Small Clinic Management System-Documentation

Name: Khong Dinh Tu

ID: 24110145

---

1. Object-Oriented Analysis (OOA)

Following the 4-step OOA model, the system has the following objects:

1.1 Objects

- **Patient**
- **ChronicPatient** (derived from Patient)
- **Doctor**
- **Appointment**
- **Medicine**
- **Nurse**

1.2 Attributes for Each Object

- **Patient**: name, ID, age, gender, phoneNumber, medicalHistory
- **ChronicPatient**: conditionType, lastCheckupDate, medicationList
- **Doctor**: name, ID, specialty, yearsExperience, availability
- **Appointment**: date, time, reason, status, patientID, doctorID, location, duration
- **Medine**: name, dosage, frequency, startDate, endDate
- **Nurse**: name, ID, shift

1.3 Methods

- **Patient**: scheduleAppointment(), cancelAppointment(), addHistory(), displayInfo()
- **ChronicPatient**:Overrides scheduleAppointment(), adds addMedication(), listMedications(), enhanced displayInfo()
- **Doctor**: viewAppointment(), updateAppointment(), displayInfo()
- **Appointment**: updateStatus(), reschedule(), displayInfo()
- **Medicine**: displayInfo()
- **Nurse**: displayInfo()

1.4 Inheritance Relationships

- ChronicPatient inherits from Patient.
- Patient, Doctor, Appointment, Medicine, Nurse are independent

2. Class Design & Example Data

2.1 Class Design Details

- **Encapsulation**: Most data attributes are private or protected; accessed via methods.
- **Inheritance**: ChronicPatient extends Patient.
- **Polymorphism**: Virtual displayInfo() and scheduleAppointment().

2.2 Example Data

These example objects are created in main() before menu:

Doctors:
Dr. Smith (Cardiology, 15 yrs)
Dr. John (General Medicine, 8 yrs)

Nurses:
Nurse Eva (Morning shift), Nurse Muda (Night shift)

Patients:

Alice — regular patient, Age 30
Medical history: “Visited for flu (10/11/2024)”

Bob — chronic patient (Diabetes), Age 65
Medical history: “Diabetes checkup (10/06/2025)”; Medications: Metformin, Insulin

Appointments:

Alice scheduled 15/09/2025 at 10:00 for Regular Checkup

Bob scheduled 20/09/2025 at 09:00 for Diabetes Follow-up

These examples are printed under headings:

Example: Patient Info

Example: Chronic Patient Info

Example: Doctor Info

Example: Nurse Info

Example: Appointment List

This helps the user see sample output immediately.

3. Code Walkthrough – Key parts

scheduleAppointment() in **Patient**

```
virtual void scheduleAppointment(vector<Appointment>& appointments,string date, string time, string reason,
int doctorID, string l = "Clinic Room 1",int d = 30) {
        appointments.push_back(Appointment(date, time, reason, id, doctorID, l, d));
        cout << "Patient " << name << " scheduled an appointment on "<< date << " at " << time << " for "
 << reason << endl;
    }
```

Override in **ChronicPatient**

```
void scheduleAppointment(vector<Appointment>& appointments,string date, string
 time, string reason,int doctorID, string location = "Clinic Room 1",int
duration = 30) override {
        appointments.push_back(Appointment(date, time, reason + " (Chronic)"
, id, doctorID, location, duration));
        cout << "Chronic Patient " << name << " scheduled an appointment on "
<< date << " at " << time << endl;
    }
```

displayInfo() in Both Classes

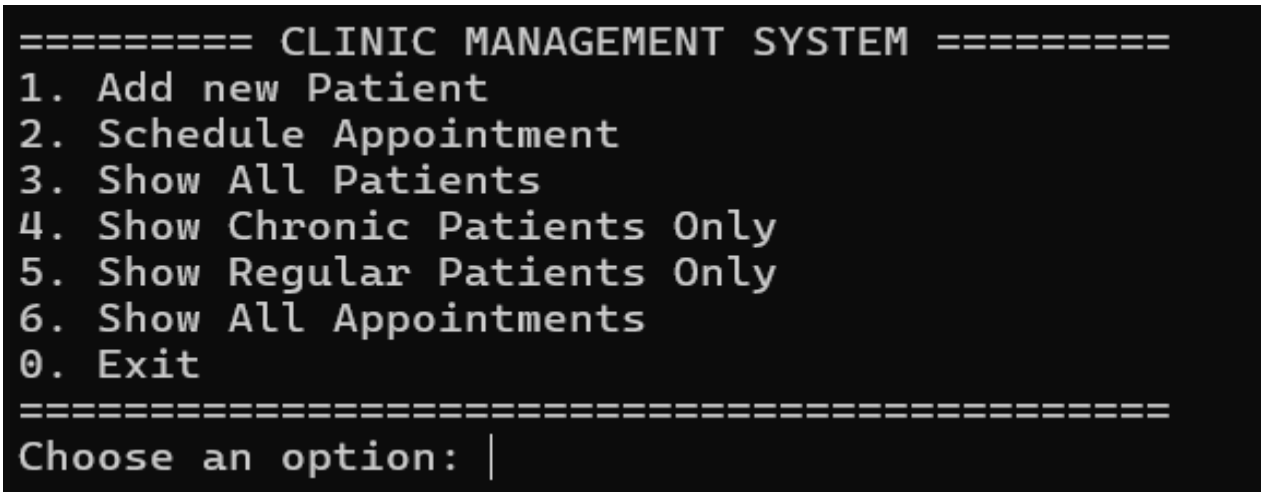
- For **Patient**: shows name, ID, age, gender, phone number, medical history.
- For **ChronicPatien**t: in addition conditionType, lastCheckupDate, displays medications via listMedications().

4. Sample Output

When running with the example data , console output includes:

```
=== The schedule has been arranged ===
Patient Alice scheduled an appointment on 15/09/2025 at 10:00 for Regular Checkup
Chronic Patient Bob scheduled an appointment on 20/09/2025 at 09:00

=== Patient Info ===
[ID: 101] Alice, Age: 30 (Adult), Gender: Female, Phone: 0123-456-789
Medical History: Visited for flu (10/11/2024);
--------------------------------

=== Chronic Patient Info ===
[ID: 102] Bob, Age: 65 (Senior), Gender: Male, Phone: 0987-654-321, Condition: Diabetes, Last Checkup: 15/06/2023
Medical History: Diabetes checkup (10/06/2025);
Medications for Bob:
Metformin | Dosage: 500mg | Frequency: 2 times/day | Start: 01/09/2025 | End: 01/12/2025
Insulin | Dosage: 10 units | Frequency: Once/day | Start: 01/09/2025 | End: 01/12/2025
--------------------------------

=== Doctor Info ===
[ID: 201] Dr. Smith, Specialty: Cardiology, Experience: 15 years, Availability: Mon-Fri 9:00-17:00
[ID: 202] Dr. John, Specialty: General Medicine, Experience: 8 years, Availability: Mon-Sat 8:00-14:00

=== Nurse Info ===
[ID: 36] Eva, Shift: Morning
[ID: 72] Muda, Shift: Night

=== Appointment List ===
Appointment: 15/09/2025 10:00 | Reason: Regular Checkup | Status: Scheduled | PatientID: 101 | DoctorID: 202 | Location:
 Clinic Room 2 | Duration: 30 minutes
Appointment: 20/09/2025 09:00 | Reason: Diabetes Follow-up (Chronic) | Status: Scheduled | PatientID: 102 | DoctorID: 20
1 | Location: Clinic Room 1 | Duration: 45 minutes
```

After the example, the system enters a loop presenting a menu:



This allows flexible operations after seeing sample data.

5. Use of LLM (ChatGPT)

I used ChatGPT for:

- Brainstorming additional features (Medicine class, Nurse class, interactive menu).
- Debugging printing output (how to separate regular vs chronic patient info).
- Clarifying C++ concepts like dynamic_cast and virtual methods.

Example Prompt:

"Suggest a method for a Patient class in a small clinic management system."

Response:

ChatGPT suggested methods like cancelAppointment(), displayInfo(),

inputMedicalHistory(),

All code was understood and personally written.