

Session 07 – Phân tích và thiết kế kiến trúc hệ thống

Bài 1:

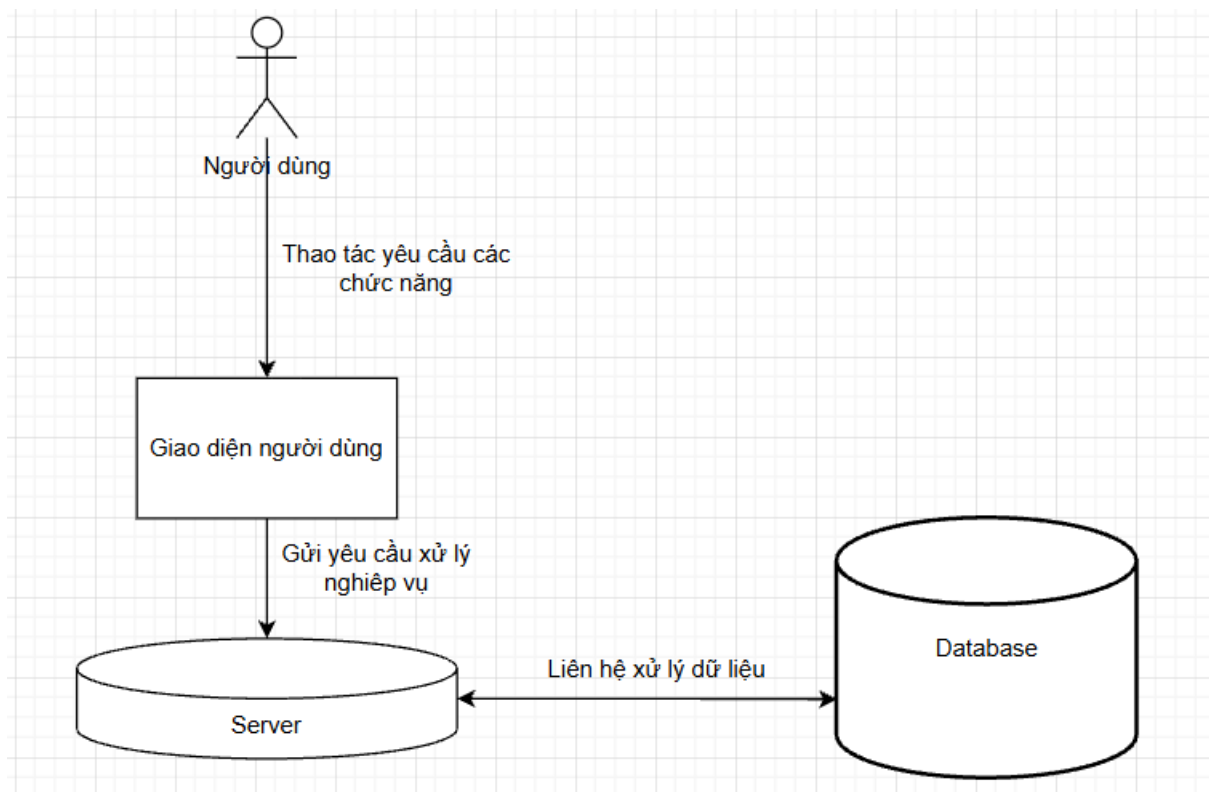
1. Kiến trúc tổng thể - Hệ thống quản lý bán hàng

- Tầng giao diện → Giao diện, đại diện hệ thống tương tác với người dùng và hiển thị các trạng thái và thông tin được trả về sau khi xử lý các nghiệp vụ quản lý
- Tầng nghiệp vụ → Đảm nhận xử lý logic các chức năng nghiệp vụ
- Tầng dữ liệu → Truy xuất và lưu trữ dữ liệu xử lý CRUD thông tin đơn hàng
- Tầng tích hợp → Bao gồm các hệ thống hỗ trợ cho việc quản lý

2. Các thành phần chính

- Frontend – bao gồm: **giao diện (Web và Mobile app)** giúp người dùng thao tác quản lý với hệ thống
- Backend – bao gồm: Nơi xử lý logic chính, tính toán, xác thực dữ liệu và gọi đến tầng dữ liệu khi cần – Đảm bảo quy tắc nghiệp vụ
- Database – cơ sở dữ liệu – Nơi chịu trách nhiệm tương tác với nguồn dữ liệu, thực hiện các thao tác CRUD
- Các dịch vụ tích hợp: Hệ thống thanh toán tích hợp, xử lý xác thực bằng email,...

3. Sơ đồ kiến trúc tổng thể:



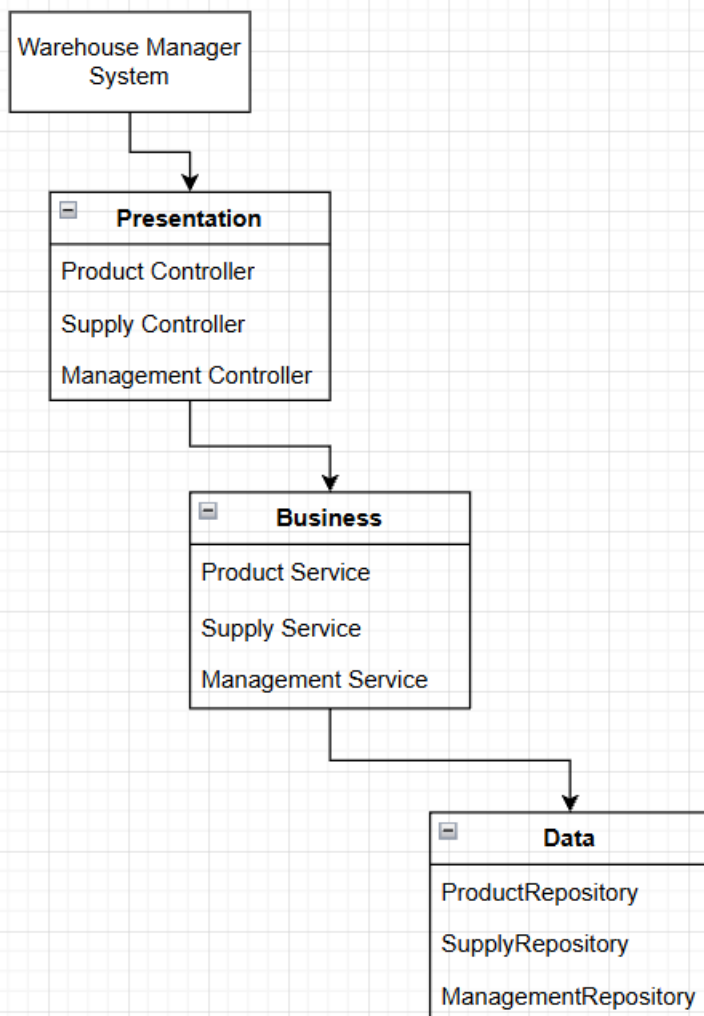
Bài 2:

❖ Thiết kế kiến trúc 3 tầng cho hệ thống:

- Presentation Tier – tầng giao diện

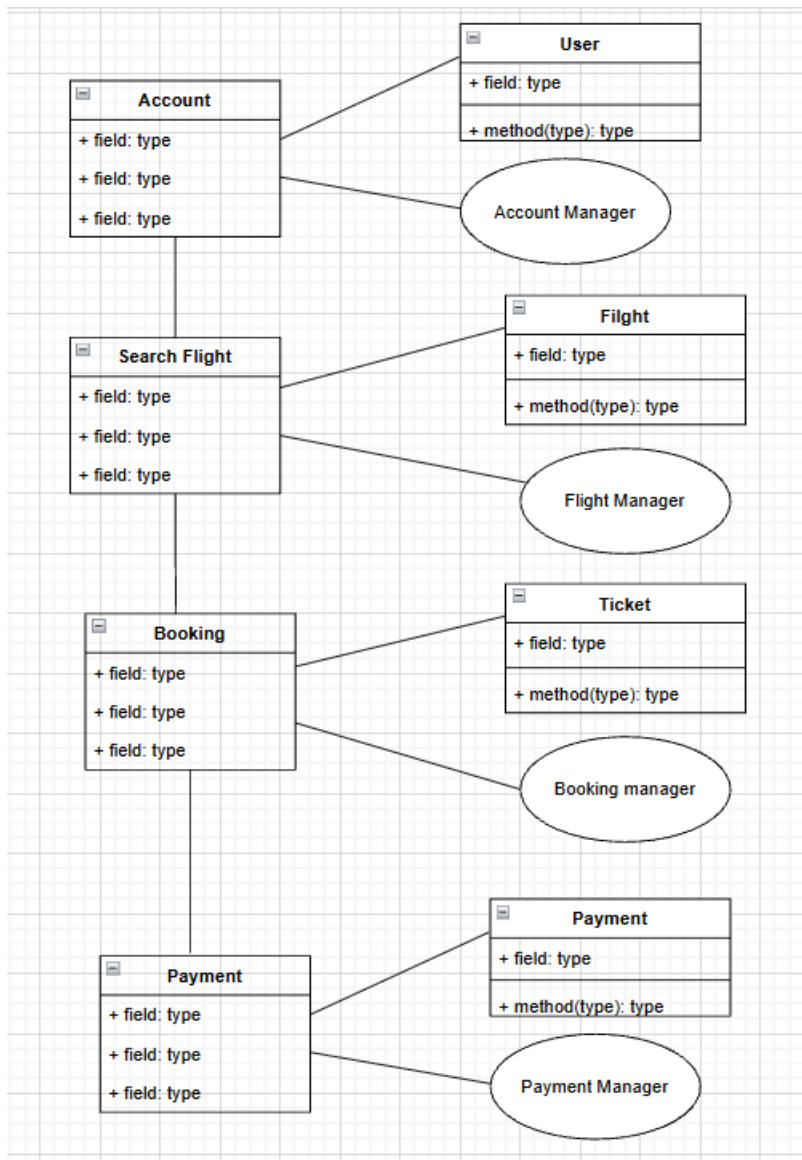
- Nơi người dùng (nhân viên kho/quản lý): tương tác qua giao diện web hoặc mobile app
- Có vai trò gửi yêu cầu từ người dùng → Tầng nghiệp vụ và hiển thị kết quả phản hồi cho người dùng
- Business Tier – Tầng nghiệp vụ
 - Xử lý toàn bộ logic của hệ thống, kiểm tra hợp lệ dữ liệu, tính toán, và điều phối giữa các tầng.
 - Có vai trò
 - Kiểm tra tính hợp lệ của yêu cầu từ Presentation Tier.
 - Gọi đến tầng Data Access để truy vấn/ghi dữ liệu.
 - Xử lý logic: ví dụ tính “Số lượng tồn kho = Tổng nhập – Tổng xuất.
- Data Access Tier – Tầng dữ liệu:
 - Quản lý việc lưu trữ, chỉnh sửa và truy xuất dữ liệu
 - Có vai trò:
 - Đảm bảo dữ liệu được lưu trữ an toàn, toàn vẹn.
 - Cấp API để tầng nghiệp vụ gọi đến.

❖ Trình bày:

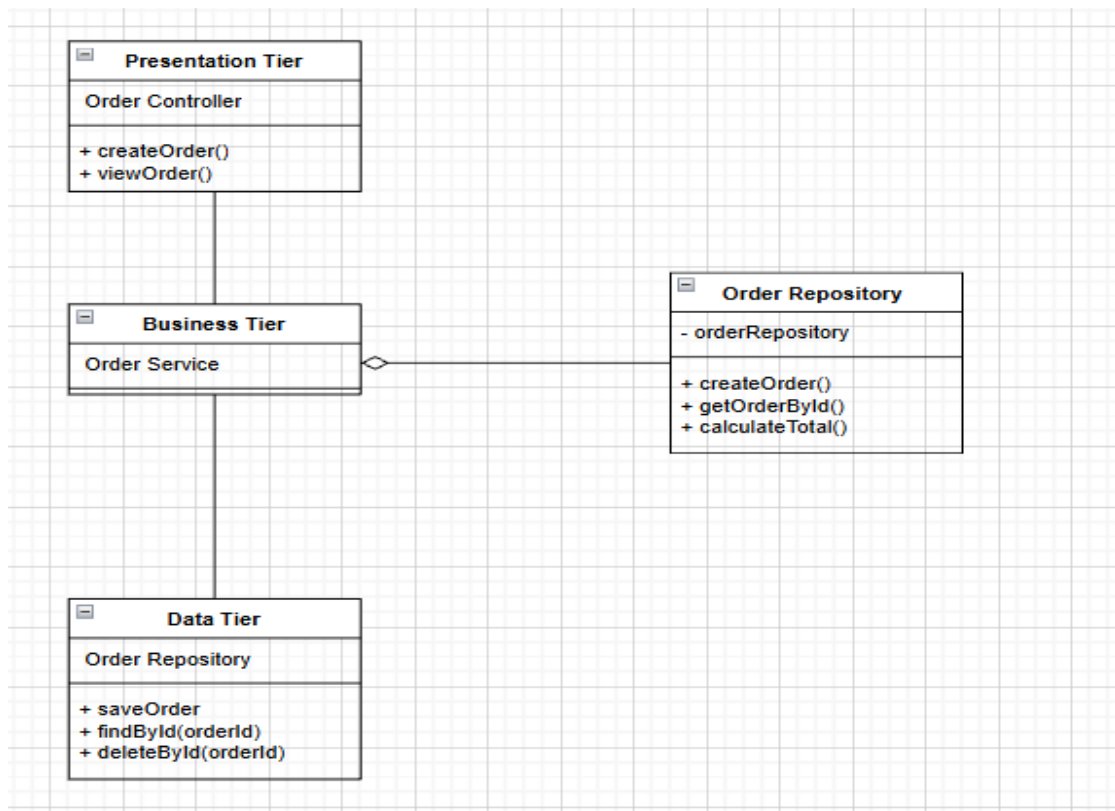


Bài 3:

Sơ đồ package của hệ thống đặt vé máy bay:



Bài 4:



1. Presentation Layer — Lớp OrderController

- **Vai trò:** Giao tiếp giữa người dùng (UI) và tầng Business.
- **Các method:**
 - **createOrder():** nhận yêu cầu từ người dùng để tạo đơn hàng mới (gửi dữ liệu đến **OrderService**).
 - **viewOrder():** hiển thị thông tin đơn hàng cho người dùng (lấy từ **OrderService**)

2. Business Layer — Lớp OrderService

- **Vai trò:** Xử lý toàn bộ **logic nghiệp vụ** của đơn hàng.
- **Các thuộc tính & phương thức:**
 - **orderRepository:** tham chiếu đến lớp truy xuất dữ liệu.
 - **createOrder(orderData):** kiểm tra dữ liệu đơn hàng (số lượng, giá, tồn kho), sau đó gọi **orderRepository.save()** để lưu.
 - **getOrderById(orderId):** Lấy thông tin đơn hàng theo mã bằng cách: gọi đến **orderRepository.findById(orderId)**.
 - **calculateTotal(order):** tính tổng tiền của đơn hàng dựa trên các sản phẩm và số lượng.

3. Data Layer — Lớp OrderRepository

- **Vai trò:** Làm việc trực tiếp với cơ sở dữ liệu (Data Access).
- **Các method:**
 - **save(order):** lưu thông tin đơn hàng vào cơ sở dữ liệu.
 - **findById(orderId):** truy xuất đơn hàng theo ID.

- **delete(orderId)**: xóa đơn hàng khỏi cơ sở dữ liệu.

Bài 5:

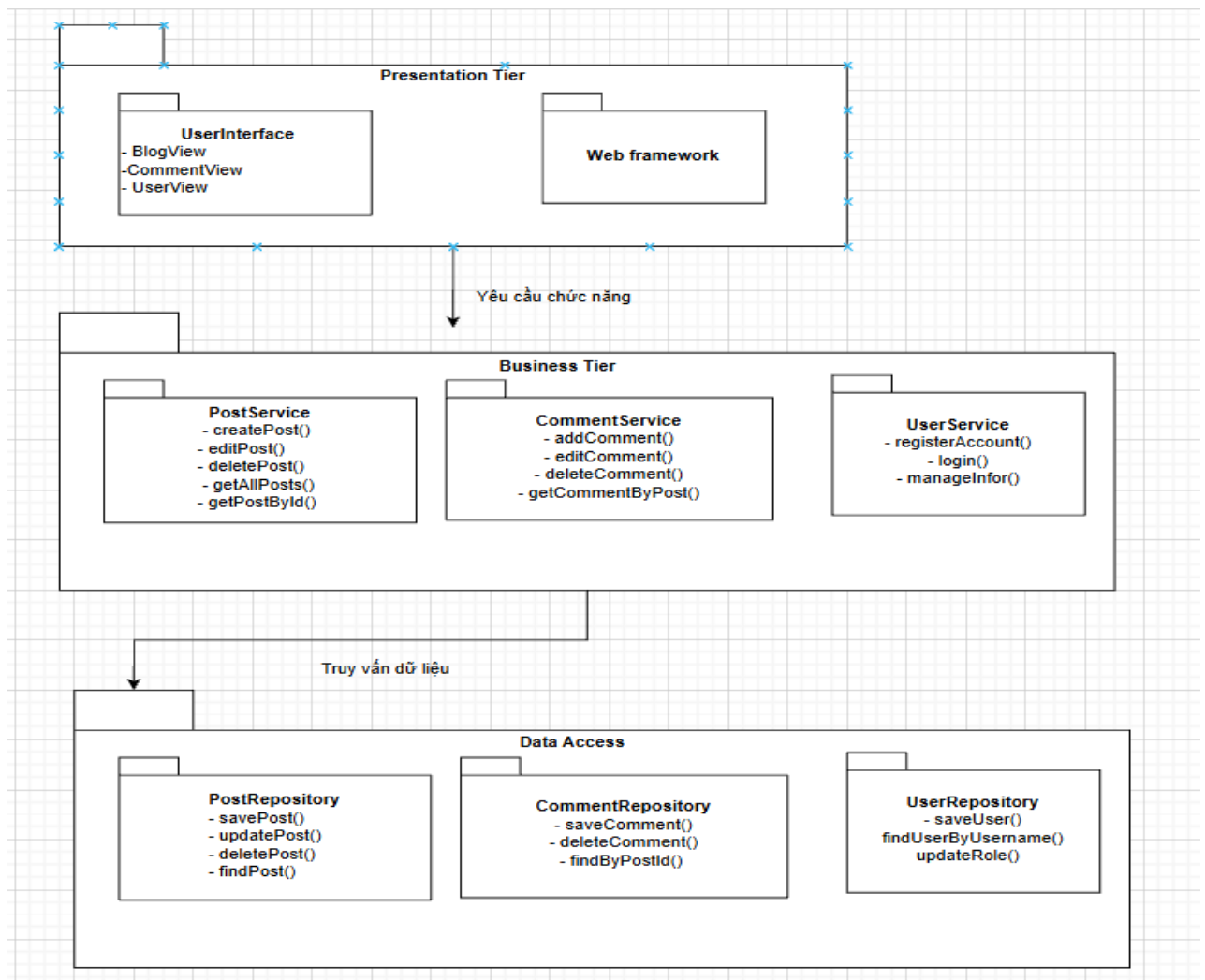
❖ Phân rã hệ thống:

- Presentation:
 - Giao diện hiển thị danh sách bài viết kèm danh sách bình luận; Giao diện hiển thị thông tin người dùng; Giao diện đăng ký, đăng nhập
 - Là nơi để người dùng tương tác với hệ thống và hiển thị thông tin được trả về từ server.
- Business Logic:
 - Nơi nhận yêu cầu từ người dùng và thực hiện logic nghiệp vụ các yêu cầu và trả về kết quả cho người dùng
- Data Access:
 - Cung cấp API để tầng nghiệp vụ gọi đến
 - Xử lý CRUD với dữ liệu
 - Đảm bảo dữ liệu được lưu 1 cách an toàn và bảo mật.

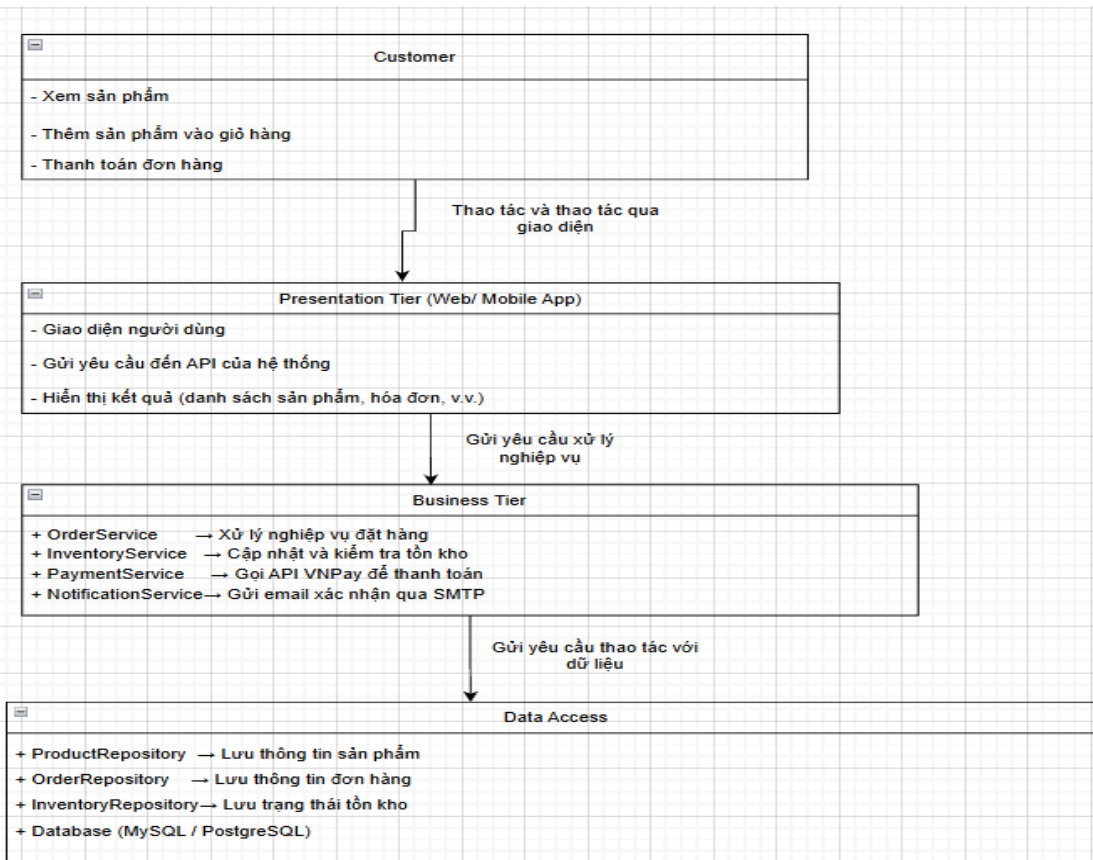
❖ Các module chính:

1. Presentation: nơi người dùng tương tác
 - PostController: Giao diện đọc, thêm, sửa, xóa, chỉnh sửa bài viết
 - CommentController: Giao diện quản lý bình luận
 - UserController: Đăng ký, đăng nhập, quản lý tài khoản người dùng
2. Business:
 - PostController: Xử lý các nghiệp vụ về bài viết
 - CommentController: Xử lý các nghiệp vụ thao tác với comment
 - UserController: Xác thực, nghiệp vụ đăng ký, đăng nhập
3. Data Access:
 - PostRepository: quản lý truy vấn dữ liệu bài viết
 - CommentRepository: quản lý truy cập dữ liệu bình luận
 - UserRepository: quản lý truy vấn dữ liệu người dùng

❖ Package Diagram:



Bài 6:



❖ Mô tả

1. Presentation Tier

- Là tầng giao diện web hoặc mobile.
- Cho phép người dùng:
 - Duyệt sản phẩm.
 - Đặt hàng.
 - Thực hiện thanh toán.
- Giao tiếp với Application Layer thông qua các REST API.

2. Business Logic Tier

- Chứa toàn bộ **logic nghiệp vụ chính**:
 - **OrderService**: Xử lý đơn hàng, xác nhận, tạo mã hóa đơn.
 - **InventoryService**: Kiểm tra và cập nhật số lượng tồn kho sau mỗi giao dịch.
 - **PaymentService**: Gọi **API VNPAY** để xử lý thanh toán online.
 - **NotificationService**: Sau khi thanh toán thành công → gửi email xác nhận thông qua **SMTP**.

3. Data Access Tier

- Giao tiếp với cơ sở dữ liệu.
- Các lớp repository thực hiện CRUD cho:
 - Sản phẩm.
 - Đơn hàng.
 - Dữ liệu tồn kho.

Bài 7:

1. Mapping Use case với chức năng phù hợp:

Đăng ký --- User Management: Quản lý thông tin người dùng, thêm mới tài khoản

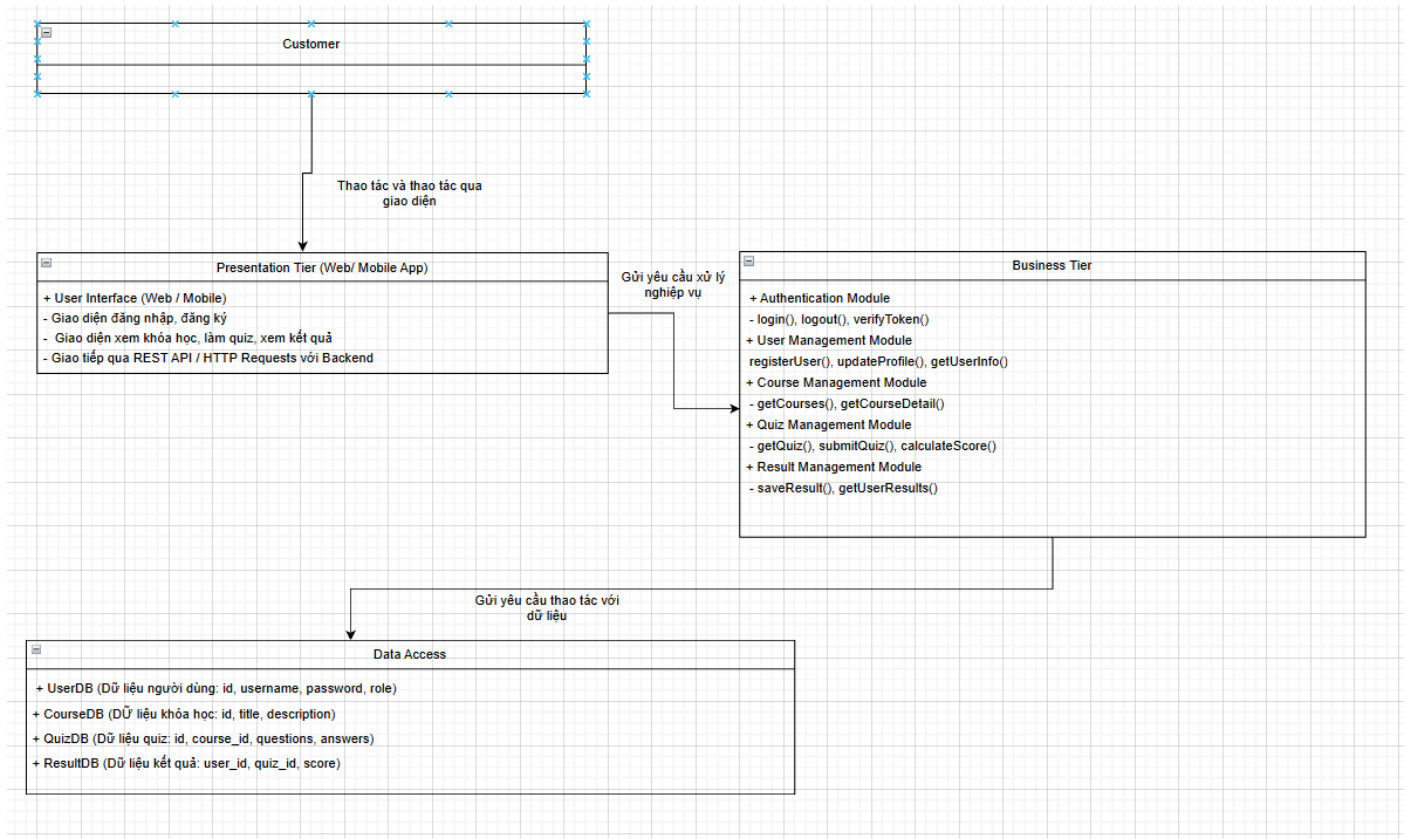
Đăng nhập --- Authentication: Xác thực thông tin người dùng nhập và truy cập hệ thống

Xem khóa học --- Course Management: Quản lý danh sách thông tin khóa học, hiển thị danh sách, thông tin khóa học

Làm bài quizz --- Quizz Management: Quản lý danh sách bài quizz, ghi nhận câu trả lời từ người dùng

Xem kết quả --- Result Management: Ghi nhận và quản lý kết quả làm quizz

2. Sơ đồ



3. Giải thích

- **Frontend:** Hiển thị giao diện và nhận thao tác từ người dùng.
- **Backend:** Xử lý logic đăng nhập, khóa học, quiz, kết quả.
- **Database:** Lưu trữ toàn bộ dữ liệu người dùng, bài học, quiz, điểm số.