Exercise 02.00.1

Cross-Cutting Concerns in practice

Scenario Overview

You are designing a Healthcare Appointment System with:

Backend Modules

1. **User Management** (Handles login, roles, and authentication)
2. **Appointment Scheduling** (Manages patient bookings and availability)
3. **Billing & Payments** (Processes transactions securely)
4. **Notification Service** (Sends appointment reminders via email/SMS)

Frontend Modules

1. **Doctor Dashboard** (Web app for doctors to manage appointments)
2. **Patient Portal** (Web and mobile access for patients)
3. **Admin Panel** (Used by IT/security teams for auditing)

Constraints

You must design a solution while adhering to specific constraints:

1. Logging & Observability

- All modules **must log important events** (logins, appointment changes, payments).
- Logs **must be centralized** for querying and analysis.
- Logs **cannot store personally identifiable information (PII)**.
- **Billing-related logs must be encrypted** before storage.
- Some modules require **real-time logging**, while others allow **batch processing** for performance optimization.

2. Error Handling & Recovery

- **All errors must be logged**, but sensitive information **cannot be exposed** in error messages.
- **Critical errors (e.g., payment failures) must trigger alerts** and be **visible to admins**.
- The system **must automatically retry failed operations where applicable**, but **should not cause cascading failures**.
- The frontend **must provide meaningful feedback to users** without exposing system internals.

3. Caching Strategy

- Some modules require caching, while others **cannot** use caching due to compliance:
    - **Appointment Scheduling & Doctor Dashboard should use caching** for fast lookups.
    - **Billing & Payments must not use caching** due to financial compliance.
    - **User sessions** cannot be cached server-side; only frontend session storage is allowed.
- The system must **invalidate stale cache entries automatically**.

4. Sending Communications to Users (Email/SMS Notifications)

- **Appointment confirmations, cancellations, and reminders** must be sent via **SMS & Email**.
- **Emails must be templated and localized** for different regions.
- The notification system **must not block other operations** (e.g., sending an email cannot slow down booking an appointment).
- **Retries should be handled efficiently** to avoid spam but ensure delivery.

Your Task

1. **Design an approach** for handling **logging, error handling, caching, and notifications** while respecting constraints.
2. **Provide a simple architecture diagram** showing how these constraints are handled.
3. **List the pros & cons of your approach** — why did you choose this over others?

Deliverables

- **Diagrams / UML / Flowcharts**
- **Pros & Cons of the Approach**