Modular Monolith Course Assessment

Description

You are tasked with creating a backend system for a doctor appointment booking application. The system will be designed for a specific single doctor and should handle the logic behind managing and booking appointments. The project focuses on implementing the necessary APIs and functionality to meet the business requirements.

Business Requirements:

Your application should adhere to the following business requirements:

1. **Doctor Availability:**
    a. As a doctor, I want to be able to list my slots
    b. As a doctor, I want to be able to add new slots where a single time slot should have the following:
        i. Id: Guid
        ii. Time: Date → 22/02/2023 04:30 pm
        iii. DoctorId: Guid
        iv. DoctorName: string
        v. IsReserved: bool
        vi. Cost: Decimal
2. **Appointment Booking:**
    a. As a **Patient,** I want to be able to **view** all doctors' available (only) **slots**
    b. As a **Patient,** I want to be able to **book** an appointment on a free **slot where.** An Appointment should have the following:
        i. Id: Guid
        ii. SlotId: Guid
        iii. PatientId: Guid
        iv. PatientName: string
        v. ReservedAt: Date

3. **Appointment Confirmation:**
    a. Once a patient schedules an appointment, the system should send a confirmation notification to the **patient** and the **doctor**
    b. The confirmation notification should include the appointment details, such as the patient's name, appointment time, and Doctor's name.
    c. For the sake of this assessment, the notification could be just a **Log message**
4. **Doctor Appointment Management:**
    a. As a Doctor, I want to be able to view my upcoming **appointments**.
    b. As a Doctor, I want to be able to mark appointments as **completed** or **cancel** them if necessary.
5. **Data Persistence:**
    a. Use any db engine or even **in-memory list** with no db at all

Specifications:

1. You don't need to care about authentication or authorization, make it public APIs
2. Assume the system is serving a single Doctor only
3. Apply **modular monolith architecture**
4. The system should consist of **four** modules each with a different architecture as follows:
    a. **Doctor Availability Module**: Traditional Layered Architecture
    b. **Appointment Booking Module**: Clean architecture
    c. **Appointment Confirmation Module**: Simplest architecture possible
    d. Doctor Appointment Management: Hexagonal Architecture
5. **(A plus point)**Write unit and integration testing

Deliverables

Push the complete source code of your application to a source code repository of your choice (e.g., GitHub, GitLab, Bitbucket).

Ensure that commits are done incrementally and reflect your progress throughout development.

Evaluation Criteria

Your project will be evaluated based on the following criteria:

1. Correct implementation of all the required business requirements.
2. Compliance with each specified architecture rule
3. Proper Modularity and integration between them
4. Code quality, including readability, maintainability, adherence to best practices, and separation of concerns

Good luck!