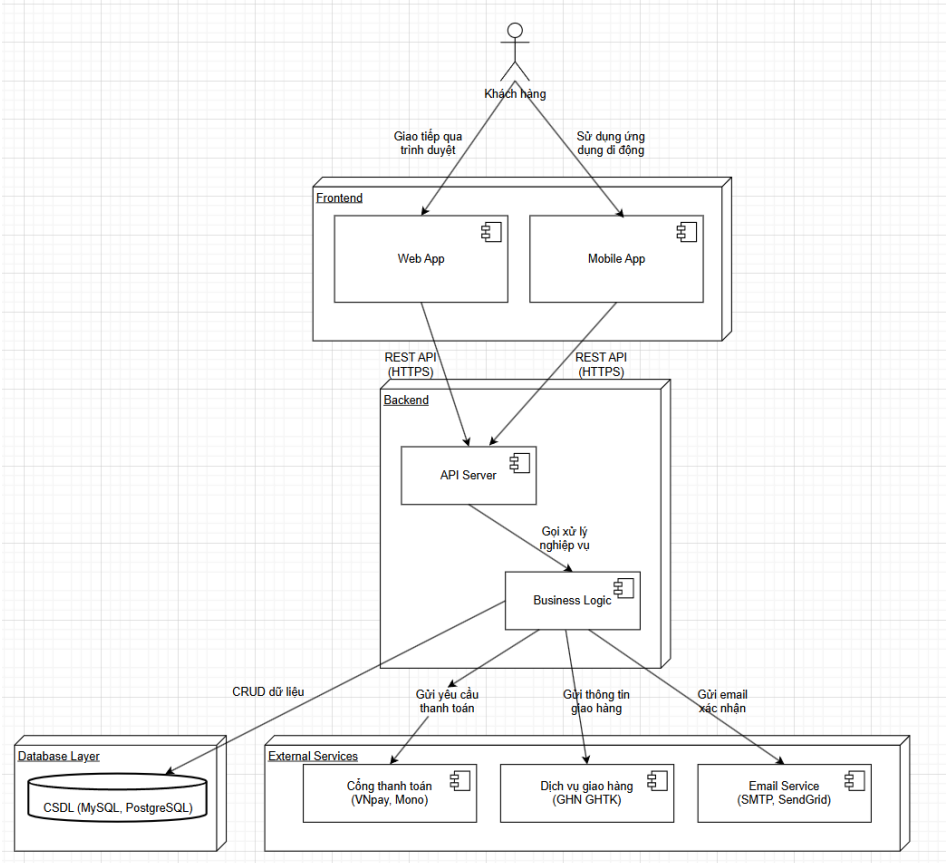


Bài 1:

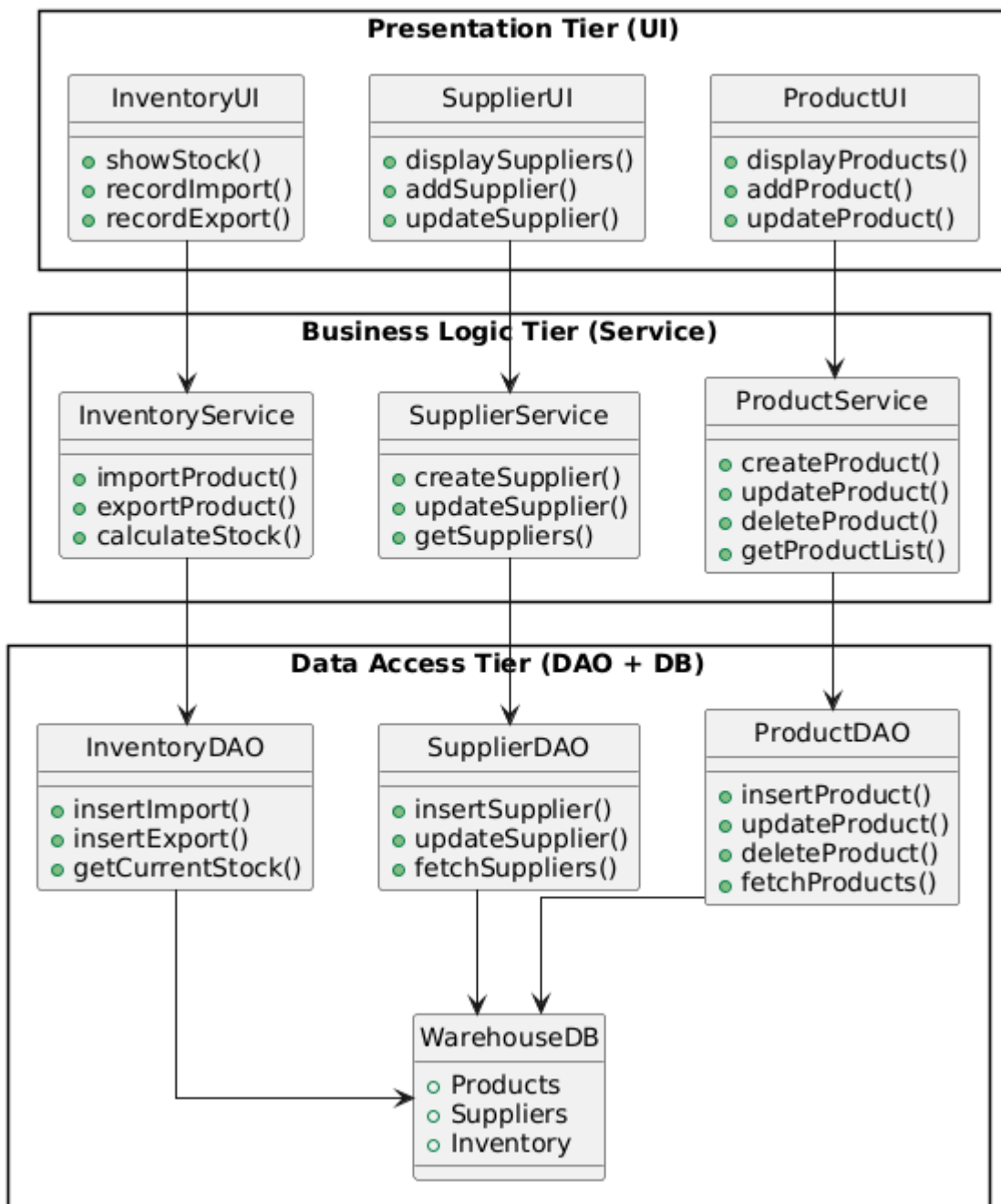
Các thành phần chính trong kiến trúc

Thành phần	Mô tả ngắn gọn
Frontend (Web & Mobile App)	Giao diện người dùng – nơi khách hàng thực hiện thao tác mua sắm, đăng nhập, thanh toán. Web chạy trên trình duyệt, Mobile là ứng dụng native hoặc hybrid.
Backend (API Server)	Xử lý nghiệp vụ như giỏ hàng, thanh toán, xác thực, quản lý đơn hàng. Cung cấp API RESTful cho cả Web và Mobile.
Database (CSDL)	Lưu trữ dữ liệu về người dùng, sản phẩm, đơn hàng, giao dịch, v.v. (VD: MySQL, PostgreSQL).
External Services	Các dịch vụ bên ngoài như cổng thanh toán (VNPay, Momo), vận chuyển (GHTK, GHN), gửi email xác nhận, v.v.

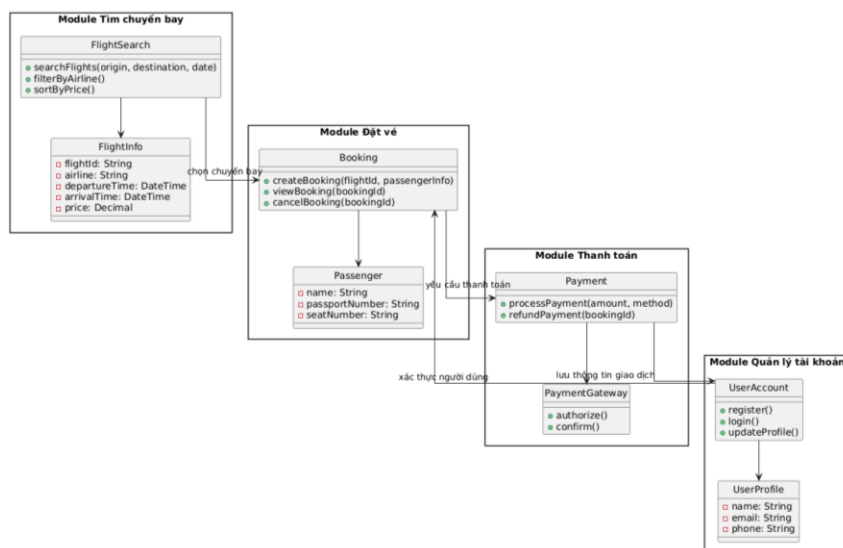


Bài 2:

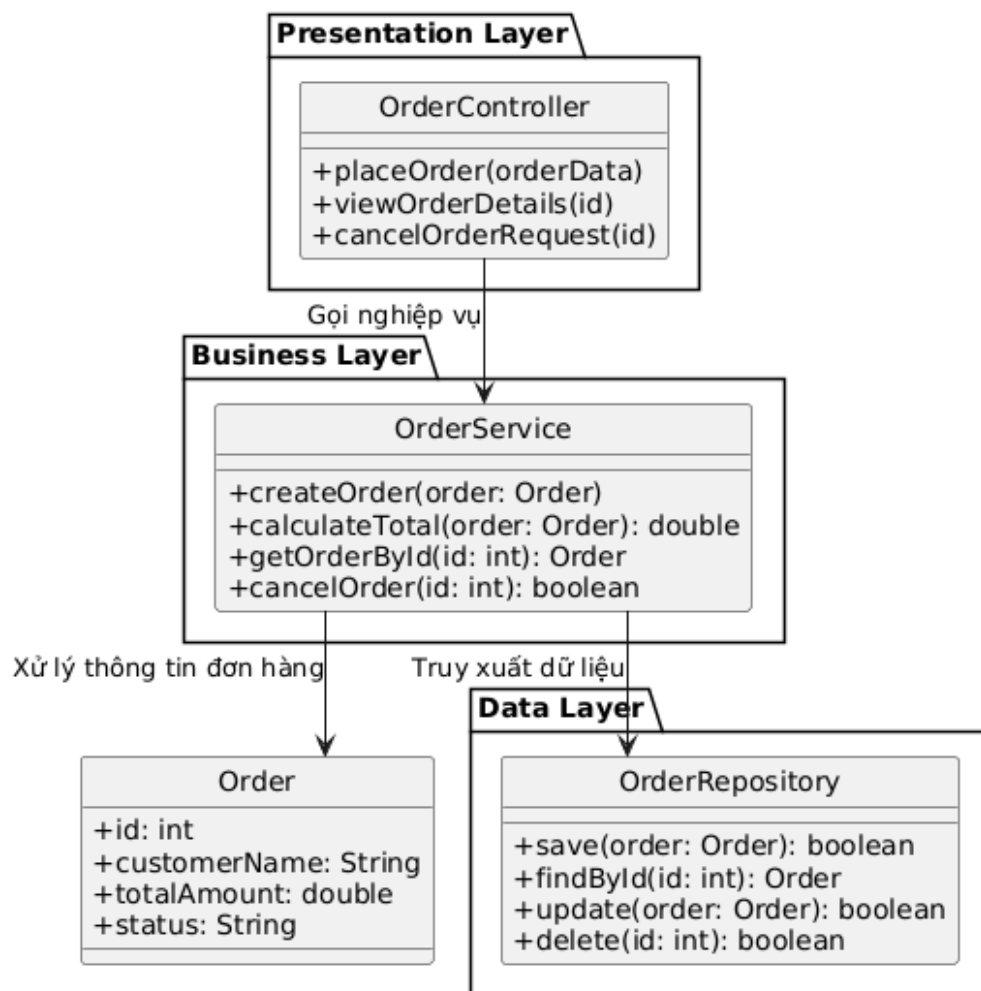
Kiến trúc 3 tầng - Hệ thống quản lý kho



Bài 3:



Bài 4:



Business Layer: OrderService

Chức năng: Xử lý nghiệp vụ đặt hàng, tính tổng tiền, xác thực dữ liệu trước khi lưu.

Thuộc tính / Phương thức	Mô tả
+ createOrder(order: Order)	Kiểm tra tính hợp lệ và gửi yêu cầu lưu đơn hàng.
+ calculateTotal(order: Order): double	Tính tổng tiền dựa trên sản phẩm và số lượng.
+ getOrderById(id: int): Order	Lấy thông tin đơn hàng cụ thể từ Repository.
+ cancelOrder(id: int): boolean	Hủy đơn hàng (kiểm tra trạng thái trước khi hủy).

Data Layer: OrderRepository

Chức năng: Truy xuất và thao tác dữ liệu với cơ sở dữ liệu (CSDL).

Thuộc tính / Phương thức	Mô tả
+ save(order: Order): boolean	Lưu đơn hàng mới vào CSDL.
+ findById(id: int): Order	Truy vấn đơn hàng theo ID.
+ update(order: Order): boolean	Cập nhật thông tin đơn hàng.
+ delete(id: int): boolean	Xóa đơn hàng khỏi CSDL.

Presentation Layer: OrderController

Chức năng: Giao tiếp với người dùng hoặc giao diện web, nhận yêu cầu và gọi Business Layer để xử lý.

Phương thức	Mô tả
+ placeOrder(orderData)	Nhận thông tin đặt hàng từ giao diện và gửi đến OrderService.
+ viewOrderDetails(id)	Hiển thị thông tin đơn hàng.
+ cancelOrderRequest(id)	Gửi yêu cầu hủy đơn hàng.

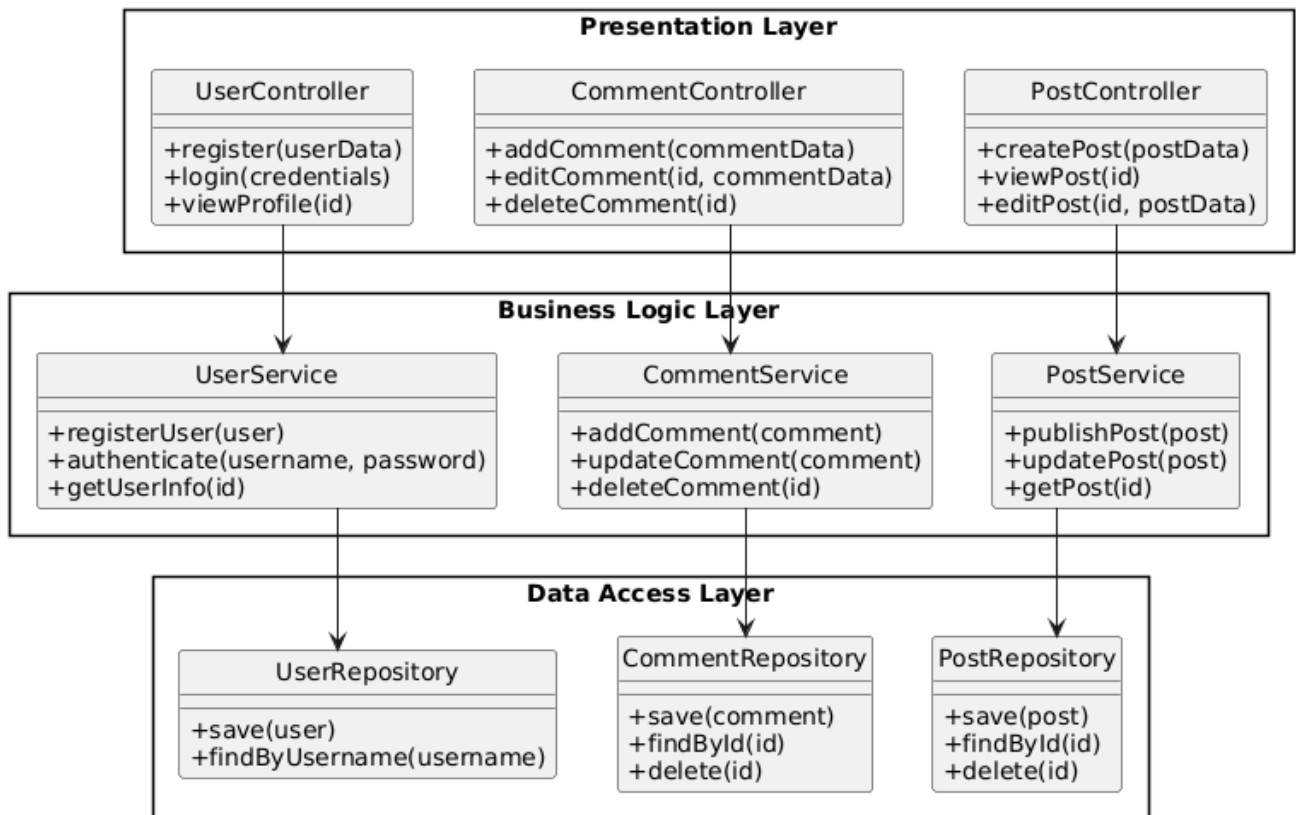
Bài 5:

Mô tả các module và vai trò

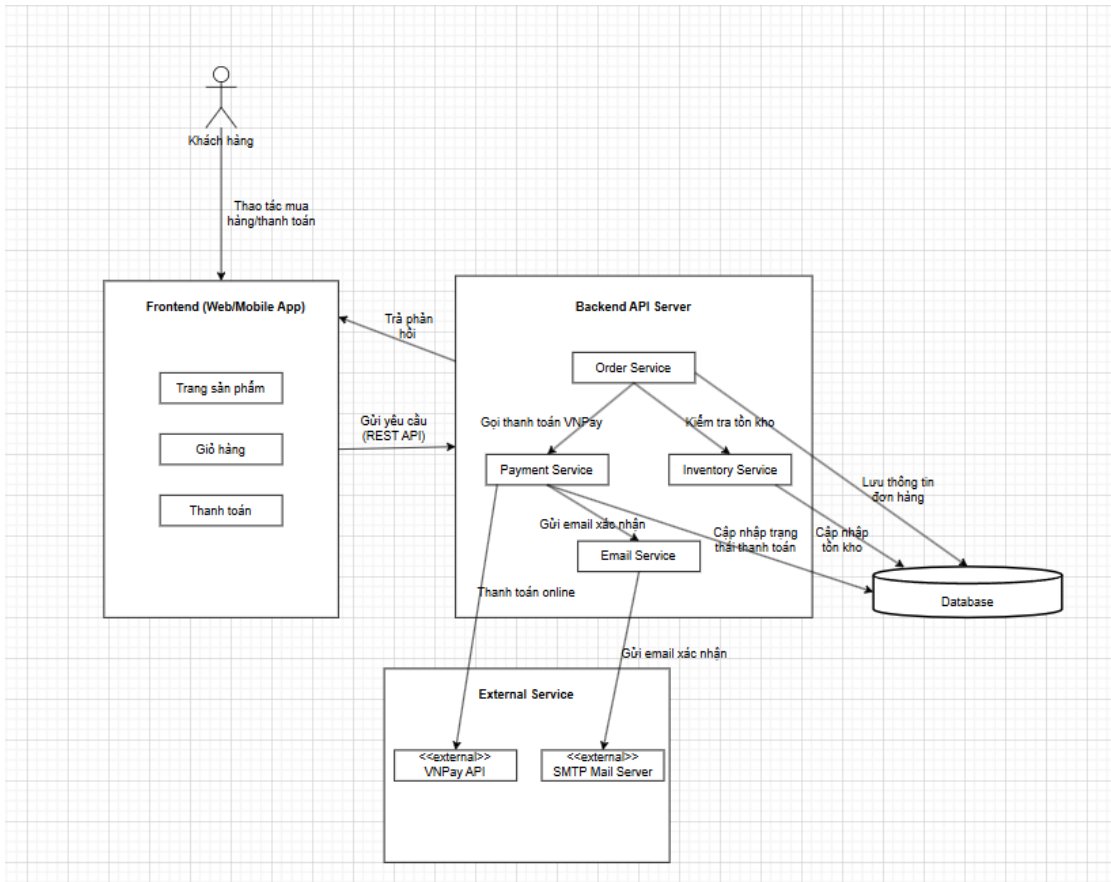
Tầng	Module	Chức năng chính
Presentation	PostController	Hiển thị danh sách bài viết, trang chi tiết bài viết, nhận yêu cầu đăng bài.
	CommentController	Gửi, chỉnh sửa, xóa bình luận từ giao diện.
	UserController	Đăng ký, đăng nhập, quản lý hồ sơ người dùng.

Business Logic	PostService	Kiểm tra nội dung bài viết, xử lý đăng và cập nhật bài viết.
	CommentService	Kiểm tra quyền bình luận, xử lý logic thêm/xóa/sửa bình luận.
	UserService	Xác thực người dùng, phân quyền, mã hóa mật khẩu.
Data Access	PostRepository	Lưu, sửa, xóa, truy xuất bài viết từ CSDL.
	CommentRepository	Quản lý dữ liệu bình luận trong CSDL.
	UserRepository	Lưu trữ và truy vấn thông tin người dùng.

Sơ đồ Class - Hệ thống Blog cá nhân theo kiến trúc 3 tầng



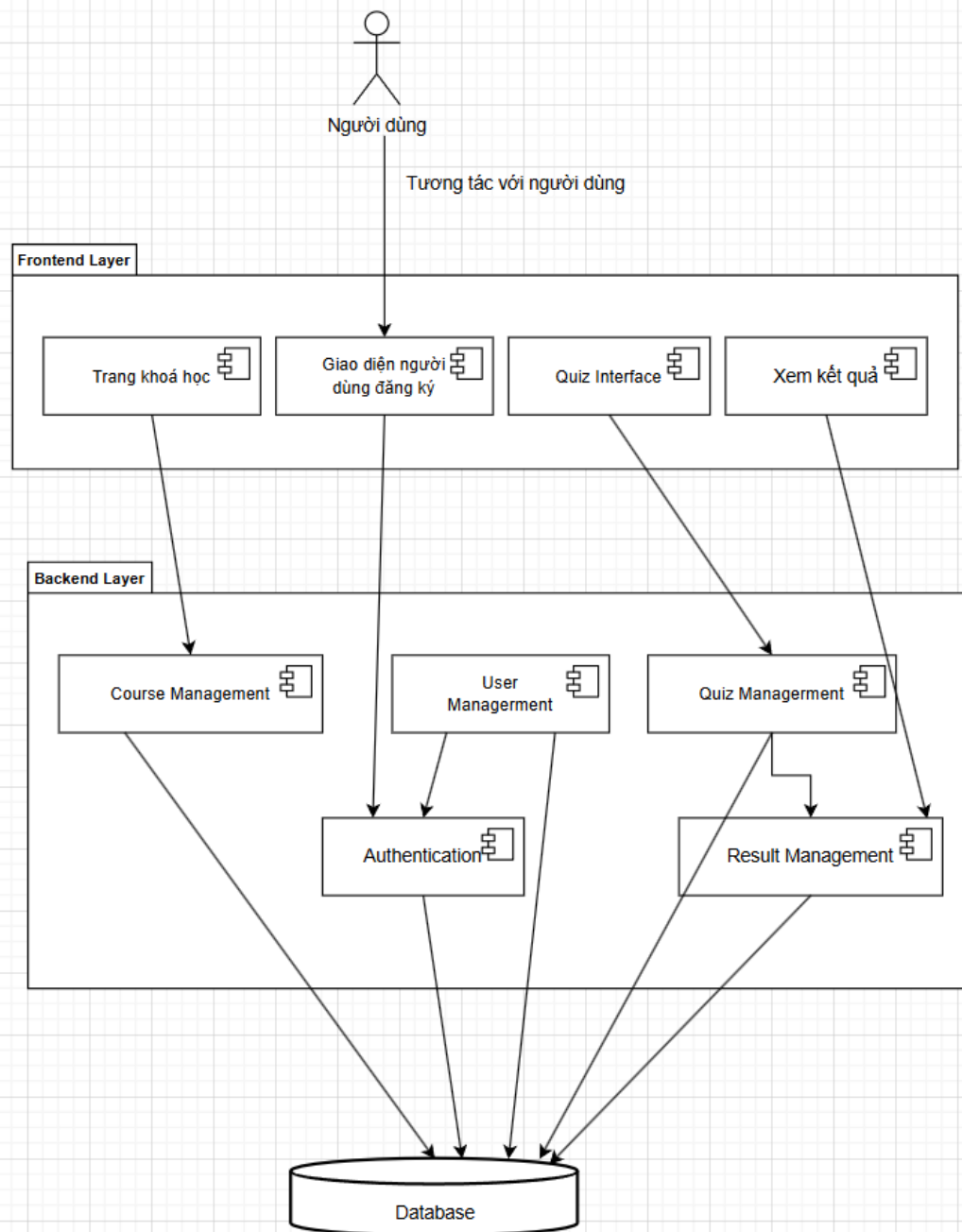
Bài 6:



Bài 7:

Bảng Mapping Use Case → Module

Use Case	Module tương ứng	Tầng kiến trúc	Mô tả ngắn
Đăng ký tài khoản	User Management	Backend + Database	Quản lý thông tin người dùng, lưu dữ liệu tài khoản mới.
Đăng nhập	Authentication	Backend + Database	Xác thực người dùng, kiểm tra thông tin đăng nhập.
Xem khóa học	Course Management	Frontend + Backend + Database	Hiển thị danh sách khóa học, lấy dữ liệu từ backend.
Làm bài quiz	Quiz Management	Frontend + Backend + Database	Cho phép người học làm bài quiz, chấm điểm tự động.
Xem kết quả	Result Management	Frontend + Backend + Database	Hiển thị kết quả học tập, điểm quiz, thống kê cá nhân.



Bài 8:

Kiến trúc 3 tầng (3-Tier Architecture)

1. Presentation Layer (Tầng trình diễn)

- Là **giao diện người dùng** – nơi khách hàng, quản lý, nhân viên tương tác.

- Chịu trách nhiệm **hiển thị thông tin** và **gửi yêu cầu** tới tầng nghiệp vụ.

Ví dụ lớp:

- ProductUI
- OrderUI
- LoginUI
- ReportUI

2. Business Logic Layer (Tầng nghiệp vụ)

- Chứa toàn bộ **logic xử lý, kiểm tra điều kiện, và tính toán**.
- Không trực tiếp truy cập cơ sở dữ liệu, mà **gọi đến tầng Data Access**.

Ví dụ lớp:

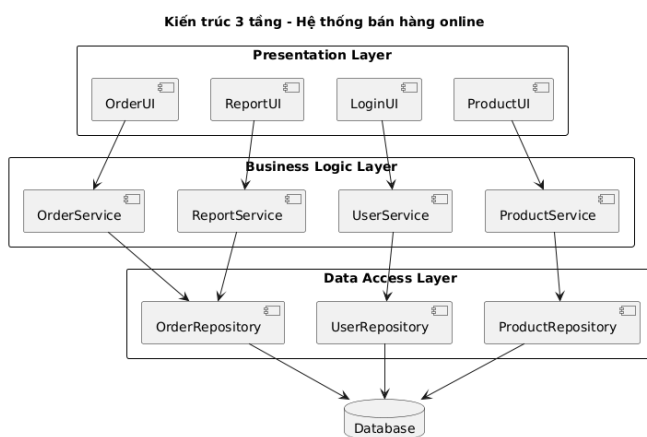
- ProductService
- OrderService
- UserService
- ReportService

3. Data Access Layer (Tầng dữ liệu)

- Phụ trách **lưu trữ và truy xuất dữ liệu** (Database, file, API...)
- Các lớp Repository hay DAO (Data Access Object) nằm tại đây.

Ví dụ lớp:

- ProductRepository
- OrderRepository
- UserRepository



Giải thích mối quan hệ

Tầng	Vai trò chính	Ví dụ
Presentation	Giao diện, thu thập đầu vào từ người dùng	ProductUI, LoginUI
Business Logic	Xử lý nghiệp vụ, kiểm tra điều kiện, gọi repository	ProductService
Data Access	Lưu và truy vấn dữ liệu	ProductRepository

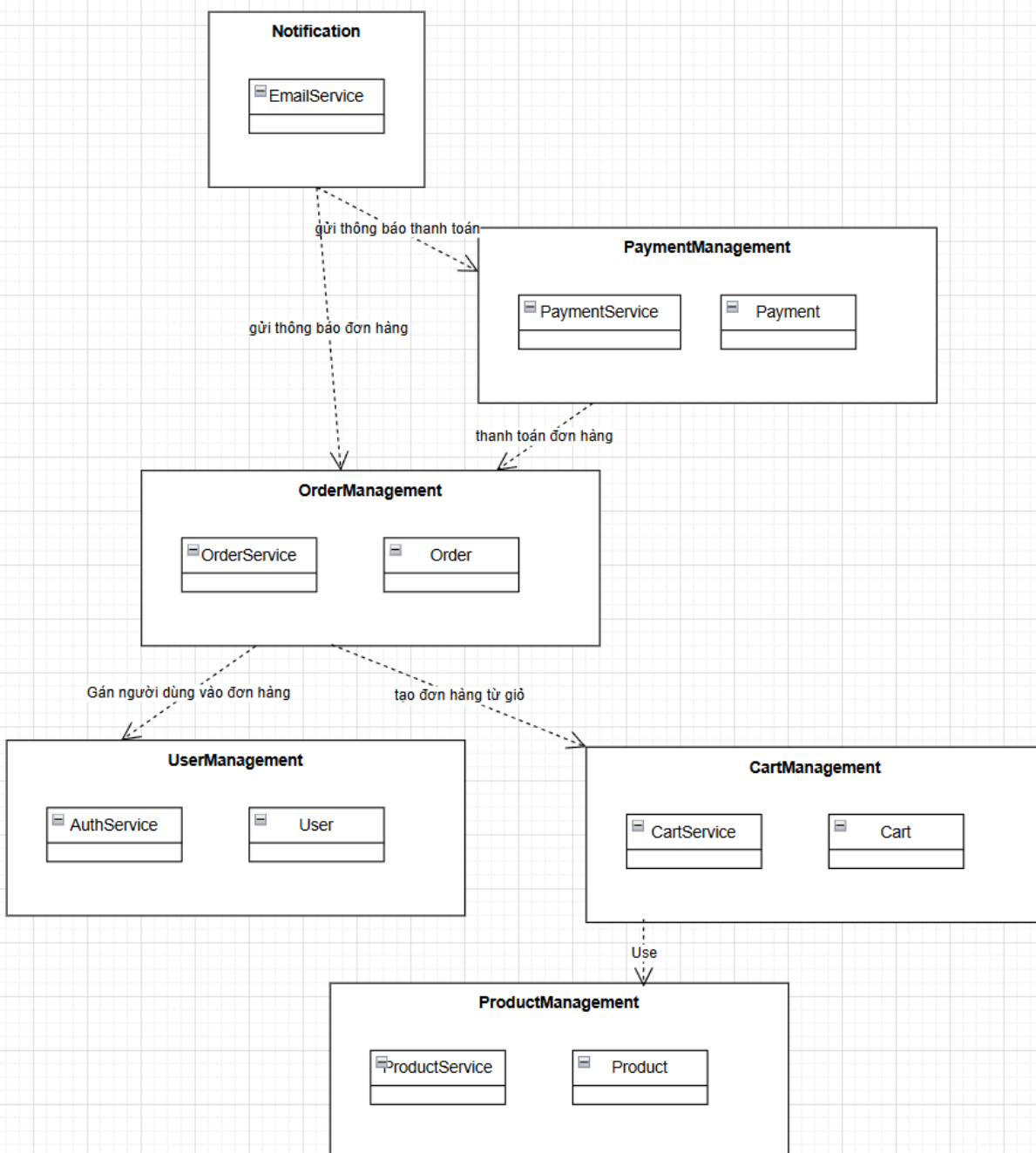
Bài 9:

Các package chính trong hệ thống

Package	Vai trò	Mô tả chức năng
ProductManagement	Quản lý sản phẩm	Thêm, sửa, xóa, xem chi tiết sản phẩm
OrderManagement	Quản lý đơn hàng	Tạo đơn hàng, cập nhật trạng thái, theo dõi đơn
CartManagement	Quản lý giỏ hàng	Thêm sản phẩm vào giỏ, chỉnh sửa hoặc xóa
PaymentManagement	Quản lý thanh toán	Xử lý thanh toán trực tuyến qua các cổng như VNPay, Momo
UserManagement	Quản lý người dùng	Đăng ký, đăng nhập, cập nhật hồ sơ người dùng
Notification (<i>tùy chọn mở rộng</i>)	Gửi thông báo	Gửi email xác nhận đơn hàng, thanh toán

Dependency logic:

- CartManagement **phụ thuộc** vào ProductManagement (cần biết thông tin sản phẩm để thêm vào giỏ).
- OrderManagement **phụ thuộc** vào CartManagement và UserManagement (để tạo đơn hàng cho người dùng).
- PaymentManagement **phụ thuộc** vào OrderManagement (để xử lý thanh toán đơn hàng).
- Notification **phụ thuộc** vào OrderManagement và PaymentManagement (gửi thông báo sau thanh toán thành công).



Bài 10:

Kiến trúc tổng thể 3 tầng

1. Presentation Layer

- **Chức năng:** Giao diện người dùng (Web hoặc Mobile)
- **Module:**

- StudentPortal: Đăng nhập, xem điểm, đăng ký môn học
- TeacherPortal: Quản lý lớp, nhập điểm, tạo lịch học
- AdminDashboard: Quản trị hệ thống, thêm/sửa môn học

2. Business Logic Layer

- **Chức năng:** Xử lý nghiệp vụ đào tạo
- **Module:**
 - StudentService
 - TeacherService
 - CourseService
 - ScheduleService
 - GradeService

3. Data Access Layer

- **Chức năng:** Giao tiếp với cơ sở dữ liệu
- **Module:**
 - StudentRepository
 - TeacherRepository
 - CourseRepository
 - ScheduleRepository
 - GradeRepository

