package com.adm.project;

import java.util.ArrayList;

import java.util.Scanner;

// Class for storing patient details

class Patient {

String name, gender, disease;

int age, roomNo, ward;

String contactNumber;

double bill;

public Patient(String name, int age, String gender, String disease, String contactNumber, int roomNo, int ward) {

this.name = name;

this.age = age;

this.gender = gender;

this.disease = disease;

this.contactNumber = contactNumber;

this.roomNo = roomNo;

this.ward = ward;

this.bill = 0.0; // Initial bill is 0

}

public void setBill(double bill) {

this.bill = bill;

}

public String toString() {

return "Patient Name: " + name + ", Age: " + age + ", Gender: " + gender + ", Disease: " + disease + ", Room No: " + roomNo + ", Ward: " + ward + ", Contact: " + contactNumber;

}

}

// Class for storing doctor details

class Doctor {

String name, gender, specialization;

int age;

double salary;

public Doctor(String name, int age, String gender, String specialization, double salary) {

this.name = name;

this.age = age;

this.gender = gender;

this.specialization = specialization;

this.salary = salary;

}

public String toString() {

return "Doctor Name: " + name + ", Age: " + age + ", Gender: " + gender + ", Specialization: " + specialization + ", Salary: " + salary;

}

}

// Class for appointments

class Appointment {

Patient patient;

Doctor doctor;

public Appointment(Patient patient, Doctor doctor) {

this.patient = patient;

this.doctor = doctor;

}

public String toString() {

return "Appointment scheduled: Patient: " + patient.name + ", Doctor: " + doctor.name;

}

}

// Main Hospital Management class

public class HospitalManagementSystem {

static ArrayList<Patient> patients = new ArrayList<>();

static ArrayList<Doctor> doctors = new ArrayList<>();

static ArrayList<Appointment> appointments = new ArrayList<>();

static boolean[] rooms = new boolean[10]; // For room availability check (true=occupied, false=available)

// Add a new patient

public static void addPatient() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter patient name: ");

String name = sc.nextLine();

System.out.println("Enter age: ");

int age = sc.nextInt();

sc.nextLine(); // consume newline

System.out.println("Enter gender: ");

String gender = sc.nextLine();

System.out.println("Enter disease: ");

String disease = sc.nextLine();

System.out.println("Enter contact number: ");

String contactNumber = sc.nextLine();

// Ensure room number is within bounds and available

int roomNo;

do {

System.out.println("Enter room number (0-9): ");

roomNo = sc.nextInt();

if (roomNo < 0 || roomNo >= rooms.length) {

System.out.println("Invalid room number. Please enter a number between 0 and 9.");

} else if (rooms[roomNo]) {

System.out.println("Room is already occupied. Please choose a different room.");

}

} while (roomNo < 0 || roomNo >= rooms.length || rooms[roomNo]);

System.out.println("Enter ward number: ");

int ward = sc.nextInt();

patients.add(new Patient(name, age, gender, disease, contactNumber, roomNo, ward));

rooms[roomNo] = true; // Mark the room as occupied

System.out.println("Patient added successfully!");

}

// Add a new doctor

public static void addDoctor() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter doctor name: ");

String name = sc.nextLine();

System.out.println("Enter age: ");

int age = sc.nextInt();

sc.nextLine(); // consume newline

System.out.println("Enter gender: ");

String gender = sc.nextLine();

System.out.println("Enter specialization: ");

String specialization = sc.nextLine();

System.out.println("Enter salary: ");

double salary = sc.nextDouble();

doctors.add(new Doctor(name, age, gender, specialization, salary));

System.out.println("Doctor added successfully!");

}

// Add billing details for a patient

public static void addBilling() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter patient name: ");

String name = sc.nextLine();

System.out.println("Enter bill amount: ");

double bill = sc.nextDouble();

for (Patient p : patients) {

if (p.name.equalsIgnoreCase(name)) {

p.setBill(bill);

System.out.println("Billing updated for patient: " + name);

return;

}

}

System.out.println("Patient not found!");

}

// Show available rooms

public static void showAvailableRooms() {

System.out.println("Available rooms:");

for (int i = 0; i < rooms.length; i++) {

if (!rooms[i]) {

System.out.println("Room " + i + " is available.");

}

}

}

// Show all patient details

public static void showAllPatients() {

for (Patient p : patients) {

System.out.println(p);

}

}

// Show all doctor details

public static void showAllDoctors() {

for (Doctor d : doctors) {

System.out.println(d);

}

}

// Schedule an appointment

public static void scheduleAppointment() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter patient name: ");

String patientName = sc.nextLine();

System.out.println("Enter doctor name: ");

String doctorName = sc.nextLine();

Patient selectedPatient = null;

Doctor selectedDoctor = null;

for (Patient p : patients) {

if (p.name.equalsIgnoreCase(patientName)) {

selectedPatient = p;

break;

}

}

for (Doctor d : doctors) {

if (d.name.equalsIgnoreCase(doctorName)) {

selectedDoctor = d;

break;

}

}

if (selectedPatient != null && selectedDoctor != null) {

appointments.add(new Appointment(selectedPatient, selectedDoctor));

System.out.println("Appointment scheduled successfully!");

} else {

System.out.println("Either doctor or patient not found!");

}

}

// Display all appointments

public static void showAppointments() {

for (Appointment ap : appointments) {

System.out.println(ap);

}

}

// Show patient details with bill when discharged

public static void dischargePatient() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter patient name to discharge: ");

String name = sc.nextLine();

Patient patientToDischarge = null;

for (Patient p : patients) {

if (p.name.equalsIgnoreCase(name)) {

patientToDischarge = p;

rooms[p.roomNo] = false; // Free the room

break;

}

}

if (patientToDischarge != null) {

System.out.println(patientToDischarge + ", Bill: " + patientToDischarge.bill);

patients.remove(patientToDischarge);

System.out.println("Patient discharged successfully!");

} else {

System.out.println("Patient not found!");

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int choice;

do {

System.out.println("\n----- Hospital Management System -----");

System.out.println("1. Add Patient");

System.out.println("2. Add Doctor");

System.out.println("3. Add Billing Details");

System.out.println("4. Show Available Rooms");

System.out.println("5. Show All Patients");

System.out.println("6. Show All Doctors");

System.out.println("7. Schedule Appointment");

System.out.println("8. Show All Appointments");

System.out.println("9. Discharge Patient");

System.out.println("10. Exit");

System.out.print("Enter your choice: ");

choice = sc.nextInt();

switch (choice) {

case 1: addPatient(); break;

case 2: addDoctor(); break;

case 3: addBilling(); break;

case 4: showAvailableRooms(); break;

case 5: showAllPatients(); break;

case 6: showAllDoctors(); break;

case 7: scheduleAppointment(); break;

case 8: showAppointments(); break;

case 9: dischargePatient(); break;

case 10: System.out.println("Exiting system..."); break;

default: System.out.println("Invalid choice! Please choose again.");

}

} while (choice != 10);

}

}