# LỜI CẢM ƠN

Lời đầu tiên, cho em xin phép được đại diện nhóm dành lời cảm ơn chân thành nhất gửi đến giáo viên hướng dẫn, Ths. Võ Huỳnh Trâm đã gợi ý và đề xuất cho chúng em đề tài này, nhờ đó chúng em đã tích luỹ được nhiều kiến thức và kinh nghiệm hơn trong quá trình thực hiện đề tài. Và trên hết, con cũng xin gửi lời cảm ơn đến cha mẹ đã luôn luôn bên cạnh, động viên, nhắc nhỡ, khuyến kích và là hậu phương vững chắc cho con trên con đường học tập. Chúng em xin cảm ơn các anh chị khoá trên trong Câu lạc bộ Tin học Khoa Công nghệ thông tin và Truyền thông đã cho chúng em lời khuyên trong quá trình thực hiện khi chúng em gặp khó khăn.

Cần Thơ, ngày 11 tháng 12 năm 2020

Người viết

Đào Minh Trung Thuận

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc58554379)

[1 GIỚI THIỆU 3](#_Toc58554380)

[1.1 MỤC TIÊU 3](#_Toc58554381)

[1.2 PHẠM VI SẢN PHẨM 3](#_Toc58554382)

[1.3 CHÚ THÍCH THUẬT NGỮ 3](#_Toc58554383)

[1.4 TÀI LIỆU THAM KHẢO 3](#_Toc58554384)

[2 MÔ TẢ TỔNG QUAN 4](#_Toc58554385)

[2.1 BỐI CẢNH SẢN PHẨM 4](#_Toc58554386)

[2.2 CÁC CHỨC NĂNG CỦA SẢN PHẨM 4](#_Toc58554387)

[2.2.1 Web Service 4](#_Toc58554388)

[2.2.2 Webapp bán hàng 4](#_Toc58554389)

[2.3 ĐẶC ĐIỂM NGƯỜI SỬ DỤNG 4](#_Toc58554390)

[2.3.1 Web Service 5](#_Toc58554391)

[2.3.2 Webapp bán hàng 5](#_Toc58554392)

[2.4 MÔI TRƯỜNG VẬN HÀNH 5](#_Toc58554393)

[2.5 CÁC RÀNG BUỘC THỰC THI VÀ THIẾT KẾ 5](#_Toc58554394)

[2.6 CÁC GIẢ ĐỊNH VÀ PHỤ THUỘC 5](#_Toc58554395)

[3 CÁC YÊU CẦU GIAO TIẾP BÊN NGOÀI 5](#_Toc58554396)

[3.1 GIAO DIỆN NGƯỜI SỬ DỤNG 5](#_Toc58554397)

[3.2 GIAO TIẾP PHẦN CỨNG 5](#_Toc58554398)

[3.3 GIAO TIẾP PHẦN MỀM 5](#_Toc58554399)

[3.4 GIAO TIẾP TRUYỀN THÔNG TIN 5](#_Toc58554400)

[4 CÁC TÍNH NĂNG CỦA HỆ THỐNG 5](#_Toc58554401)

[5 CÁC YÊU CẦU PHI CHỨC NĂNG 5](#_Toc58554402)

[6 CÁC YÊU CẦU KHÁC 5](#_Toc58554403)

# GIỚI THIỆU

## ĐẶT VẤN ĐỀ

Trong thời đại bùng nổ Internet, mua hang trực tuyến qua mạng trở nên phổ biến và rộng rãi. Tuy nhiên với nhu cầu đa dạng về nền tảng mua sắm cũng đặt lên hàng đầu, cụ thể là mua sắm trên nhiều thiết bị, nền tảng như máy tính để bàn, trên trình duyệt web, trên ứng dụng điện thoại… Nhưng dù cho nhiều nền tảng thì vẫn phải đảm bảo được tính thống nhất hệ thống, đám ứng đầy đủ nhu cầu cho người dung tư chuyển đổi các nên tảng để sử dụng.

Để giải quyết được vấn đề đã đặt ra phía trên, thì cần phải có một hệ thống lớn được cài đặt riêng, phục vụ truy cập từ các nền tảng khác. Và để có thể cho hệ thống và các nên tảng đó có thể giao tiếp được với nhau, mặc dù khác cách thức xây dựng, môi trường phát triển, cài đặt thì chúng phải được thống nhất về cách thức gửi và nhận nhận dữ liệu, cách truyền tải dữ liệu. Bằng cách xây dựng một mô hình dịch vụ web (Web Service) sẽ đám ứng được yêu cầu đã đặt ra.

## mỤC TIÊU

Mục tiêu phát triển của đề đài phải đáp ứng giải quyết được yêu cầu bài toán đặt ra, xây dự một mô hình máy chủ dịch vụ. Từ đây các nên tảng khác như web, ứng dụng trên máy tính, trên điện thoại… có thể gọi yêu cầu đến máy chủ thực hiện.

## ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đối với mô hình web service:

Phía Server sẽ xây dựng một hệ thống được chạy riêng luôn luôn lắng nghe yêu cầu từ Client. Ở đây, Server sẽ được phát triển bằng ngôn ngữ PHP 7, sử dụng Xampp làm Host ảo để lắng nghe yêu cầu từ Client.

Về phía Client, có thể xây dựng bất kỳ nên tảng nào, miễn có thể xây dựng cách thức giao tiếp với Server đúng với yêu cầu đặt ra là HTTP API Restful. Cho đơn giản và hiểu quả, thì Client sẽ được phát triển trên nền tảng Web sử dụng framework NuxtJS.

Về phía quản trị cơ sở dữ liệu, để tăng tính thống nhất với Server, nên thiết kế bằng hệ quản trị cơ sở dữ liệu MySQL MariaDB.

## NỘI DUNG NGHIÊN CỨU

Cài đặt Xampp, mở cổng và chạy được trên máy tính cục bộ. Cài đặt môi trường phát triển PHP và NodeJS (vì framework NuxtJS cần NodeJS để làm host).

Tìm hiểu cách thức hoạt đoạt động mô hình Web Service, tìm hiểu cách giao tiếp dữ liệu giữa các nền tảng bằng API Restful.

Xây dựng, phát triển Server nhận yêu cầu Client thực thi và trả kế quả. Server đóng vài trò tương tác với cơ sở dữ liệu.

Xây dựng, phát triển Website bán hàng trực tuyến gọi yêu cầu đến Server và dựa vào kết quả trả về để xử lý yêu cầu từ người dung.

## ĐÓNG GÓP CHÍNH CỦA ĐỀ TÀI

Bằng việc tách riêng Server và Client, mỗi thành phần sẽ đảm bảo tính chuyên biệt cạo, dễ dàng thay đổi, nâng cấp, ngoài ra còn có thể giảm tải cho hệ thống, vì giờ đây, việc thực thi nghiệp vụ sẽ được chia đề cho cả hai phía. Service có thể tương tác với nhiều nên tảng Client khác nhau, điều này làm tăng tính đa dạng và thực hiện những yêu cầu chuyên sâu hơn cho từng loại người dùng mà không bị gò bó bất kì điều gì về nên tảng phát triển, hay ngôn ngữ, môi trường thực thi.

# MÔ TẢ BÀI TOÁN

Xây dựng, phát triển một Web Service giao tiếp thông qua HTTP API Restful

* Server: Dịch vụ Web, được phát triển bằng PHP, MySQL. Nhận yêu cầu từ Client, thực thi và trả lại kết quả.
* Client: Website bán hàng trực tuyến phát triển bằng framework NuxtJS trên môi trường NodeJS. Gửi yêu cầu đến Server, nhận kết quả và thực thi yêu cầu từ khách hàng.

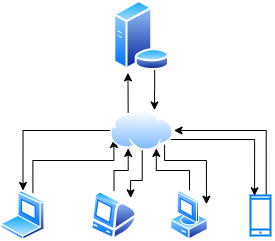
Chi tiết về Website bán hàng, loại mặt hàng là máy tính.. Hệ thống gồm 2 người dùng là Người quản trị (Admin) và người dùng thường (User):

* Người dùng thường đại diện cho khách hàng, có thể xem hàng, mua hàng, đánh giá sản phẩm đã mua.
* Admin đóng vai trò như người lý cửa hàng, được phép quản lý tài nguyên cửa hàng, xem lịch sử truy cập dịch vụ đến Server, xác nhận các đơn hàng từ khách hàng.

# THIẾT KẾ, CÀI ĐẶT

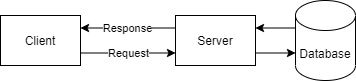
## Thiết kế

### Mô hình dịch vụ



Hình 1: Mô hình Web Service

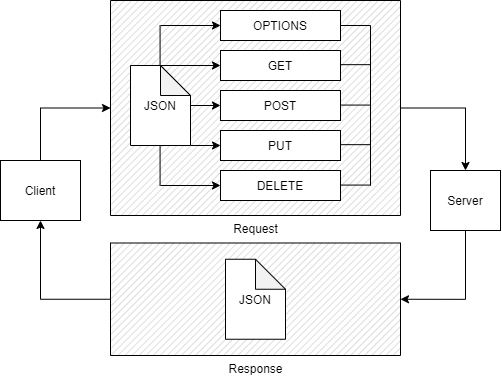
Xây dụng được một Web Service phụ vụ được bất kì nền tàng nào, chỉ cần nền tảng đó được xây dựng giao tiếp đúng theo yêu cầu mà Server và Client đã đặt ra.



Hình 2: Tương tác Server – Client

Client sẽ gửi yêu cầu Request đến Server, Server sẽ xử lý yêu cầu đó, có thể sẽ tương tác đến cơ sở dữ liệu, sau khi xử lý xong sẽ trả về Response cho Client.

Client và Server sẽ giao tiếp với nhau thông qua một chuẩn thống nhất. Ở đây sẽ sử dụng API Restful làm phương thức giao tiếp. Dữ liệu được truyền qua lại sẽ được chuyển về dạng JSON. Như vậy, dù cho là nên tảng nào thì Client và Server đều có thể giao tiếp được miễn có hỗ trợ giao thức HTTP.



Hình 3: Giao tiếp HTTP API Restful giữa Server và Client

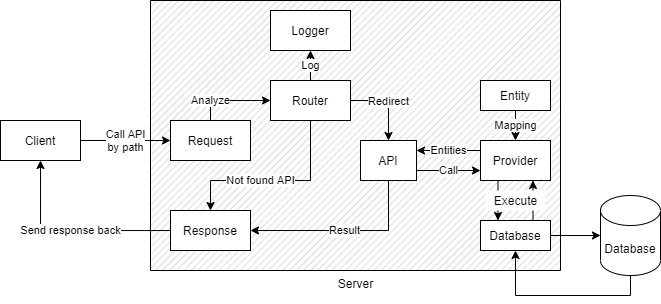
### Kiến trúc hệ thống

#### Server

Server sẽ chịu trách nhiệm luôn luôn lắng nghe kết nối đến từ Client và thực thi yêu cầu. Để cho đơn giản, ta sử dụng Host Server cục bộ của Xampp. Từ đó, chỉ cần phát triển điều hướng, xử lý nghiệp vụ yêu cầu từ Client.

Các thành phần có trong Server:

* Phần lõi (Core): Gồm các thành phần đóng vai trò cấu hình Server, sử lý luồng dữ liệu, những mã nguồn điều khiển Server. Bên trong gồm các thành phần con:
  + Service: thành phần chính gọi các thành phần con trong lõi.
  + Session: quản lý phiên làm việc của Client.
  + Router: điều hướng yêu cầu của Client.
  + Request: nhận và phân tích yêu cầu của Client
  + Response: chịu trách nhiệm gửi kết quả về cho Client
  + Database: chịu trách nhiệm tương tác cơ sở dữ liệu
  + Entity: mapping dữ liệu từ cơ sở dữ liệu về các đối tượng
  + Api: cấu hình đường dẫn dịch vụ
  + Http: cấu hình giao thức
  + File: quản lý file trong Server
  + Logger: để ghi vết truy cập dịch vụ
* Api: nơi chứa các Api của Server
* Entity: nơi chứa các cấu hình thực thể khi mapping từ cơ sở dữ liệu
* Provider: chức các bộ vận chuyển dữ liệu từ cơ sở dữ liệu
* Log: nơi chứa lịch sử ghi vết
* Tài sản (Asset): chứa nguồn tài nguyên của Server, gồm hình ảnh, bài đăng...
* Tiện ích (Plugin): các tiện ích của bên thứ ba, ví dụ như gửi mail, phân tích html...



Hình 4: Hoạt động của các thành phần trong Server

Khi Client gửi một yêu cầu đến Server, Service sẽ gọi Request để tiếp nhận yêu cầu, sau khi đã phân tích yêu cầu sẽ được chuyển đển Router để điều hướng. Tại đây, Router sẽ ghi vết dịch vụ thông qua Logger. Nếu không tìm thấy API nào phù hợp cho yêu cầu thì Router sẽ gửi Response về cho Client với trạng thái 404. Nếu tìm thấy API phù hợp thì sẽ chuyển dữ liệu đến phương thức mà Request đã yêu cầu của API đó để xử lý. Tại API, sẽ thực hiện xử lý nghiệp vụ, tương tác với cơ sở dữ liệu (nếu có) thông qua bộ vận chuyển dữ liệu Provider. Provider sẽ sử dụng Entity để ánh xạ các bảng trong cơ sở dữ liệu và sử dụng thành phần Database để truy vấn.

#### Client

Kiến trúc Client được xây dựng trên framework NuxtJS gồm các thành phần

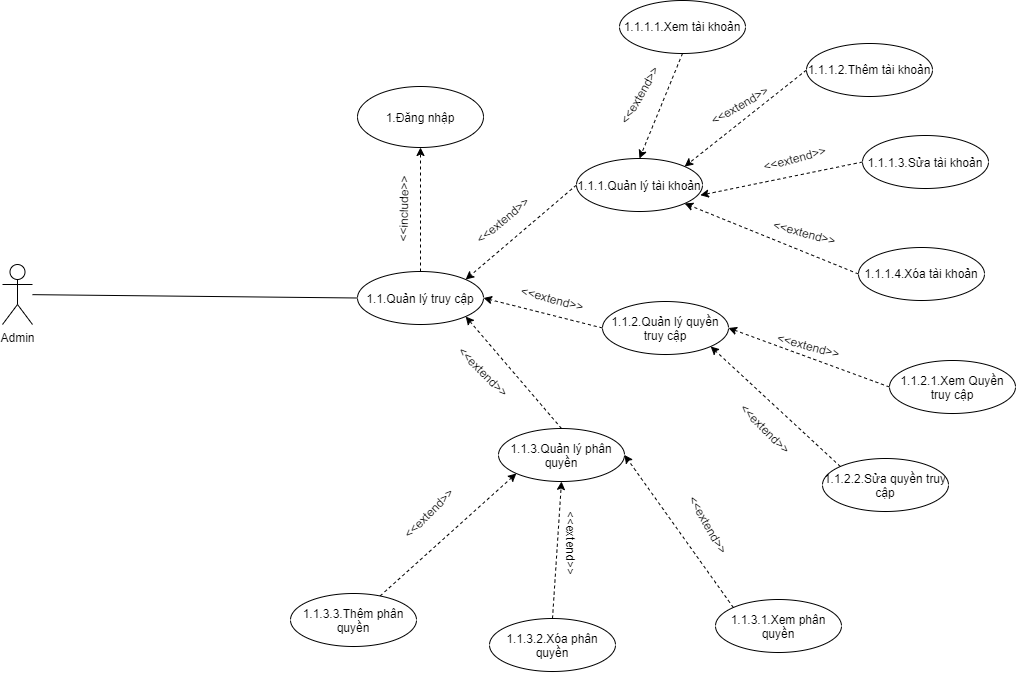
* Asset: chức các tài sản client (các file scss, script, các file đầu vào để render khi chạy hệ thống)
* Component: chứa các thành phần con được sử dụng lại
* Layout: chứa khung chung của trang web
* Page: chứa các trang của web ứng với từng đường dẫn
* Plugin: chứa các file sciprt tiện ích được viết thêm hỗ trợ
* Static: chứa các file tĩnh của Client
* Store: chứa bộ lưu trữ dữ liệu Vuex
* Type: chứa các file định nghĩa kiểu dữ liệu cho typescript

### Thiết kế chức năng

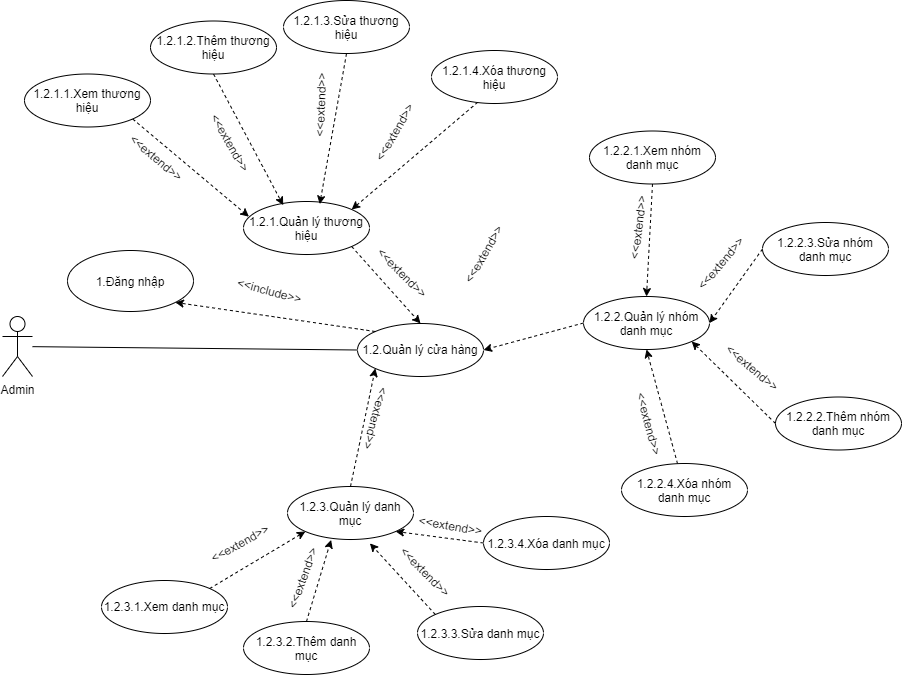
#### Sơ đồ trường hợp sử dụng

Hệ thống gồm có 2 tác nhân tương ứng với hai loại người dùng: Admin và User.

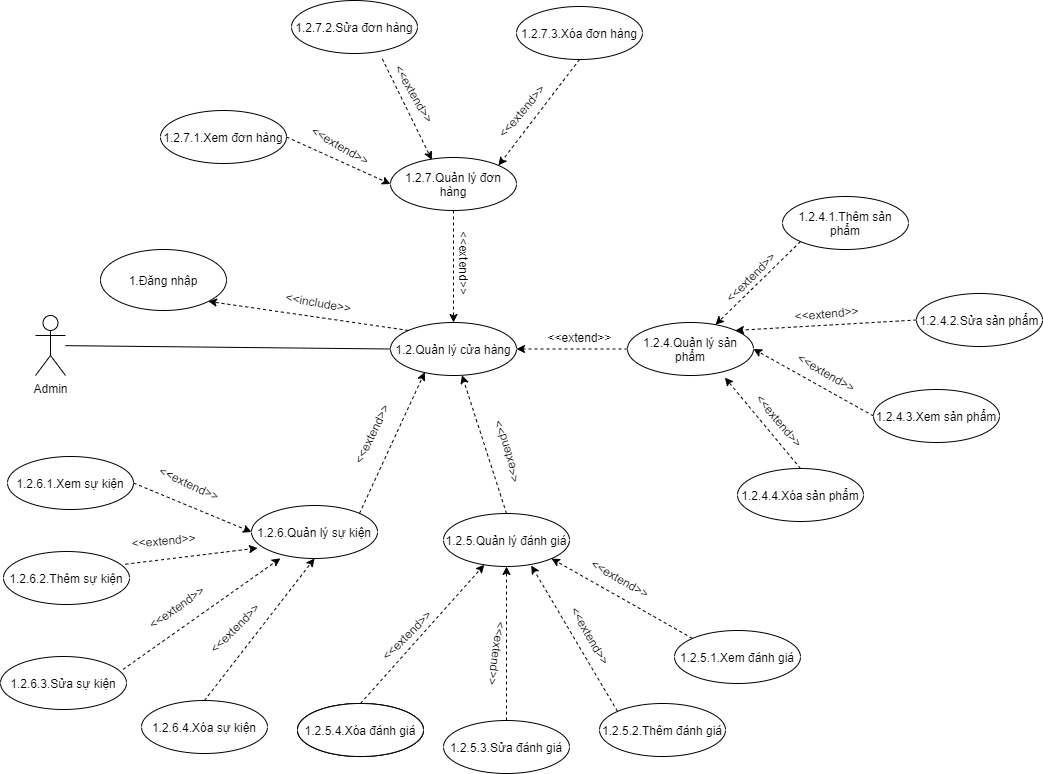
* Đổi với Admin: gồm các chức năng
  + Quản lý truy cập (quản lý tài khoản, quản lý quyền truy cập, quản lý phân quyền).
  + Quản lý cửa hàng (quản lý thương hiệu, quản lý nhóm danh mục, quản lý danh mục, quản lý sản phẩm, quản lý đánh giá, quản lý sự kiện, quản lý đơn hàng).
  + Quản lý ghi vết (quản lý truy cập dịch vụ).
  + Ngoài ra Admin kế thừa User, có tất cả chức năng của User
* Đối với User:
  + Xem hàng hoá
  + Mua hàng
  + Quản lý giỏ hàng của mình
  + Quản lý đơn hàng của mình
  + Đánh giá sản phẩm



Hình 5: Sơ đồ trường hợp sử dụng quản lý truy cập



Hình 6: Sơ đồ trường hợp sử dụng quản lý cửa hàng



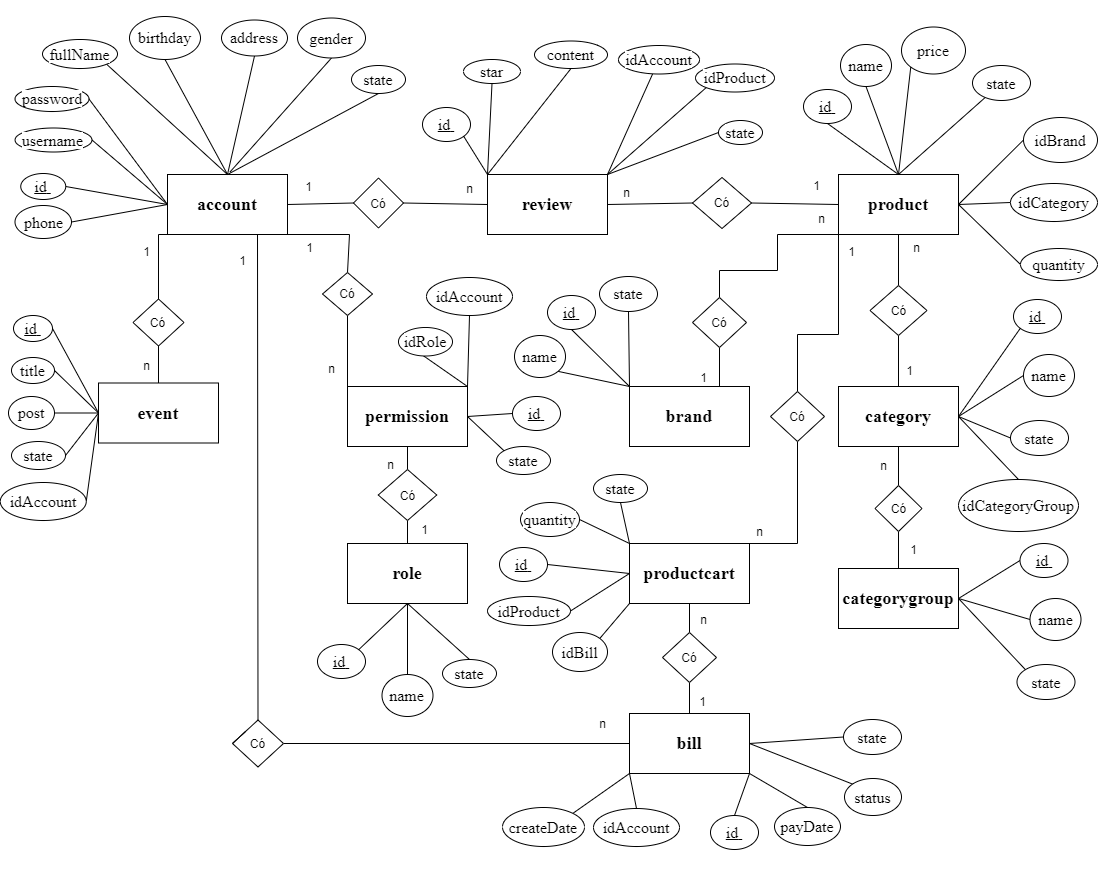
Hình 7: Sơ đồ trường hợp sử dụng quản lý cửa hàng



Hình 8: Sơ đồ trường hợp sử dụng của admin và user

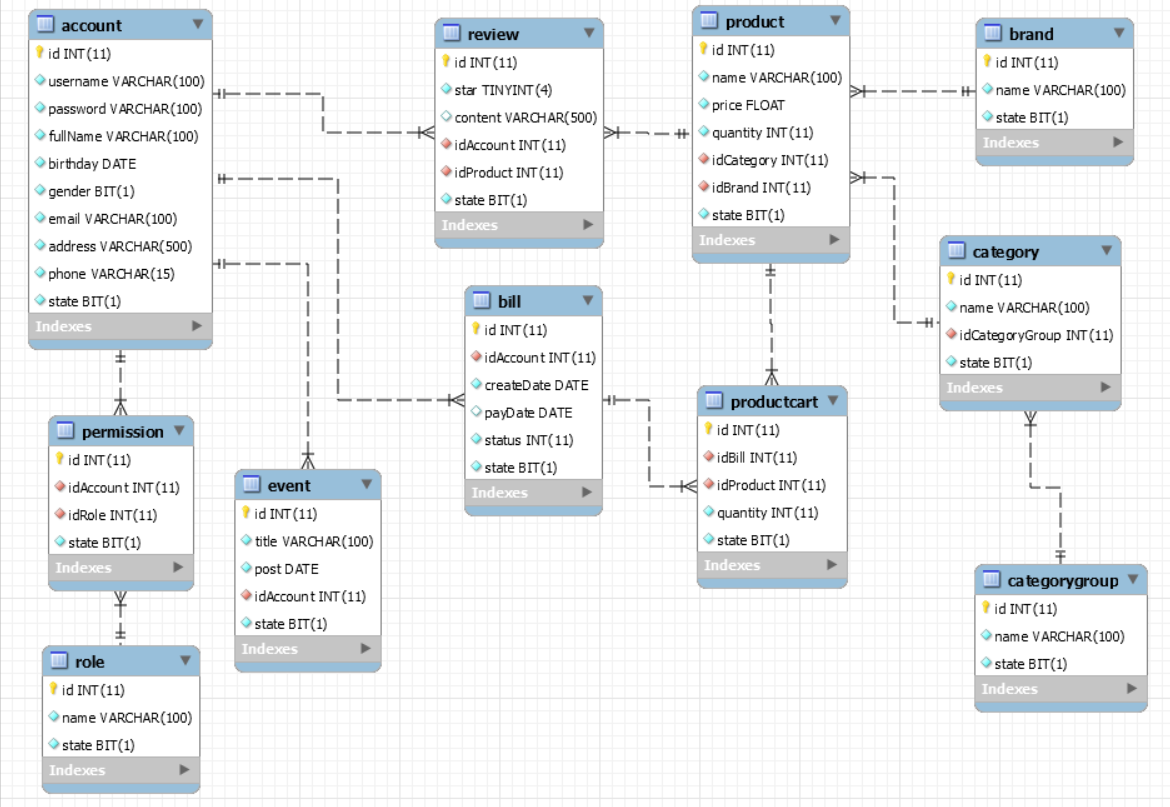
### Thiết kế dữ liệu

#### Mô hình hoá dữ liệu



Hình 9: Mô hình quan hệ thực thể

#### Thiết kế cơ sở dữ liệu



Hình 10: Mô hình thực thể liên kết

# KIỂM THỬ VÀ ĐÁNH GIẤ

# KẾT LUẬN

# PHỤ LỤC