CLINIC MANAGEMENT SYSTEM - DESIGN & IMPLEMENTATION REPORT

1. Collection Choices and Justifications

A. Patient Management: HashMap<Integer, Patient>
- **Usage**: Used in ClinicManagementSystem to store all patients.
- **Justification**: The most frequent operation is looking up a patient by their ID (e.g., during appointment scheduling). HashMap makes it easy and faster data retrieval.

B. Doctor Management: ArrayList<Doctor>
- **Usage**: Used in ClinicManagementSystem to store the list of doctors.
- **Justification**: The number of doctors is expected to be relatively small compared to patients.

C. Appointment Management (Whole System): TreeMap<String, ArrayList<Appointment>>
- **Usage**: Stores appointments based by date (String "YYYY-MM-DD").
-**Justification**: Reporting requirements often involve generating daily schedules. TreeMap automatically keeps the keys sorted naturally allowing generating reports in chronological order without sorting.

D. Appointment Management (Specific Patient): TreeMap<String, Appointment>
- **Usage**: Inside the Patient class to store their personal schedule.
- **Justification**: Similar to the whole system record, it is beneficial to keep a patient's history sorted. The key is a composite string of "Date Time" to ensure uniqueness and chronological sorting.

E. Medical History: ArrayList<MedicalRecord>
- **Usage**: Inside the Patient class.
- **Justification**: Medical history is a sequential record of events. An ArrayList preserves the insertion order (chronological) and allows for dynamic growth as new records are added.

2. OOP design choices

A. Abstract 'Person' Class
- **Decision**: Created an abstract base class 'Person' containing common attributes (name, email, phone).
- **Benefit**: Reduces code duplication between Doctor and Patient classes and establishes a clear hierarchy

B. Interfaces
- **Decision**: Implemented 'FileOperations' and 'ReportGenerator' as separate interfaces.
- **Benefit**: Decouples the definition of behavior from the implementation.
  - **FileOperations** ensures any class implementing it can handle file saving and loading.
  - **ReportGenerator** ensures standard reporting capabilities.

This design allows the ClinicManagementSystem to implement multiple specific behaviors while keeping the contracts clear.

C. Composition
- **Decision**: The 'ClinicManagementSystem' is composed of collections of Patients and Doctors, rather than inheriting from them.
- **Benefit**: Properly models the relationships, ensuring better encapsulation and maintainability.

---

1 RULE: Open Main.java, run it, then follow the system requiremetes