**LAB 5 – Thiết kế Kiến trúc Hệ thống**

**1. Tên kiến trúc hệ thống: Kiến trúc Phân tầng 3 Lớp**

**1.1. Mô tả tổng quan**

Kiến trúc 3 Lớp là một mô hình thiết kế phần mềm phổ biến, phân chia ứng dụng thành ba tầng logic và vật lý riêng biệt: Tầng Trình bày (Presentation Tier), Tầng Logic Nghiệp vụ (Logic Tier), và Tầng Dữ liệu (Data Tier). Mỗi tầng có một trách nhiệm cụ thể và có thể được phát triển, triển khai và bảo trì một cách độc lập.

* **Tầng Trình bày (Client):** Chịu trách nhiệm hiển thị giao diện người dùng (UI) và thu nhận tương tác từ người dùng.
* **Tầng Logic Nghiệp vụ (Server):** Xử lý các quy tắc nghiệp vụ, tính toán và điều phối luồng dữ liệu giữa Tầng Trình bày và Tầng Dữ liệu.
* **Tầng Dữ liệu (Database):** Quản lý việc lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu.

**1.2. Lý do lựa chọn**

Nhóm quyết định lựa chọn kiến trúc 3 Lớp cho dự án "Ứng dụng Quản Lý Gara Sửa Chữa" vì những lý do sau:

* Phù hợp với yêu cầu đa nền tảng: Hệ thống yêu cầu phải hoạt động trên Web, Windows và Mobile (Android/iOS). Kiến trúc 3 Lớp cho phép phát triển nhiều loại giao diện (Tầng Trình bày) khác nhau mà không cần thay đổi Tầng Logic Nghiệp vụ và Tầng Dữ liệu.
* Dễ bảo trì và nâng cấp: Việc tách biệt các tầng giúp việc sửa đổi một tầng không ảnh hưởng lớn đến các tầng khác. Ví dụ, có thể nâng cấp giao diện người dùng mà không cần thay đổi logic xử lý nghiệp vụ.
* Tăng cường khả năng mở rộng: Mỗi tầng có thể được mở rộng độc lập. Nếu hệ thống có lượng truy cập lớn, ta có thể tăng cường tài nguyên cho Tầng Logic (Application Server) mà không ảnh hưởng đến Database Server.
* Tăng cường bảo mật: Tầng Trình bày không thể truy cập trực tiếp vào Tầng Dữ liệu. Mọi yêu cầu đều phải đi qua Tầng Logic Nghiệp vụ, nơi các quy tắc bảo mật và xác thực được thực thi, giúp bảo vệ dữ liệu nhạy cảm của khách hàng và gara.

**2. Phân tích các tầng/thành phần**

**2.1. Tầng Trình bày (Presentation Tier)**

* Vai trò: Là lớp giao tiếp trực tiếp với người dùng cuối. Tầng này có trách nhiệm hiển thị dữ liệu cho người dùng và thu thập thông tin đầu vào từ họ. Trong dự án này, Tầng Trình bày sẽ bao gồm các giao diện trên Web, ứng dụng Desktop và ứng dụng Mobile.
* Mối quan hệ: Gửi yêu cầu (HTTP Request) đến Tầng Logic Nghiệp vụ và nhận lại kết quả (HTTP Response) để hiển thị cho người dùng.

**2.2. Tầng Logic Nghiệp vụ (Logic Tier/Application Tier)**

* Vai trò: Là "bộ não" của hệ thống, xử lý tất cả các logic nghiệp vụ phức tạp. Ví dụ:
  + Xác thực đăng nhập của nhân viên.
  + Tính toán tổng chi phí hóa đơn dựa trên tiền công và phụ tùng.
  + Tự động trừ kho khi phụ tùng được sử dụng.
  + Điều phối luồng dữ liệu giữa các tầng khác.
* Mối quan hệ: Nhận yêu cầu từ Tầng Trình bày, xử lý chúng và giao tiếp với Tầng Dữ liệu để truy vấn hoặc cập nhật thông tin.

**2.3. Tầng Dữ liệu (Data Tier)**

* Vai trò: Chịu trách nhiệm lưu trữ, quản lý và truy xuất dữ liệu một cách bền vững. Tầng này bao gồm hệ quản trị cơ sở dữ liệu (ví dụ: MySQL, SQL Server) và các logic truy cập dữ liệu.
* Mối quan hệ: Chỉ nhận yêu cầu từ Tầng Logic Nghiệp vụ và trả về dữ liệu thô. Nó hoàn toàn độc lập và không biết về sự tồn tại của Tầng Trình bày.

**3. Sơ đồ kiến trúc**

**3.1. Hình vẽ sơ đồ**

A diagram of a process

AI-generated content may be incorrect.

**3.2. Giải thích tóm tắt**

Sơ đồ minh họa luồng tương tác trong hệ thống Quản lý Gara:

1. Actor (ví dụ: Nhân viên Lễ tân, Thợ sửa) tương tác với Tầng Trình bày thông qua các thiết bị (Web Browser, App Mobile).
2. Tầng Trình bày gửi một yêu cầu (ví dụ: "tạo phiếu tiếp nhận xe") dưới dạng HTTP Request đến Tầng Logic Nghiệp vụ.
3. Tầng Logic Nghiệp vụ nhận yêu cầu, xử lý logic (kiểm tra dữ liệu, xác thực quyền) và gửi các truy vấn (SQL Query) đến Tầng Dữ liệu.
4. Tầng Dữ liệu thực thi truy vấn và trả kết quả (Data Response) về cho Tầng Logic.
5. Tầng Logic nhận dữ liệu, định dạng lại và gửi kết quả cuối cùng (HTTP Response) về Tầng Trình bày để hiển thị cho người dùng.

**4. Nhận xét**

**4.1. Điểm mạnh**

* Tính linh hoạt: Dễ dàng thay đổi hoặc thêm các loại giao diện mới (ví dụ: ứng dụng cho khách hàng trong tương lai) mà không ảnh hưởng đến phần lõi của hệ thống.
* Phát triển song song: Các nhóm phát triển khác nhau có thể làm việc đồng thời trên các tầng khác nhau (ví dụ: team front-end làm giao diện, team back-end xử lý logic), giúp đẩy nhanh tiến độ dự án.
* Tái sử dụng: Logic nghiệp vụ ở tầng giữa có thể được tái sử dụng bởi nhiều giao diện khác nhau.

**4.2. Hạn chế khi áp dụng vào hệ thống**

* Tăng độ phức tạp: So với kiến trúc nguyên khối (monolithic), việc quản lý và triển khai 3 tầng riêng biệt sẽ phức tạp hơn, đòi hỏi cấu hình mạng và kết nối giữa các tầng phải chính xác.
* Tăng độ trễ (Latency): Việc dữ liệu phải di chuyển qua lại giữa các tầng có thể làm tăng thời gian phản hồi của hệ thống. Tuy nhiên, với quy mô nhỏ và vừa của gara và yêu cầu phản hồi trong 2 giây, hạn chế này có thể kiểm soát được thông qua tối ưu hóa mạng và truy vấn CSDL.