System Design Assignment

Please design and implement a system consisting of **three independent modules**: **User Module**, **Booking Module**, and **Finance Module**. Each module must be treated as a **separate Bounded Context**.

User Module

The User Module is responsible for user registration and user lifecycle management.

User information includes:

- User ID
- First name
- Last name
- Email
- Mobile number
- Creation date

After a successful registration, a **UserCreated** domain event must be published.

Booking Module

The Booking Module is responsible for creating and managing bookings. When a simple booking is created, user information must be retrieved **synchronously** from the User Module.

Booking information includes:

- Booking ID
- User ID
- Booking date
- Amount
- Status

Finance Module

The Finance Module is responsible for financial processing. After a booking is successfully created, booking data must be received **asynchronously** and used to generate an invoice.

Invoice information includes:

- Invoice ID
- Booking ID
- User ID
- Total amount
- Creation date
- Payment status

Communication & Architecture Constraints

- Synchronous communication between modules must be implemented using **gRPC**.
- Asynchronous communication must be implemented using **RabbitMQ**.
- Clean Architecture principles must be followed.
- Domain-Driven Design (DDD) must be applied with clear Bounded Contexts.
- CQRS must be used where applicable.
- Each module must use **SQL Server** as its persistence database, with a dedicated database per module. Direct access to another module's database is not allowed.

The primary goal of this assignment is to evaluate design quality, proper domain boundaries, and architectural decision-making rather than implementation completeness.

If you have any questions or require clarification, feel free to ask.