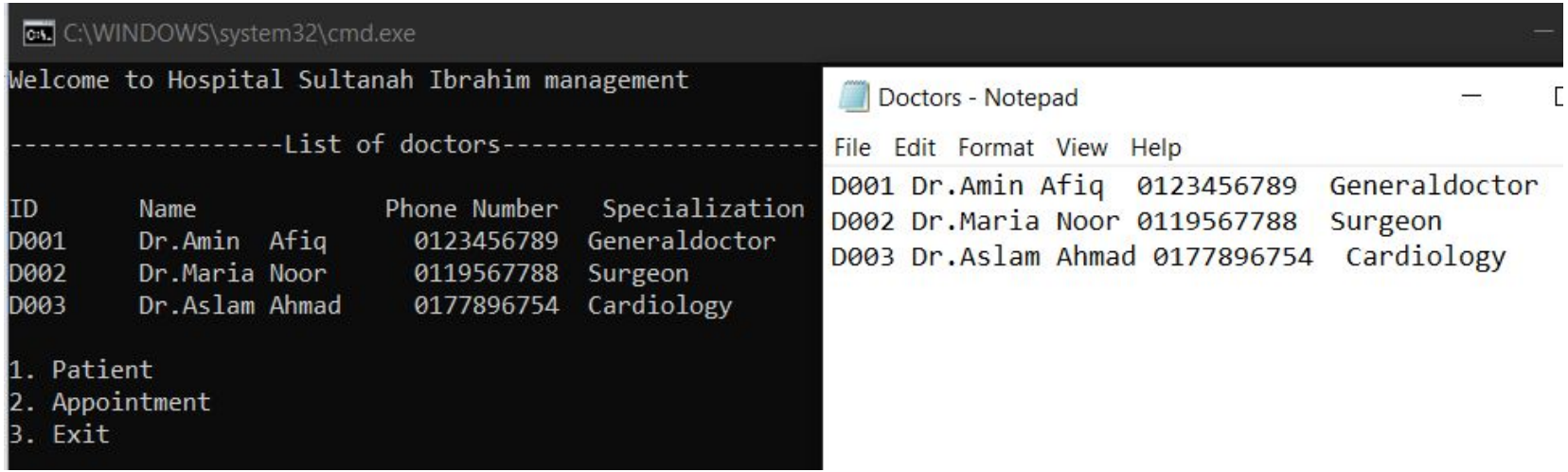
INHERITANCE

```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;
    }
```

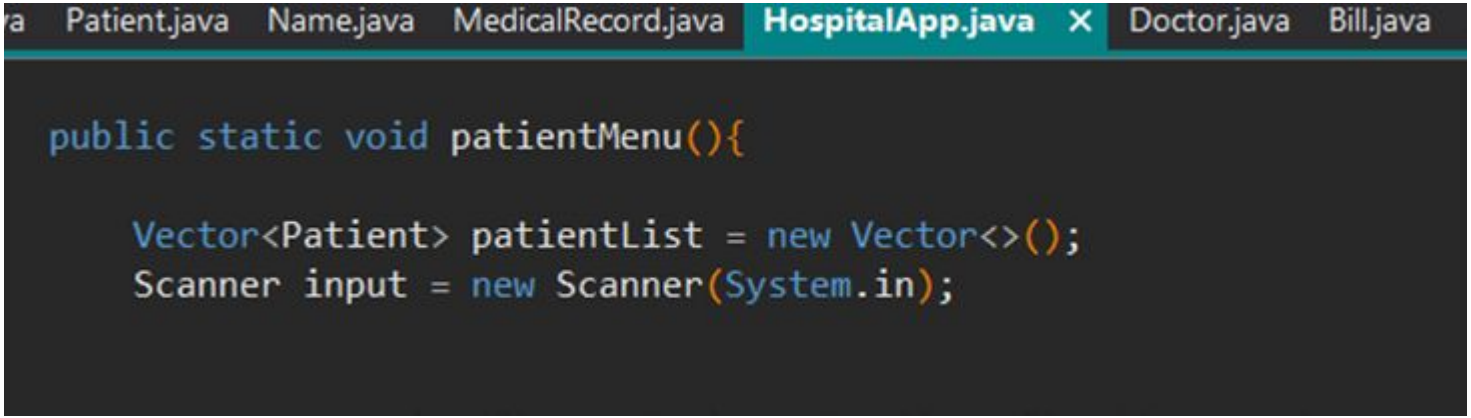
FILE INPUT

```
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 + "" );

    }
```

FILE INPUT


C:\WINDOWS\system32\cmd.exe
Welcome to Hospital Sultanah Ibrahim management
----------List of doctors----------
ID Name Phone Number Specialization
D001 Dr.Amin Afiq 0123456789 Generaldoctor
D002 Dr.Maria Noor 0119567788 Surgeon
D003 Dr.Aslam Ahmad 0177896754 Cardiology
1. Patient
2. Appointment
3. Exit
Doctors - Notepad
File Edit Format View Help
D001 Dr.Amin Afiq 0123456789 Generaldoctor
D002 Dr.Maria Noor 0119567788 Surgeon
D003 Dr.Aslam Ahmad 0177896754 Cardiology

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);

FILE OUTPUT

```
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();
```

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());
        }
```

THAT'S ALL FROM US
THANK YOU