

Trường Đại học Công nghệ - ĐHQGHN  
Khoa Công nghệ thông tin

-----

***BÀI TẬP LỚN: PHÂN TÍCH & THIẾT KẾ HƯỚNG ĐỐI TƯỢNG***  
***Giảng viên: Đặng Đức Hạnh & Trần Mạnh Cường***



**SOFTWARE ARCHITECTURE**  
**GREENMART - CHỢ NÔNG SẢN SẠCH**  
**HÀNG ĐẦU VIỆT NAM**

21020035 - Nguyễn Huy Thái  
21020394 - Kiều Minh Tuấn  
2102055 - Trần Hồng Quân  
21020032 - Đỗ Quốc Tuấn  
21020031- Ngô Văn Tuấn

*Hà Nội, ngày 4 tháng 4 năm 2024*

# Mục lục

Lịch sử sửa đổi.....	3
1. Tổng quan.....	4
1.1. Giới thiệu.....	4
1.2. Mục đích.....	4
1.3. Phạm vi.....	5
1.4. Tham khảo.....	5
2. Mô tả kiến trúc.....	6
2.1. Logical View.....	6
2.2. Process view.....	6
2.3. Implementation view.....	6
2.4. Deployment view.....	6
3. Mục đích và ràng buộc kiến trúc.....	8
3.1. Nền tảng hệ thống.....	8
3.2. Bảo mật.....	8
3.3. Độ tin cậy và tính sẵn sàng.....	9
4. Use-case View.....	10
4.1. Actors.....	11
4.2. Use-case realization.....	11
5. Logical View.....	12
5.1. Tổng quan.....	12
5.2. Architectural Layer Dependencies.....	12
6. Process View.....	14
7. Deployment View.....	15
8. Data View.....	16
9. Quy mô và hiệu suất.....	17
9.1. Quy mô.....	17
9.2. Hiệu suất.....	17
10. Chất lượng.....	18

## Lịch sử sửa đổi

Người sửa đổi	Thời gian	Các thay đổi	Phiên bản
Nguyễn Huy Thái	04/04/2023	Khởi tạo mẫu tài liệu	1.0
Nguyễn Huy Thái	13/04/2023	Hoàn thành tài liệu	1.1

# 1. Tổng quan

## 1.1. Giới thiệu

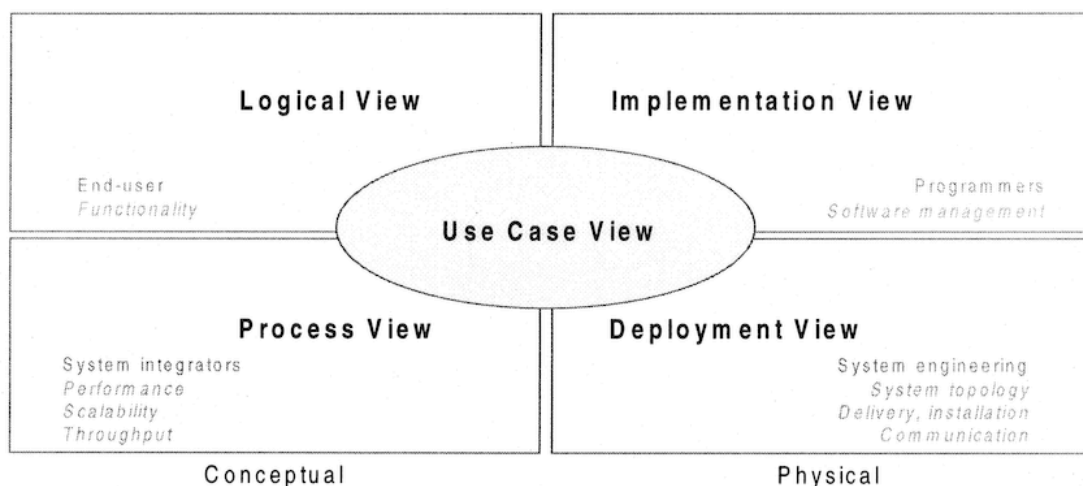
Đây là bài báo cáo cho bài tập lớp học phần Phân tích và thiết kế hướng đối tượng của nhóm sinh viên về phân tích kiến trúc của ứng dụng.

Tài liệu này cung cấp một góc nhìn tổng quan cấp độ cao và giải thích quá trình Hệ thống Quản Lý Khách Sạn hoạt động.

## 1.2. Mục đích

Tài liệu phân tích cung cấp một cái nhìn toàn diện hơn và giải thích chi tiết về kiến trúc của hệ thống **GreenMart - Chợ nông sản sạch hàng đầu Việt Nam**, sử dụng một số góc nhìn kiến trúc khác nhau để mô tả các khía cạnh khác nhau của hệ thống. Tài liệu này nhằm nắm bắt và truyền đạt các quyết định kiến trúc quan trọng đã được thực hiện trên hệ thống.

Để mô tả phần mềm một cách chính xác nhất, cấu trúc của tài liệu này sẽ dựa trên mô hình kiến trúc “4+1” view:



### **1.3. Phạm vi**

Tài liệu phân tích kiến trúc mô tả kiến trúc của hệ thống GreenMart - chợ nông sản sạch hàng đầu Việt Nam được phát triển bởi nhóm sinh viên.

Tài liệu này mô tả các khía cạnh quan trọng về mặt kiến trúc ở thiết kế của hệ thống chợ nông sản, bao gồm các thành phần và hành vi cơ bản nhất cho việc xây dựng nên hệ thống và hiểu dự án này một cách tổng thể. Các bên liên quan cần một cách hiểu kỹ thuật về hệ thống có thể bắt đầu bằng việc đọc tài liệu này, sau đó xem mô hình UML của hệ thống chợ nông sản và cuối cùng xem mã nguồn.

### **1.4. Tham khảo**

- [1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
- [2] Slide môn học Phân tích và thiết kế hướng đối tượng do giảng viên cung cấp.
- [3] Các tài liệu mẫu của giảng viên Nguyễn Tùng Lâm.

## 2. Mô tả kiến trúc

Tài liệu này trình bày chi tiết kiến trúc bằng cách sử dụng các góc nhìn được định nghĩa trong mô hình “4+1” nhưng sử dụng quy ước đặt tên RUP. Các góc nhìn được sử dụng để mô tả Hệ thống chợ nông sản được sử dụng trong tài liệu là:

### 2.1. Logical View

- **Đối tượng:** Nhà thiết kế
- **Phạm vi:** Yêu cầu chức năng: Mô tả mô hình đối tượng của thiết kế, đồng thời mô tả ý nghĩa của những ca sử dụng quan trọng nhất và yêu cầu nghiệp vụ của hệ thống
- **Thành phần liên quan:** Mô hình thiết kế

### 2.2. Process view

- **Đối tượng:** Người tích hợp
- **Phạm vi:** Yêu cầu phi chức năng: mô tả các khía cạnh đồng bộ và đồng bộ hóa của thiết kế.
- **Thành phần liên quan:** Không

### 2.3. Implementation view

- **Đối tượng:** Lập trình viên
- **Phạm vi:** Thành phần ứng dụng: mô tả các lớp và hệ thống con của ứng dụng
- **Thành phần liên quan:** Mô hình cài đặt

### 2.4. Deployment view

- **Đối tượng:** Người quản lý triển khai

- **Phạm vi:** Cấu trúc liên kết: mô tả ánh xạ của phần mềm lên phần cứng và hiển thị các khía cạnh phân tán của hệ thống. Mô tả các cấu trúc triển khai tiềm năng bằng cách đặt ra các trường hợp bao gồm các kịch bản triển khai đã biết và dự đoán trong kiến trúc.
- **Thành phần liên quan:** Mô hình triển khai

### 3. Mục đích và ràng buộc kiến trúc

Phần này mô tả các yêu cầu và mục tiêu phần mềm có tác động đáng kể đến kiến trúc.

#### 3.1. Nền tảng hệ thống

- **Server-side:** Hệ thống chợ nông sản được viết bằng Python Django, sau đó triển khai trên Amazon Web Services, và được kết nối với cơ sở dữ liệu RDS PostgreSQL lưu trên Amazon. Mọi giao tiếp sử dụng giao thức HTTPS với chứng chỉ SSL.
- **Client-side:** Chủ cửa hàng hoặc khách hàng truy cập vào hệ thống thông qua trình duyệt web ở máy tính. Những hệ điều hành có thể sử dụng bao gồm: Windows, MacOS và Linux.

#### 3.2. Bảo mật

Giao thức HTTPS sẽ được sử dụng để tạo điều kiện cho việc truyền thông giữa máy khách và máy chủ, đồng thời hỗ trợ bảo mật hai lớp. Hệ thống phải được bảo mật tuyệt đối để tránh làm rò rỉ thông tin cá nhân của khách hàng.

- Các tính năng bảo mật cơ bản cần cài đặt như:
  - + **Xác thực (authentication):** Khi một người dùng đăng nhập bằng username & password, hệ thống phải kiểm tra trong cơ sở dữ liệu xem thông tin đó có hợp lệ không. Khi lưu trữ thông tin trong cơ sở dữ liệu cũng sẽ cần mã hoá bằng hash để tránh trường hợp bị đánh cắp dữ liệu.
  - + **Phân quyền (authorization):** Khi người dùng đăng nhập hợp lệ, hệ thống phải biết người đăng nhập là ai và cấp quyền, chuyển hướng người dùng sang giao diện hợp lý.
- Về đường truyền cần đảm bảo:
  - + **Bảo mật thông tin:** dữ liệu nhạy cảm phải được mã hóa



- + **Tính toàn vẹn dữ liệu:** Dữ liệu gửi qua mạng không được mất mát hoặc bị thay đổi, chỉnh sửa.
- + **Khả năng kiểm soát:** Mọi hành động nhạy cảm, đáng ngờ đều phải được ghi lại trong nhật ký.

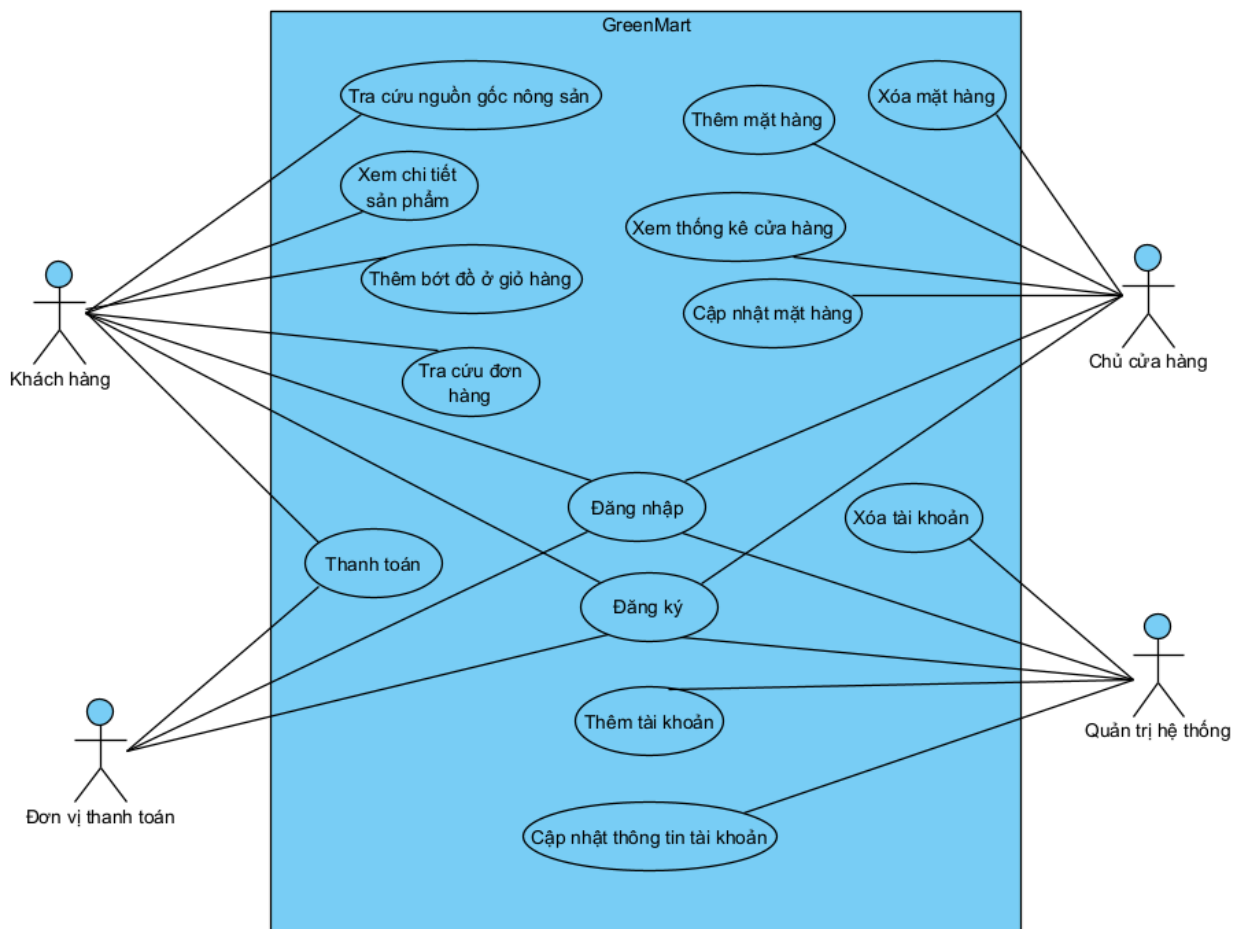
### **3.3. Độ tin cậy và tính sẵn sàng**

Hệ thống phải sẵn sàng phục vụ 24h trong ngày, 7 ngày trong tuần, không được sập quá 5% thời gian một năm.

## 4. Use-case View

Đây là danh sách các use-case model đại diện cho chức năng trung tâm, quan trọng của hệ thống cuối cùng. Các trường hợp sử dụng có tác động đáng kể đến kiến trúc là [UC-Doc]:

- Đăng nhập
- Thêm bớt đồ ở giỏ hàng
- Tra cứu đơn hàng
- Xem thống kê cửa hàng
- Thanh toán
- Cập nhật mật hàng
- Thêm mật hàng



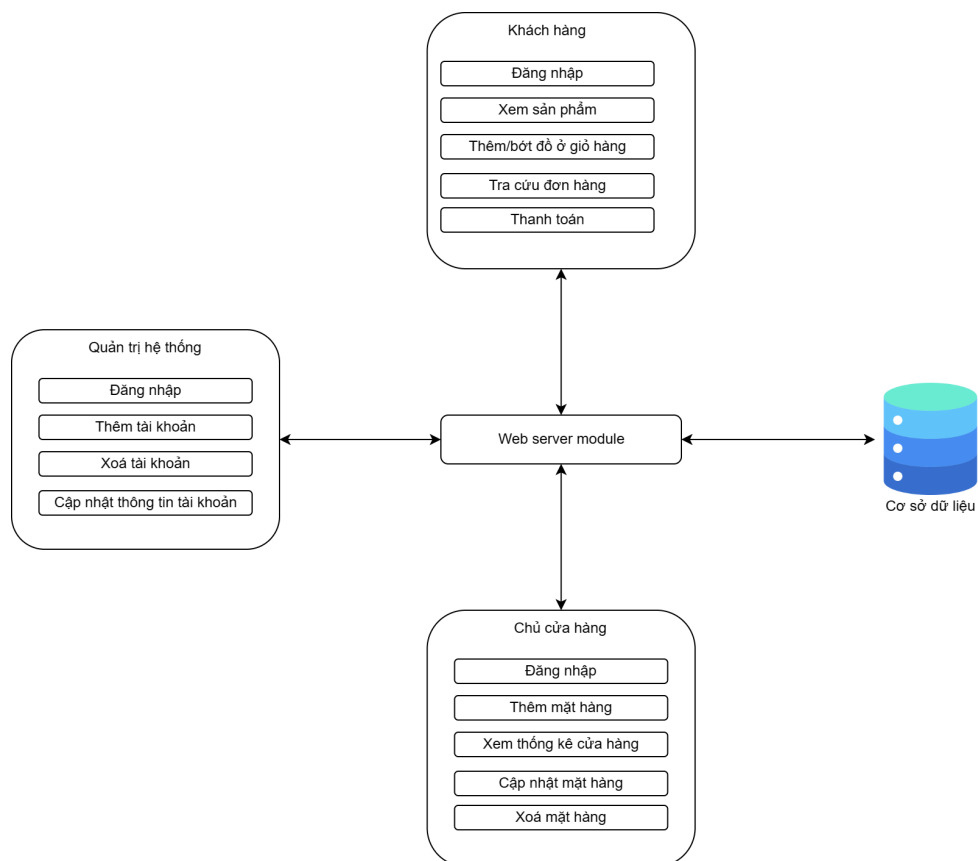
## 4.1. Actors

Các actors có thể chia làm 4 nhóm:

- Chủ cửa hàng
- Quản trị hệ thống
- Khách hàng
- Hệ thống con: đơn vị thanh toán

## 4.2. Use-case realization

Biểu đồ use-case realization cho hệ thống chợ nông sản ở dưới mô tả cách mà hệ thống sẽ cung cấp các chức năng cho các actors sử dụng phần mềm. Các use case được mô tả như các chức năng cho hệ thống. Một chức năng có thể bao hàm một hoặc nhiều use-cases. Biểu đồ cung cấp một cái nhìn tổng quan hơn về thiết kế của hệ thống.

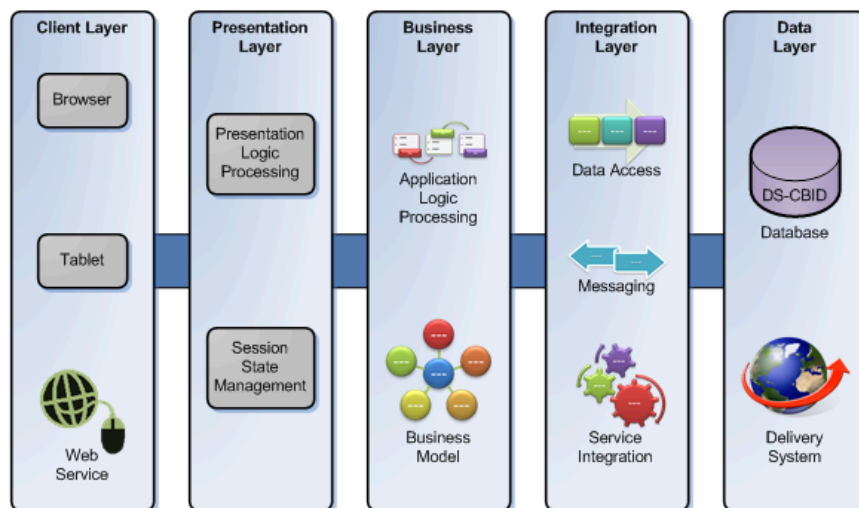


## 5. Logical View

### 5.1. Tổng quan

Hệ thống chia thành các layers dựa trên kiến trúc N-tier, trong đó mô hình phân lớp của hệ thống dựa trên sự liên kết chặt chẽ của các lớp nhỏ với từng trách nhiệm cụ thể. Mô hình này chia nhỏ hệ thống lớn thành các hệ thống con khác nhau nó đảm bảo cả về việc phát triển và bảo trì hệ thống.

#### n-Tier Architecture



### 5.2. Architectural Layer Dependencies

Mỗi tầng có các trách nhiệm cụ thể như sau:

1. Tầng trình diễn - Presentation layer:
  - Hiển thị logic, giao diện người dùng tương tác trên hệ thống.
  - Bao gồm tất cả các thành phần cần thiết để tương tác với người dùng cuối
2. Tầng điều khiển - Control layer:
  - Quản lý quyền truy cập vào tầng nghiệp vụ.
  - Chứa các thành phần chịu trách nhiệm truy cập và tầng nghiệp vụ hoặc truy cập trực tiếp vào tầng tài nguyên.

3. Tầng tài nguyên - Resource layer:

- Nhiệm vụ truy cập vào hệ thống database.
- Chứa các thành phần cầu nối giữa tầng Business và hệ thống thông tin.

4. Tầng nghiệp vụ - Business layer:

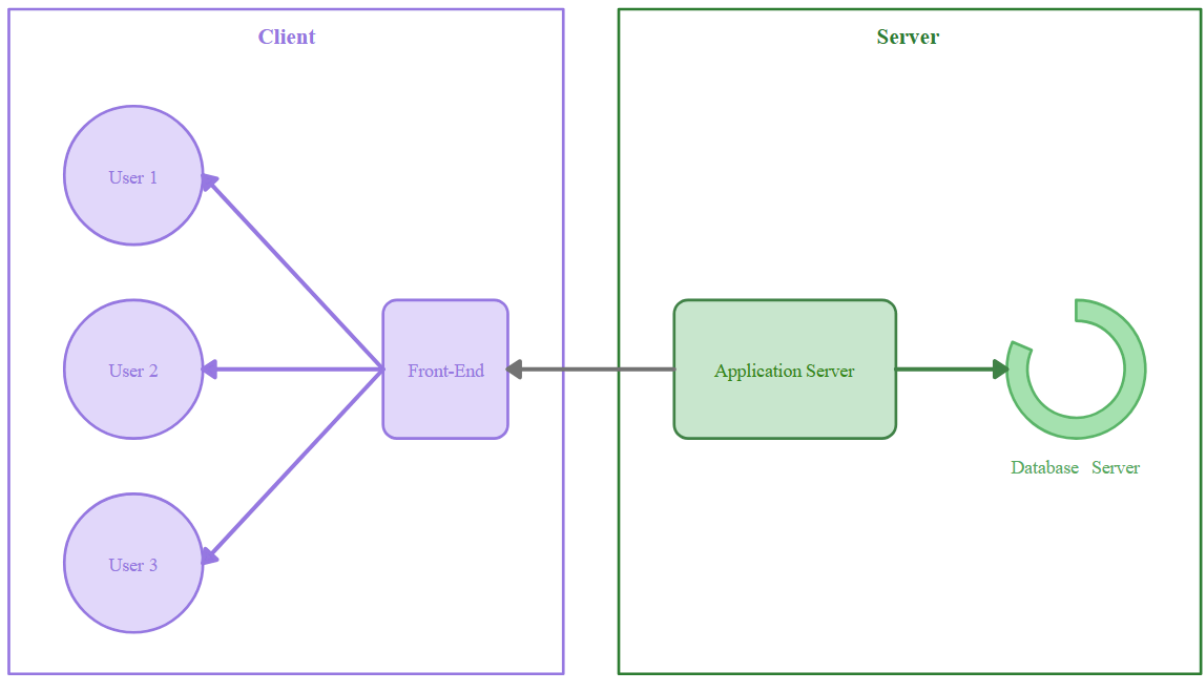
- Xử lý logic nghiệp vụ và quản lý việc truy cập vào Resource Layer.
- Chứa các thành phần chủ chốt về mặt Logic nghiệp vụ.

5. Tầng các phần tử phổ biến - Common elements layer:

- Tập hợp các đối tượng chung được sử dụng lại qua tất cả các lớp như người dùng.
- Chứa các thành phần được sử dụng lại trong một số lớp và có thể được sử dụng trong tương lai

## 6. Process View:

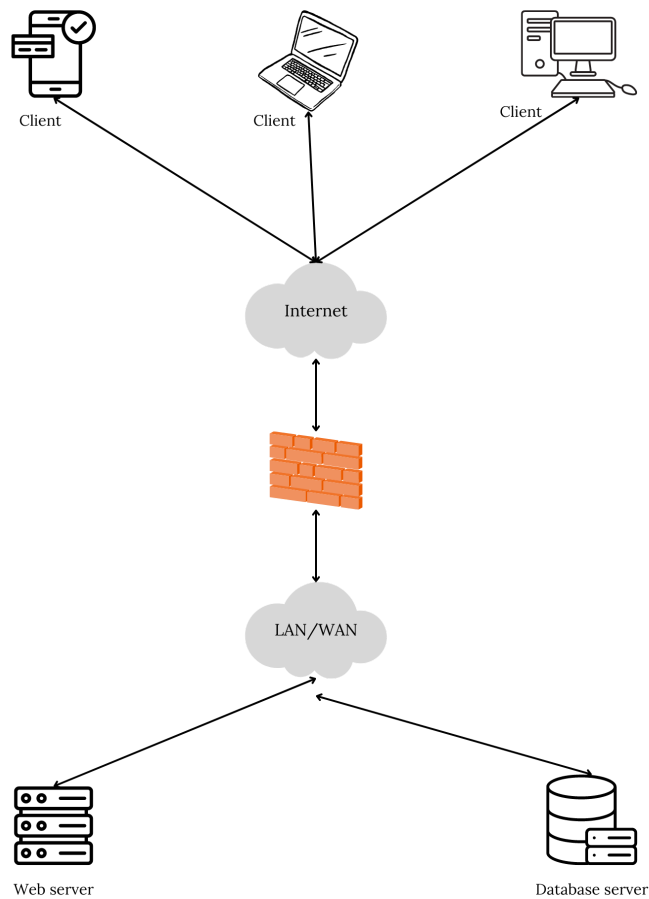
Ta có kiến trúc client-server cơ bản như hình. Hệ thống của AWS sẽ tự động xử lý các luồng và scale-up khi có lượng truy cập đột biến.



Client-Server architecture

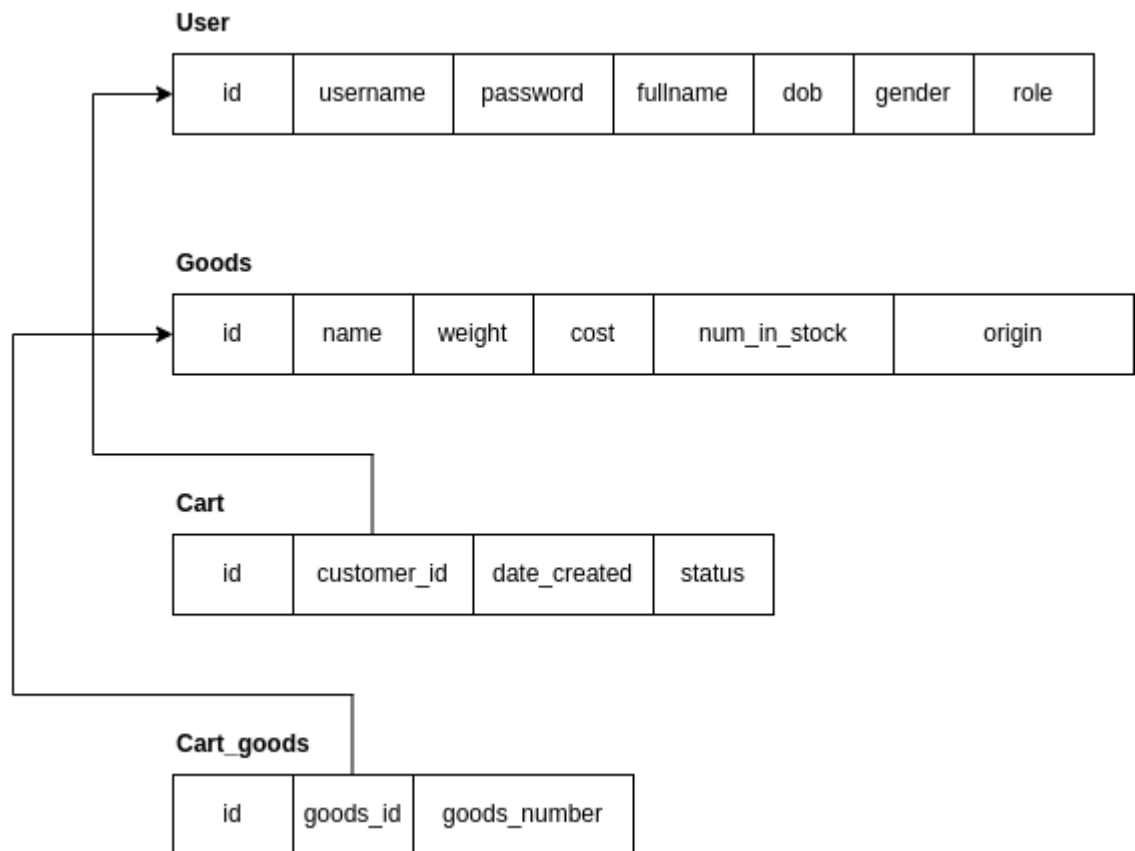
## 7. Deployment View

Khi deploy, tất cả các thành phần phát triển đều sẽ có một vị trí vật lý trên máy chủ và tất cả các thực thể dữ liệu, relationship tương ứng sẽ được đặt trên máy chủ CSDL PostgreSQL.



## 8. Data View

- Ta có biểu đồ Relational Schema của hệ thống như sau





## **9. Quy mô và hiệu suất**

### **9.1. Quy mô**

- Số lượt đặt hàng ước tính: khoảng 1000 lượt mỗi ngày, tập trung vào những giờ cao điểm.
- Hệ thống sẽ hỗ trợ tối thiểu 10.000 yêu cầu mỗi năm.
- Hệ thống sẽ hỗ trợ tối thiểu 500.000 khách hàng.
- Hệ thống sẽ hỗ trợ tối thiểu 1.000 phiên sử dụng mỗi 15 phút

### **9.2. Hiệu suất**

- Hệ thống có khả năng hoàn thành 90% các giao dịch trong chưa đầy 1 phút.
- Thời gian tải thông tin mặt hàng và khách hàng dưới 0.5 giây.
- Khả năng truy cập tới cơ sở dữ liệu và phản hồi chậm không quá 3 giây.

## 10. Chất lượng

- Sản phẩm được thiết kế dưới dạng một ứng dụng web tương thích với các trình duyệt web phổ thông như Microsoft Edge, Google Chrome, Mozilla Firefox, Safari.
- Hệ thống sẽ có giao diện cho phép xem thông tin khách hàng, lịch sử mua hàng, thông tin nhà phân phối, quản lý doanh thu cho chủ cửa hàng.
- Hệ thống có khả năng sao lưu dữ liệu, phục hồi cao khi gặp sự cố, backup mỗi 24h.
- Phiên bản bảo trì sẽ được thống nhất tối đa 2 lần/năm, tùy theo nhu cầu của khách hàng.
- Hệ thống có khả năng phát hiện và ngăn chặn các cá nhân hoặc chương trình trái phép cố gắng truy cập hoặc sửa đổi dữ liệu trên hệ thống.