

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO BÀI TẬP
MÔN HỌC: LẬP TRÌNH CẤU TRÚC

Nhóm sinh viên thực hiện:

Họ tên	MSSV
1. Vũ Đức Nguyễn	20162999
2. Nguyễn Việt Hưng	20166256
3. Phùng Thị Trang	20165645
4. Lê Đức Anh	20167462

Giảng viên hướng dẫn: TS. Vũ Thị Hương Giang

Hà Nội, tháng 12 năm 2019

Mục lục

Lời mở đầu.....	2
1. Giới thiệu đề tài	3
2. AS Group Assignment topic 1-3.....	4
3. AS Group Assignment 4.....	14
4. AS Group Assignment 5.....	19
5. AS Group Assignment 6.....	29
6. AS Group Assignment 7.....	35
7. Hướng dẫn sử dụng	37
8. Hướng dẫn cài đặt, triển khai, phát triển	42
9. Phân công công việc.....	44
10.Kết luận.....	45
11. Tài liệu tham khảo.....	45

Lời mở đầu

Lời đầu tiên, chúng em xin được chân thành cảm ơn sự giúp đỡ, giảng dạy nhiệt tình của cô Vũ Thị Hương Giang. Nhờ những kiến thức trong môn học lập trình cấu trúc mà cô đã nhiệt tình chỉ dạy trên những giờ học trên lớp, chúng em đã hoàn thành được các bài tập và phần công việc được giao của nhóm: Quản lý sản phẩm trong khuôn khổ báo cáo môn học.

Mặc dù chúng em đã làm bài tập theo các yêu cầu đề ra, tuy nhiên những thiếu sót trong quá trình thực hiện và tổng hợp là không thể tránh khỏi, rất mong nhận được những ý kiến đóng góp, chỉ bảo từ cô để chúng em có thể nắm vững hơn các vấn đề đặt ra của môn học.

Chúng em xin chân thành cảm ơn!

Hà Nội, tháng 12 năm 2019

1. Giới thiệu đề tài

Đề tài là cần phải xây dựng 1 website bán hàng. Website được chia thành 11 modules nhỏ tương ứng với 11 nhóm, các nhóm phát triển theo phương pháp Agile theo 3 vòng.

Nhóm chúng em được phân công module "Quản lý sản phẩm". Dưới đây là báo cáo các bài tập tuần cùng với hướng dẫn sử dụng, hướng dẫn cài đặt, triển khai, kiểm thử của module "Quản lý sản phẩm".

2. AS - Group Assignment topic 1-3

1.1 Topic 1

I. Identify the must-have functionalities of the selling website.

- Function 1: Quản lý tài khoản
- Function 2: Quản lý sản phẩm
- Function 3: Quản lý đơn hàng
- Function 4: Quản lý giỏ hàng, thanh toán
- Function 5: Tìm kiếm, phân tích, thống kê
- Function 6: Quảng cáo - khuyến mãi, khuyến mại

II. How to deal with complexity in the following cases:

a, Developing new selling website.

- Đối tượng sử dụng hệ thống là khách hàng nên khi phát triển trang web mới cần thực hiện khảo sát, thu thập ý kiến từ khách hàng để khách hàng có thể hài lòng khi sử dụng trang web mới.
- Vì hệ thống có nhiều người sử dụng nên cần phân chia quyền hạn
- Giao diện sử dụng rõ ràng, dễ hiểu và đẹp

b. Activating the website in the whole system configuration.

- Tính năng cũ có thể không phù hợp hay bất tiện, có thể có yêu cầu sửa lại. Trong quá trình sửa đổi cần chú ý dữ liệu đầu vào, nếu hệ thống cũ có function quá phức tạp thì nên viết mới hoàn toàn.
- Sử dụng nền tảng mới là phiên bản update của hệ thống cũ để duy trì tính ổn định của hệ thống
- Backup dữ liệu cũ của hệ thống, có cơ chế tiếp nhận dữ liệu từ người dùng khi vẫn đang tích hợp lại hệ thống.

c. Reusing a search engine and a member administration module of the main system.

- Cần kiểm tra lại tính ổn định và chính xác của mô đun cũ
- Nhu cầu sử dụng môi trường mới để tăng hiệu suất sử dụng
- Xây dựng giao diện mới theo những mô đun cũ có sẵn.
- Có thể nâng cấp tốc độ truyền tải hoặc tối ưu hóa truy xuất dữ liệu trong mô đun cũ.

III. How to deal with changes in the following cases:

a. Developing new selling website.

- Tùy vào sự thay đổi về sản phẩm mà có cách hiển thị khác nhau

nên cần thay đổi cấu trúc dữ liệu với mỗi sản phẩm nhưng vẫn đảm bảo không có trục trặc xảy ra.

- Thay đổi về chức năng mà khách hàng yêu cầu xử lý các hàm bên trong, thường xuyên trao đổi với khách hàng để không xảy ra các xung đột.

b. Activating the website in the whole system configuration.

- Tận dụng những tài nguyên sẵn có để giảm bớt gánh nặng về tài chính khi thực hiện cấu hình hệ thống.
- Yêu cầu về thay đổi địa điểm cài đặt máy chủ cần nhanh chóng giải quyết vấn đề và lỗi thay đổi hệ điều hành hay hệ thống mạng.

IV. Identify the quality indicators of the website in the following cases

- Kiểm tra thông tin của khách hàng có được lưu chính xác hay không?
- Kiểm tra đơn hàng có được lưu đúng như những gì đã chọn hay không?
- Khách hàng kiểm tra thông tin về đơn hàng đã đặt.
- Khả năng tương tác của người dùng với hệ thống.
- Khả năng sử dụng của hệ thống
- Độ tin cậy của hệ thống

1.2 Topic 2

I. Clarification of the website's core functionalities

1. Đăng ký tài khoản

- Đăng ký tài khoản bằng email, password hoặc sử dụng bên thứ 3 như google để đăng ký
- Kiểm tra thông tin form có chính xác hay không, hoặc người dùng đã tồn tại chưa
- Thông báo về trạng thái thành công hay thất bại của quá trình tạo tài khoản người dùng

2. Đăng nhập

- Dùng tài khoản đã đăng ký đăng nhập vào hệ thống
- Kiểm tra xem thông tin có chính xác hay không hoặc tài khoản đã tồn tại hay chưa
- Thông báo đăng nhập thành công hay thất bại

3. Lấy lại mật khẩu

- Lấy lại mật khẩu người dùng

- Nhập username và password, kiểm tra xem username đó có tồn tại hay không, sau đó gửi email đến địa chỉ email đó với yêu cầu xác nhận, email kèm password mới của người dùng.

4. Tìm kiếm sản phẩm

- Tìm kiếm sản phẩm theo tên, chủng loại hoặc tìm trực tiếp trên danh sách các sản phẩm.

- Đưa ra được các sản phẩm mà khách hàng muốn.

5. Thêm sản phẩm mà người dùng muốn vào giỏ hàng

- Người dùng thực hiện thêm các sản phẩm mà mình muốn vào trong giỏ hàng.

- Thông tin về sản phẩm được truy xuất từ cơ sở dữ liệu sau đó khách hàng có thể xem thông tin chi tiết của sản phẩm.

6. Chức năng thanh toán hóa đơn

- Thanh toán thông qua thẻ tín dụng hoặc bằng tiền khi giao dịch.

- Bên bán thực hiện xác minh các thông tin mà khách hàng đưa sau đó xuất đơn hàng để thực hiện chuyển hàng đến người mua.

II. The assement criteria of website

- Khi nhận đầu vào đúng format, hàm trả về đầu ra như kì vọng và giống trong đặc tả.

- Xác suất xảy ra lỗi tối thiểu, không phát sinh thêm khi sử dụng lâu hoặc lượng người dùng tăng lên.

- Hệ thống luôn sẵn sàng phục vụ dù gặp sự cố phần cứng hay đầu vào sai.

- Tối ưu hóa sử dụng tài nguyên .

- Thiết kế hệ thống thân thiện, các chức năng dễ tìm kiếm , sử dụng.

- Khả năng bảo mật

- Phần mềm có thể chạy trên các nền tảng phần cứng hoặc phần mềm, môi trường khác nhau.

- Quy trình phát triển được chuẩn hóa, khả năng tích hợp các mô đun nhanh chóng.

- Sản phẩm đạt kỳ vọng về chất lượng đúng như kế hoạch hoặc trước dự định.

III. Users's requirements

1.Function

- Đăng ký

- Đăng ký thông thường, nhập thông tin người dùng để đăng ký
- Đăng ký bằng tài khoản bên thứ 3
- Xác thực email
- Cập nhật thêm thông tin chi tiết sau đăng ký
- Đăng nhập
 - Nhập username,password
 - Nhập mã đăng nhập được gửi đến điện thoại
 - Đăng nhập bằng tài khoản bên thứ 3
 - Nhận thông báo khi đăng nhập sai
 - Có thể yêu cầu thay đổi mật khẩu
- Reset password
 - Yêu cầu gửi email xác nhận về việc reset password
 - Nhập password mới và gửi yêu cầu update mật khẩu
 - Nhận thông báo thành công hoặc thất bại của quá trình reset password
- Trang tìm kiếm sản phẩm
 - Khi vào trang phải show những sản phẩm bán chạy nhất lên trước- Khi hover chuột vào ảnh sản phẩm có thể xem thông tin cơ bản của sản phẩm
 - Có đường dẫn tới các sản phẩm tiếp theo paginate cho trang tìm kiếm
 - Có hộp input keyword, checkbox category và thanh slider tùy chỉnh mức giá mong muốn
 - Có thể tìm kiếm sản phẩm bằng cách kết hợp cả 3 tiêu chí, sản phẩm được tìm thấy cũng cần paginate nếu quá nhiều
- Trang thông tin chi tiết về sản phẩm
 - Các thông tin về thông số viết thành bảng tách biệt với review của cửa hàng
 - Có button thêm sản phẩm vào giỏ hàng kèm số lượng
 - Có kèm hình ảnh các sản phẩm ở sidebar
 - Bên dưới mỗi sản phẩm có thông tin review của khách hàng
- Trang được thiết kế dạng bảng bao gồm tên, ảnh, số lượng và tổng thanh toán
 - Mỗi record hàng có các button để remove sản phẩm khỏi giỏ hàng
 - Có thể thay đổi số lượng mỗi sản phẩm trong giỏ hàng
 - Tổng số tiền đơn hàng đã tính thuế

2. Quality of service

- Hiệu suất thực hiện

- Thời gian đáp ứng cho nhu cầu sử dụng của khách hàng, tìm kiếm sản phẩm.

- Độ tin cậy
- Tính khả dụng của hệ thống

3. Policy

- Chính sách hủy đơn hàng
- Chính sách đổi trả hàng

4. System

- Khả năng bảo trì
- Tính chính xác

IV. Concerns

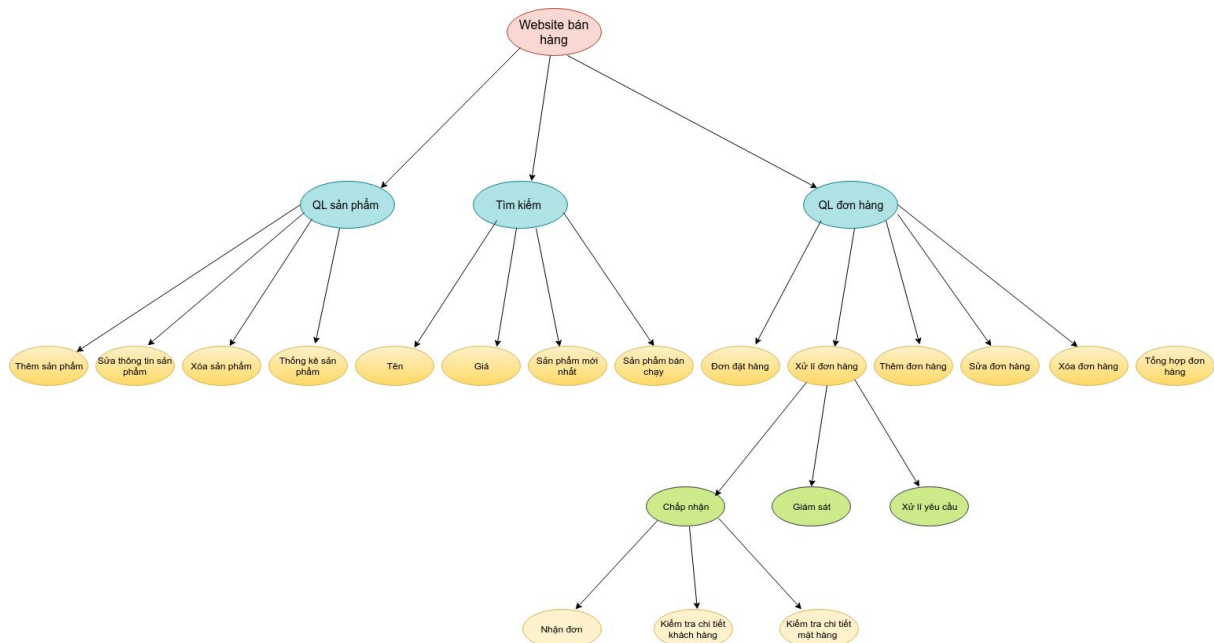
1. Quality

- Bảo mật : đảm bảo không đánh mất thông tin người dùng, sử dụng nhiều cách thức bảo mật khác nhau.
- Tính chính xác: Thông tin người dùng chính xác, cơ chế tính toán tiền của hệ thống chính xác, địa chỉ người gửi người nhận , thời gian đổi trả hàng cũng chính xác.

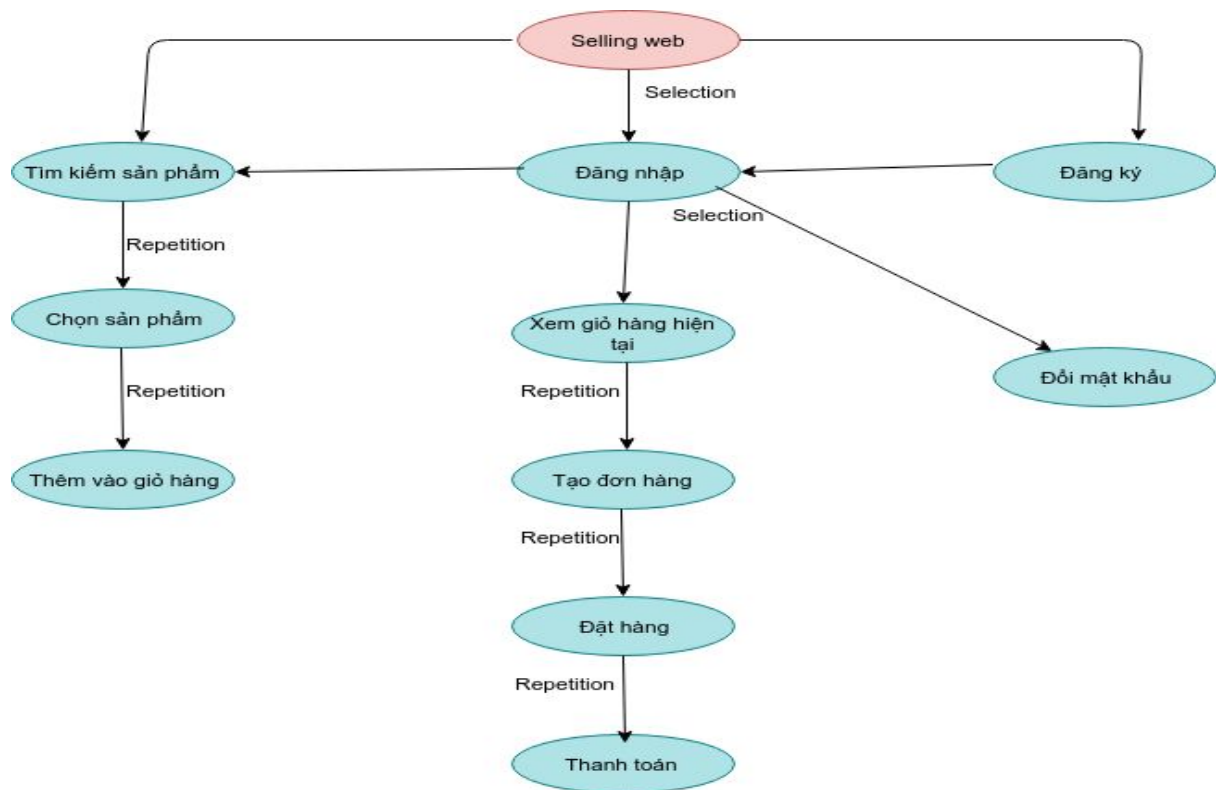
1.3 Topic 3

I. Decompose the selling website (i.e. online selling software) using three approaches: functional, algorithmic and object-oriented.

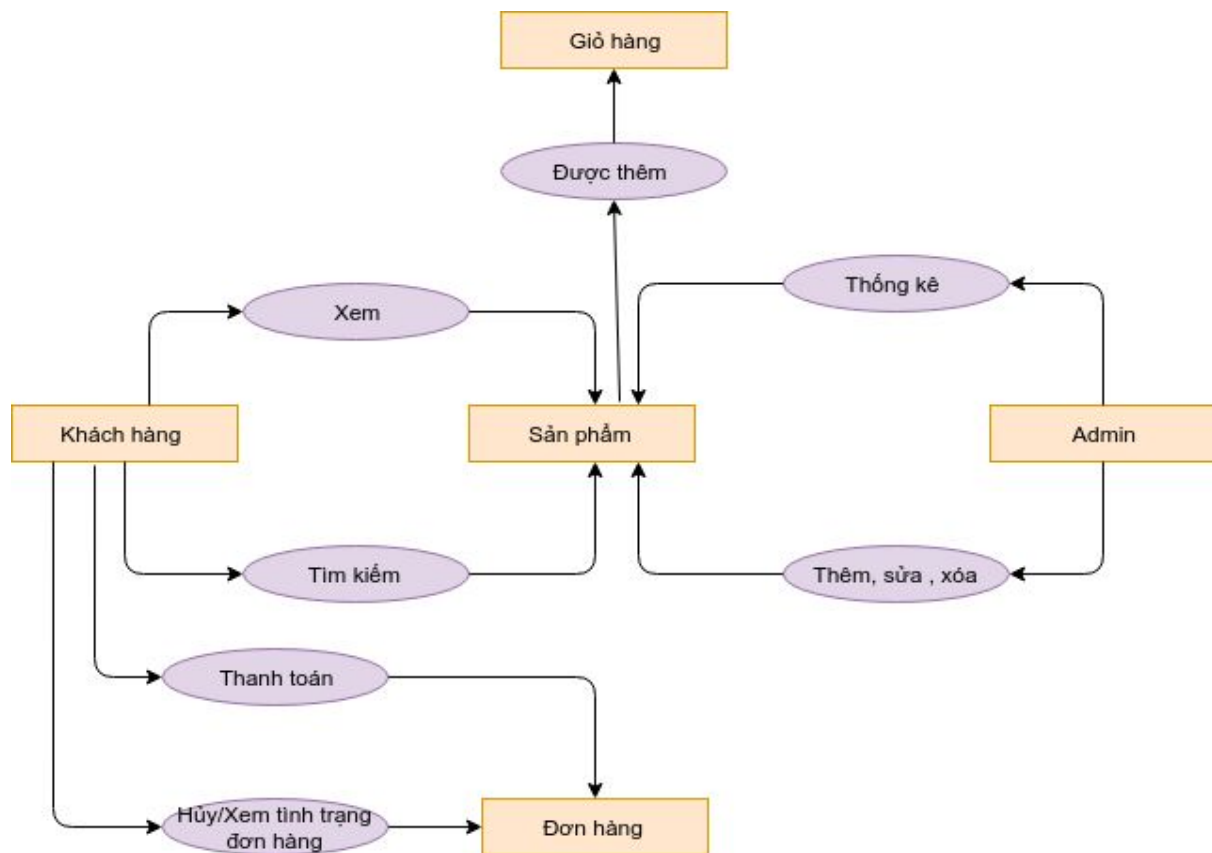
- Functional decompose



- Algorithmic decompose



- Object-oriented



II. Apply the modularization techniques for designing the online selling software. Point out the modularization techniques that are suitable to cope with the changes identified in the 1st in-class assignment. Explain.

2.1 Thiết kế phần mềm bán hàng trực tuyến.

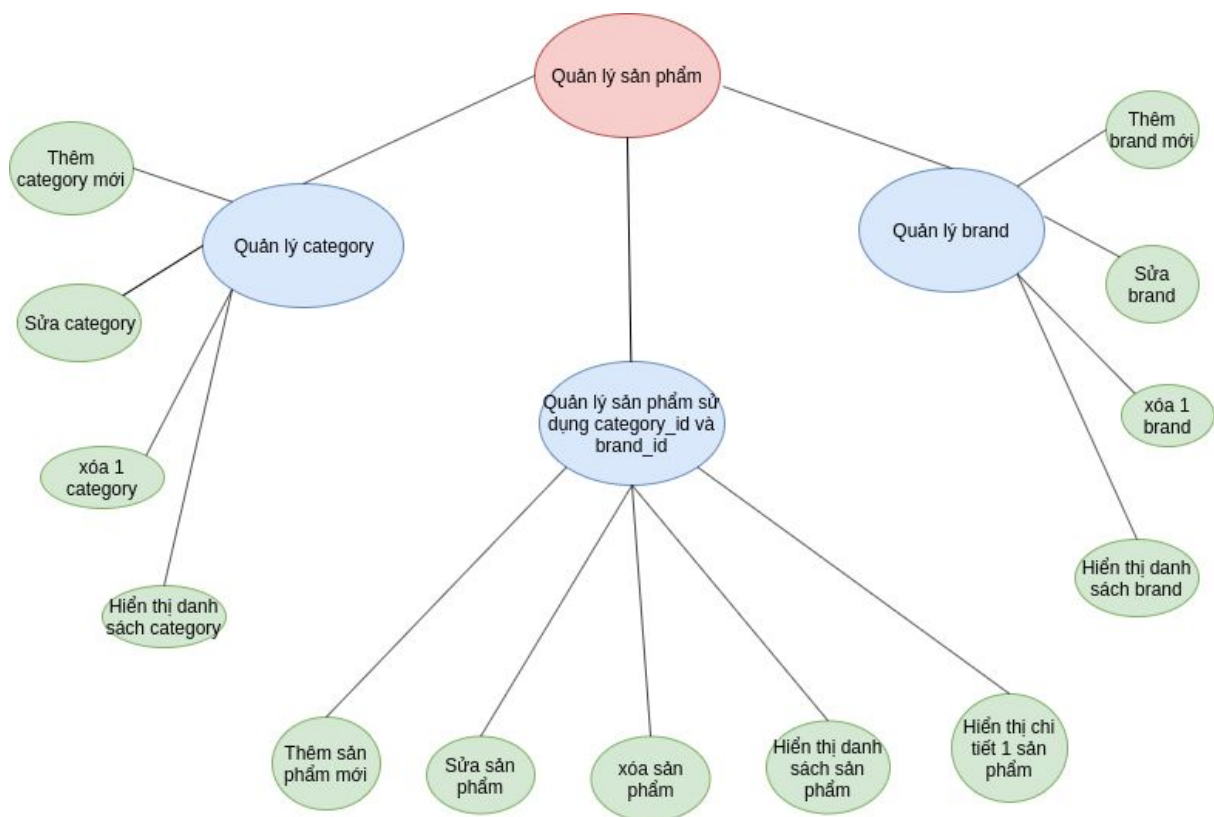
Các module của website:

- Quản lý sản phẩm
- Tìm kiếm
- Quản lý tài khoản
- Quản lý giao hàng
- Quản lý kho
- Quản lý giỏ hàng, thanh toán
- Quản lý đơn hàng
- Quảng cáo
- Quản trị cấu hình
- Liên kết với dữ liệu bán hàng
- Chăm sóc khách hàng

Trong module "Quản lý sản phẩm" có thể chia nhỏ thành các module sau:

- Quản lý sản phẩm sử dụng category_id và brand_id
 - Mô tả ngắn: module thực hiện chức năng thêm, sửa, xóa sản phẩm vào trong cơ sở dữ liệu
 - Input: Thông tin sản phẩm:
 - Output: 1 - sản phẩm được lưu, hoặc cập nhật, hoặc xóa khỏi trong cơ sở dữ liệu hoặc thông báo nếu xảy ra lỗi
- Quản lý category
 - Mô tả ngắn: module thực hiện chức năng thêm, sửa, xóa category vào trong cơ sở dữ liệu
 - Input: thông tin category
 - Output: category được lưu hoặc cập nhật, hoặc xóa khỏi cơ sở dữ liệu hoặc thông báo nếu xảy ra lỗi
- Quản lý brand
 - Mô tả ngắn: module thực hiện chức năng thêm, sửa, xóa brand vào trong cơ sở dữ liệu
 - Input: Thông tin brand
 - Output: brand được lưu hoặc cập nhật, hoặc xóa khỏi cơ sở dữ liệu hoặc thông báo nếu xảy ra lỗi

Hình vẽ minh họa:



2.2 Giải thích

- Kỹ thuật module hóa: Function Modules, Chia website thành các module nhỏ với các chức năng riêng biệt (chia để trị), áp dụng nguyên tắc ẩn thông tin (information hiding).

- Giải thích: Website sẽ được chia thành các module lớn, trong mỗi module lớn có thể gồm các module nhỏ hơn, như thế dễ quản lý, có thể tái sử dụng,...Áp dụng nguyên tắc "information hiding" chỉ cần đưa ra "module interface" và có thể che dấu các thông tin, các xử lý bên trong của module. Mỗi module có thể được phát triển độc lập và dễ dàng đối phó với sự thay đổi hơn.

- Ví dụ: + Khi khách hàng yêu cầu thêm chức năng thêm sản phẩm từ file excel thì chỉ cần tạo thêm module "thêm sản phẩm từ excel" rồi tích hợp vào hệ thống và hệ thống vẫn chạy bình thường.

- + Khi khách hàng muốn hiển thị danh sách sản phẩm kèm số lượng và nhà cung cấp, chỉ cần thay đổi trong module hiển thị danh sách sản phẩm.

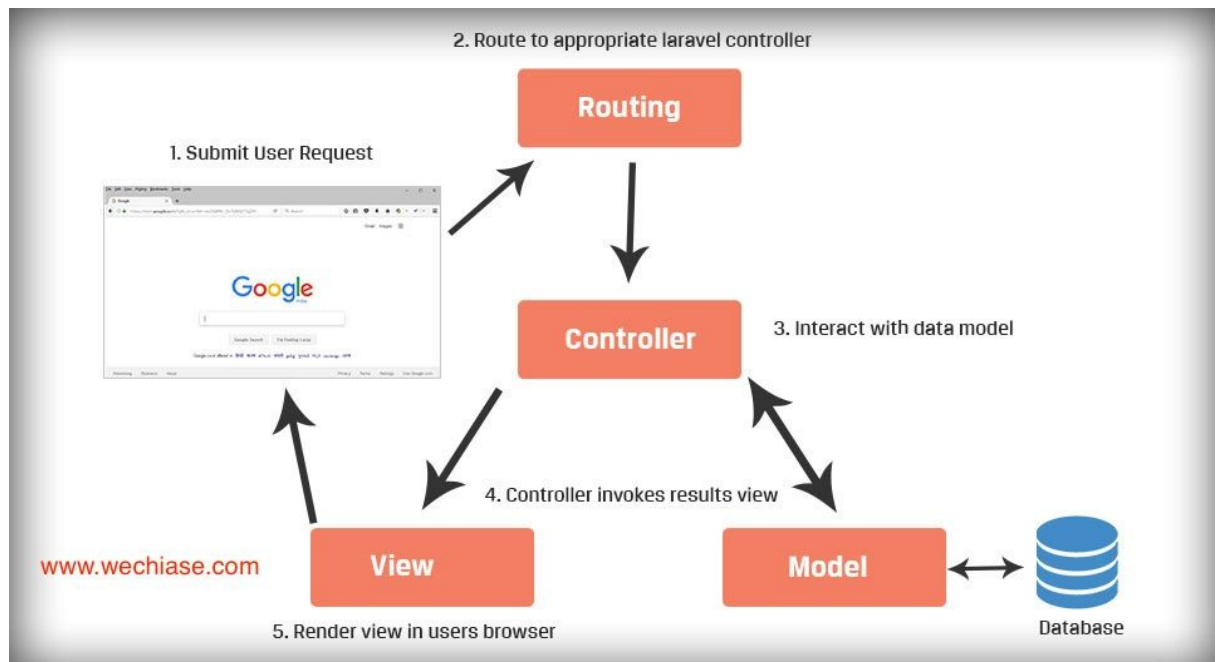
- + Khi khách hàng muốn thay đổi chức năng xóa thì dữ liệu xóa đi nhưng vẫn lưu lại lịch sử xóa thì chỉ cần tạo thêm module "lưu trữ lịch sử xóa" nằm trong module "Xóa sản phẩm".

- + Khi khách hàng muốn có thêm 1 trường thuộc tính mới của sản phẩm thì chỉ cần sửa trong module "Thêm sản phẩm" và thêm trường đó vào cơ sở dữ liệu...

-----> Như vậy khi chia website bán hàng thành các module theo chức năng thì có thể dễ dàng ứng phó với sự thay đổi.

Point out a software architecture that is convenient to build the online selling software. Explain.

- Kiến trúc để xây dựng phần mềm bán hàng trực tuyến là: MVC architecture



- Kiến trúc MVC được chia làm 3 lớp xử lý gồm Model – View – Controller :

- Model : là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql...); nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu...
- View : là nơi chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh... nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.
- Controller : là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.

- Ưu điểm của kiến trúc MVC:

- Trình tự xử lý rất rõ ràng
- Mô hình MVC quy hoạch các class/function vào các thành phần riêng biệt Controller - Model - View, việc đó làm cho quá trình phát triển - quản lý - vận hành - bảo trì web diễn ra thuận lợi hơn, tạo ra được các chức năng chuyên biệt hoá đồng thời kiểm soát được luồng xử lý.

- Tạo mô hình chuẩn cho dự án, khi người có chuyên môn ngoài dự án tiếp cận với dự án dễ dàng hơn.
- Mô hình đơn giản, dễ hiểu, xử lý những nghiệp vụ đơn giản, và dễ dàng triển khai với các dự án nhỏ.
- Phân vùng nhiệm vụ giúp Lập trình viên chuyên sâu trong việc phát triển và nâng cấp trong tương lai.

3. AS Group Assignment 4

1. Kiểm tra cấu trúc bên ngoài

- Module quản lý sản phẩm có thể đóng gói riêng được, triển khai độc lập và có sản phẩm demo riêng
- Có giao diện liên quan đến phần quản lý của admin
- Có API cung cấp và API nội bộ để vận hành module

2. Kiểm tra cấu trúc bên trong

Luồng hoạt động của task card thể hiện đúng với User story

Quản lý sản phẩm là một module có thể nói là không chia nhỏ hơn thành các module con được, chính vì vậy việc chia công việc các vòng ở bên dưới chỉ làm cho module này ngày càng hoàn thiện hơn.

Phân chia công việc các vòng:

- Vòng 1: Quản lý sản phẩm sử dụng category_id và brand_id.
- Vòng 2: Quản lý category
- Vòng 3: Quản lý brand

Vòng 1: Quản lý sản phẩm sử dụng category_id và brand_id.

Luồng hoạt động	Sau khi người dùng đăng nhập với tư cách người quản trị viên , màn hình sẽ chuyển đến trang quản lý sản phẩm. Tại đây quản trị viên có thể xem danh sách các sản phẩm có trong cơ sở dữ liệu, xem thông tin chi tiết của 1 sản phẩm, thêm,sửa, xóa
-----------------	--

	sản phẩm.
User Story	Là Admin, có thể quản lí sản phẩm (thêm, sửa, xóa,...)
Task card	Test card
<p>1. Xem danh sách sản phẩm</p> <p>- Sau khi đăng nhập, người dùng sẽ truy cập vào trang quản lý sản phẩm với tư cách là Admin. Tại đây người dùng sẽ xem được tất cả các sản phẩm sắp xếp theo thứ tự tạo.</p> <p>- Người dùng có thể thấy các action button New Product ,Edit, Delete tương ứng với từng sản phẩm.</p>	<p>Input: product id</p> <p>Test:</p> <p>- Truy vấn vào cơ sở dữ liệu, lấy ra product có id đó</p> <p>- Hiển thị tất cả thông tin lên trang product detail</p> <p>Output:</p> <p>Hiển thị nếu id tồn tại hoặc hiển thị thông báo lỗi, không tìm được id</p>
<p>2. Xem chi tiết sản phẩm</p> <p>- Người dùng có thể xem chi tiết sản phẩm bằng cách nhấn vào tên hoặc id của sản phẩm. Hệ thống sẽ chuyển hướng người dùng đến trang product detail. Bao gồm tất cả thông tin chi tiết của sản phẩm đó.</p>	<p>Input: product id</p> <p>Test:</p> <p>- Truy vấn vào cơ sở dữ liệu, lấy ra product có id đó</p> <p>- Hiển thị tất cả thông tin lên trang product detail</p> <p>Output:</p> <p>Hiển thị nếu id tồn tại hoặc hiển thị thông báo lỗi, không tìm được id</p>
<p>3. Xóa sản phẩm</p> <p>- Người dùng có thể xóa sản phẩm bằng cách nhấn vào button DELETE trên hàng sản phẩm muốn xóa. Sau khi xóa thành công, người dùng sẽ được điều hướng quay lại trang hiển thị danh sách sản phẩm.</p>	<p>Người dùng thực hiện thao tác xóa sản phẩm ra khỏi danh sách sản phẩm có trong kho hàng.</p> <p>Input: product id (Nhấn vào button DELETE).</p> <p>Test:</p> <p>- Người dùng ấn vào button DELETE, trang web hiển thị dialog thông báo người dùng chắc chắn muốn xóa không.</p> <p>- Khi người dùng ấn vào nút "Ok", Truy</p>

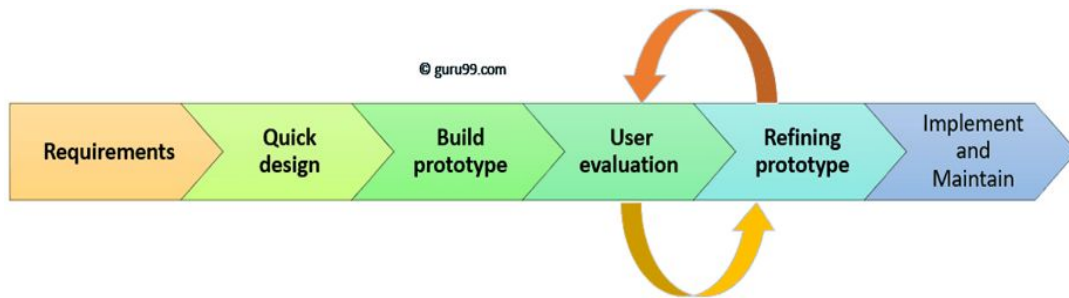
	<p>vẫn dữ liệu, xóa đi product có id đó</p> <p>Output:</p> <p>Thông báo xác nhận: "Bạn có chắc chắn muốn xóa sản phẩm này không ?"+ Người dùng xác nhận xóa -> Dữ liệu về sản phẩm vừa xóa sẽ được xóa khỏi cơ sở dữ liệu. -> Redirect người dùng tới trang Hiển thị danh sách sản phẩm.</p> <p>+ Người dùng xác nhận không xóa -> Dữ liệu về sản phẩm vẫn còn trên cơ sở dữ liệu.</p>
<p>4. Thêm sản phẩm</p> <p>- Người dùng có thể thêm thông tin sản phẩm bằng cách nhấn vào button NEW PRODUCT. Hệ thống sẽ tự động hiện 1 trang tạo sản phẩm mới cho phép người dùng nhập thông tin sản phẩm muốn tạo. Hệ thống sẽ gửi thông báo lỗi nếu có lỗi xảy ra trong quá trình cập nhật thông tin sản phẩm.</p>	<p>- Sau khi hệ thống đã chuyển sang trang tạo sản phẩm mới, người dùng tiến hành nhập thông tin sản phẩm cho sản phẩm mình muốn thêm vào kho hàng. Nhập thông tin sản phẩm -> thông tin hợp lệ -> hệ thống tiến hành thêm vào cơ sở dữ liệu.</p> <p>- Người dùng nhập thiếu thông tin cho các trường thông tin của sản phẩm -> thông báo lỗi nhập thiếu thông tin -> hệ thống yêu cầu người dùng nhập đầy đủ thông tin.</p> <p>- Người dùng nhập sai cú pháp, giá trị không hợp lệ(Ví dụ: nhập giá tiền nhỏ hơn 0, số lượng nhỏ hơn 0 hoặc giá tiền nhưng lại nhập bằng một string ...) -> yêu cầu người dùng nhập lại thông tin chính xác.</p> <p>Input: new product</p> <p>Test:</p> <p>- Kiểm tra các trường có đầy đủ thông tin không</p> <p>- Kiểm tra các trường đã đúng thông tin chưa</p>

	<p>- Lưu vào cơ sở dữ liệu</p> <p>Output:</p> <p>Ok và redirect người dùng tới trang product detail, hoặc thông báo lỗi tương ứng</p>
<p>6. Người dùng có thể sửa thông tin sản phẩm bằng cách nhấn vào button EDIT bên cột action. Hệ thống sẽ tự động chuyển đến trang /:id/edit của sản phẩm đó. Tại đó người dùng có thể sửa đổi thông tin sản phẩm. Hệ thống sẽ gửi thông báo lỗi nếu có lỗi xảy ra trong quá trình cập nhật thông tin sản phẩm.</p>	<p>Trong trường hợp người dùng cần thay đổi thông tin của các sản phẩm đã có trong kho hàng như: name, price, category...</p> <p>- Người dùng nhập sai thông tin của sản phẩm (giá cả, số lượng ..)-> hệ thống yêu cầu người dùng nhập cho đúng.</p> <p>- Người dùng nhập sai cú pháp (ví dụ trong giá của sản phẩm có chứa ký tự đặc biệt ..)-> yêu cầu người dùng nhập lại.</p> <p>- Trong quá trình sửa lại thông tin của sản phẩm, người dùng vô tình để trống một trường thông tin nào đó và ấn hoàn tất -> báo lỗi cho người dùng -> yêu cầu nhập đầy đủ tất cả thông tin.</p> <p>Input: product</p> <p>Test:</p> <p>- Kiểm tra các trường có đầy đủ thông tin không</p> <p>- Kiểm tra các trường đã đúng thông tin chưa</p> <p>- Update vào cơ sở dữ liệu</p> <p>Output:</p> <p>Ok và redirect người dùng tới trang product detail, hoặc thông báo lỗi tương ứng</p>

4. Once finished, show the way to assemble this version with the next versions (if any).

Lắp ráp theo module, mỗi module chứa trong 1 folder. nếu cần sử dụng sẽ require file cần thiết vào.

5. Identify a suitable approach to build the prototype of the next version.



Step 1: Requirements gathering and analysis

Step 2: Quick design

Step 3: Build a Prototype

Step 4: Initial user evaluation

Step 5: Refining prototype

Step 6: Implement Product and Maintain

Các loại prototype:

- Rapid Throwaway prototypes

- Evolutionary prototype

- Incremental prototype

- Extreme prototype

6. Is your program well structured? Does the way to assemble different versions of your program make good your program? Explain.

Chương trình có cấu trúc tốt và lắp ráp tốt vì phân chia theo các module chức năng, sử dụng mô hình mvc, khi lắp ráp chỉ cần thêm file và nếu file nào yêu cầu đến thì chỉ cần require là sử dụng được.

4. AS Group Assignment 5

Vòng 2: Quản lý category.

Luồng hoạt động	Sau khi người dùng đăng nhập với tư cách người quản trị viên , màn hình sẽ chuyển đến trang quản lý sản phẩm. Tại đây quản trị viên nhấn vào button category để điều hướng sang phần quản lý category, sau đó quản trị viên có thể xem danh sách các category có trong cơ sở dữ liệu, xem thông tin chi tiết của 1 category, thêm,sửa, xóa category.
User Story	Là Admin, có thể quản lý category (thêm, sửa, xóa,...)
Task card	Test card
1. Xem danh sách category - Sau khi đăng nhập, người dùng sẽ truy cập vào trang quản lý sản phẩm với tư cách là Admin. Tại đây người dùng sẽ xem được tất cả các category sắp xếp theo thứ tự tạo. - Người dùng có thể thấy các action button New Product ,Edit, Delete tương ứng với từng category.	Input: category id Test: - Truy vấn vào cơ sở dữ liệu, lấy ra category có id đó - Hiện thị tất cả thông tin lên trang category detail Output: Hiện thị nếu id tồn tại hoặc hiện thị thông báo lỗi, không tìm được id
2. Xem chi tiết category - Người dùng có thể xem chi tiết category bằng cách nhấn vào tên hoặc id của category	Input: category id Test: - Truy vấn vào cơ sở dữ liệu, lấy ra category có id đó

<p>đó. Hệ thống sẽ chuyển hướng người dùng đến trang category detail. Bao gồm tất cả thông tin chi tiết của category đó.</p>	<p>- Hiện thị tất cả thông tin lên trang category detail</p> <p>Output:</p> <p>Hiện thị nếu id tồn tại hoặc hiện thị thông báo lỗi, không tìm được id</p>
<p>3. Xóa category</p> <p>- Người dùng có thể category bằng cách nhấn vào button DELETE trên hàng category muốn xóa. Sau khi xóa thành công, người dùng sẽ được điều hướng quay lại trang hiển thị danh sách category.</p>	<p>Người dùng thực hiện thao tác xóa category ra khỏi danh sách category có trong kho hàng.</p> <p>Input: category id (Nhấn vào button DELETE).</p> <p>Test:</p> <ul style="list-style-type: none"> - Người dùng ấn vào button DELETE, trang web hiển thị dialog thông báo người dùng chắc chắn muốn xóa không. - Khi người dùng ấn vào nút "Ok", Truy vấn dữ liệu, xóa đi category có id đó <p>Output:</p> <p>Thông báo xác nhận: "Bạn có chắc chắn muốn xóa category này không ?"+ Người dùng xác nhận xóa -> Dữ liệu về sản phẩm vừa xóa sẽ được xóa khỏi cơ sở dữ liệu. -> Redirect người dùng tới trang Hiển thị danh sách category.</p> <p>+ Người dùng xác nhận không xóa -> Dữ liệu về category vẫn còn trên cơ sở dữ liệu.</p>

<p>4. Thêm category</p> <p>- Người dùng có thể thêm thông tin category bằng cách nhấn vào button NEW CATEGORY. Hệ thống sẽ tự động hiện 1 trang tạo category mới cho phép người dùng nhập thông tin category muốn tạo. Hệ thống sẽ gửi thông báo lỗi nếu có lỗi xảy ra trong quá trình cập nhật thông tin category.</p>	<ul style="list-style-type: none"> - Sau khi hệ thống đã chuyển sang trang tạo category mới, người dùng tiến hành nhập thông tin category cho category mình muốn thêm vào kho hàng. Nhập thông tin category -> thông tin hợp lệ -> hệ thống tiến hành thêm vào cơ sở dữ liệu. - Người dùng nhập thiếu thông tin cho các trường thông tin của category -> thông báo lỗi nhập thiếu thông tin -> hệ thống yêu cầu người dùng nhập đầy đủ thông tin. - Người dùng nhập sai cú pháp, giá trị không hợp lệ(Ví dụ: id nhưng lại nhập bằng một string ...) -> yêu cầu người dùng nhập lại thông tin chính xác. <p>Input: new category</p> <p>Test:</p> <ul style="list-style-type: none"> - Kiểm tra các trường có đầy đủ thông tin không - Kiểm tra các trường đã đúng thông tin chưa - Lưu vào cơ sở dữ liệu <p>Output:</p> <p>Ok và redirect người dùng tới trang category detail, hoặc thông báo lỗi tương ứng</p>
---	---

	<p>Trong trường hợp người dùng cần thay đổi thông tin của các sản phẩm đã có trong kho hàng như: description, name...</p> <ul style="list-style-type: none"> - Người dùng nhập sai thông tin của category (description, name ..)-> hệ thống yêu cầu người dùng nhập cho đúng. - Người dùng nhập sai cú pháp (ví dụ trong id có chứa ký tự đặc biệt ..)-> yêu cầu người dùng nhập lại. - Trong quá trình sửa lại thông tin của category, người dùng vô tình để trống một trường thông tin nào đó và ấn hoàn tất -> báo lỗi cho người dùng -> yêu cầu nhập đầy đủ tất cả thông tin.
<p>5.Sửa category</p> <p>Người dùng có thể sửa thông tin category bằng cách nhấn vào button EDIT bên cột action. Hệ thống sẽ tự động chuyển đến trang /:id/edit của category đó. Tại đó người dùng có thể sửa đổi thông tin category. Hệ thống sẽ gửi thông báo lỗi nếu có lỗi xảy ra trong quá trình cập nhật thông tin category.</p>	<p>Input: category</p> <p>Test:</p> <ul style="list-style-type: none"> - Kiểm tra các trường có đầy đủ thông tin không - Kiểm tra các trường đã đúng thông tin chưa - Update vào cơ sở dữ liệu <p>Output:</p> <p>Ok và redirect người dùng tới trang category detail, hoặc thông báo lỗi tương ứng</p>

Danh sách các API cung cấp của vòng 2

GET	/api/categories	Thông tin tất cả các category
GET	/api/products/:id	Thông

		tin 1 category
POST	/api/products	Tạo category mới
PUT	/api/products/:id	Cập nhật category
DELETE	/api/products/:id	Xóa category

* Phần code thực hiện theo mô hình MVC

II. Justify the application of SOLID principles to your design.

A. Nguyên lý SOLID

SOLID là nguyên lý cơ bản trong thiết kế phần mềm hướng đối tượng, giúp code trở nên dễ hiểu, linh động và dễ bảo trì hơn. Năm nguyên lý trong SOLID bao gồm:

- **S:** Single responsibility principle
- **O:** Open/closed principle
- **L:** Liskov substitution principle
- **I:** Interface segregation principle
- **D:** Dependency inversion principle

1. Single responsibility principle

Nguyên lý này có thể hiểu là một class chỉ nên đảm nhận một và chỉ một trách nhiệm duy nhất. Hay nói theo một cách khác, chúng ta chỉ nên viết và sửa đổi một class bởi vì một lý do duy nhất mà thôi. Nguyên lý giúp code dễ quản lý và bảo trì hơn.

Trong phát triển sản phẩm, nhóm đã chia module theo mô hình MVC, trong đó mỗi class chỉ đảm nhiệm 1 nhiệm vụ duy nhất. ví dụ: view chỉ dùng để hiển thị, model liên kết với dữ liệu,....

2. Open/closed principle

Nguyên lý này có thể hiểu là nếu như có thêm yêu cầu phát sinh, chúng ta nên mở rộng/kế thừa các class đã tồn tại thay vì sửa đổi chúng.

3. Liskov substitution principle

Nguyên lý này có thể hiểu là các đối tượng của class cha có thể được thay thế bởi các đối tượng của các class con mà không làm thay đổi tính đúng đắn của chương trình.

4. Interface segregation principle

Nguyên lý này có thể hiểu là thay vì viết một interface cho một mục đích chung chung, chúng ta nên tách thành nhiều interface nhỏ cho các mục đích riêng. Chúng ta không nên bắt buộc client phải implement các method mà client không cần đến.

5. Dependency inversion principle

Các module cấp cao không nên phụ thuộc vào các module cấp thấp. Cả hai nên phụ thuộc vào abstraction.

* Thực hiện nguyên lý SOLID

- Nguyên lý 1: Một class chỉ nên đảm nhận và chỉ 1 trách nhiệm duy nhất

Trong phần code thực hiện có rất nhiều class khác nhau: create product, edit product ... mỗi class này chỉ đảm nhận 1 nhiệm vụ

- Nguyên lý 2: Có thể thoải mái mở rộng một class nhưng không được sửa đổi bên trong class đó.

Để thực hiện theo nguyên lý SOLID là rất khó, nhưng kết quả mang lại là rất nhiều, nhóm đang cố gắng refactor lại code 1 cách hợp lý nhất.

B. Phương pháp agile

Agile là tập hợp các nguyên lý tạo ra phương thức phát triển phần mềm linh hoạt với mục tiêu đưa sản phẩm đến tay người dùng càng nhanh càng tốt.

Mô hình Agile trong phát triển phần mềm là mô hình gia tăng, hoạt động trên quy trình phát triển lặp với những điểm cơ bản như:

- Phân chia dự án thành mảng nhỏ để dễ xử lý và thay đổi khi khách hàng yêu cầu thay đổi.
- Trong quá trình làm dự án thì từng phần nhỏ sẽ được test ngay.
- Nhóm dự án Agile yêu cầu cùng có mặt để trao đổi thường xuyên.

Về cơ bản, quy trình Agile được phân chia thành các giai đoạn chính như sau:

- Lập kế hoạch
- Phân tích
- Thiết kế và lập trình
- Kiểm thử sản phẩm
- Bàn giao sản phẩm

Trong quá trình phát triển nhóm luôn tuân theo phương pháp agile:

Xác định yêu cầu, các bên liên quan -> Muồng tượng những câu chuyện người dùng (story) sẽ xảy ra -> Đưa ra mức độ ưu tiên cho các câu chuyện người dùng (story), Ước tính nỗ lực cho các câu chuyện người dùng (story) -> Chia nhỏ công việc, Chuẩn bị kịch bản kiểm thử cho mỗi công việc -> Lập trình, Kiểm thử -> Phát hành sản phẩm quy mô nhỏ -> Xác định yêu cầu, các bên liên quan.

Vòng 3: Quản lý brand.

Luồng hoạt động	Sau khi người dùng đăng nhập với tư cách người quản trị viên , màn hình sẽ chuyển đến trang quản lý sản phẩm. Tại đây quản trị viên nhấn vào button brand để điều hướng sang phần quản lý brand, sau đó quản trị viên có thể xem danh sách các brand có trong cơ sở dữ liệu, xem thông tin chi tiết của 1 brand, thêm,sửa, xóa brand.
User Story	Là Admin, có thể quản lý brand(thêm, sửa, xóa,...)
Task card	Test card
1. Xem danh sách brand - Sau khi đăng nhập, người dùng sẽ truy cập vào trang quản lý sản phẩm với tư cách là Admin.Tại đây người dùng sẽ xem được tất cả các brand sắp	Input: brand id Test: - Truy vấn vào cơ sở dữ liệu, lấy ra brand có id đó - Hiển thị tất cả thông tin lên trang brand detail

<p>xếp theo thứ tự tạo.</p> <ul style="list-style-type: none"> - Người dùng có thể thấy các action button New Brand ,Edit, Delete tương ứng với từng brand. 	<p>Output:</p> <p>Hiển thị nếu id tồn tại hoặc hiển thị thông báo lỗi, không tìm được id</p>
<p>2. Xem chi tiết brand</p> <ul style="list-style-type: none"> - Người dùng có thể xem chi tiết brand bằng cách nhấn vào tên hoặc id của brand đó. Hệ thống sẽ chuyển hướng người dùng đến trang brand detail. Bao gồm tất cả thông tin chi tiết của brand đó. 	<p>Input: brand id</p> <p>Test:</p> <ul style="list-style-type: none"> - Truy vấn vào cơ sở dữ liệu, lấy ra brand có id đó - Hiển thị tất cả thông tin lên trang brand detail <p>Output:</p> <p>Hiển thị nếu id tồn tại hoặc hiển thị thông báo lỗi, không tìm được id</p>
<p>3. Xóa brand</p> <ul style="list-style-type: none"> - Người dùng có thể brand bằng cách nhấn vào button DELETE trên hàng brand muốn xóa. Sau khi xóa thành công, người dùng sẽ được điều hướng quay lại trang hiển thị danh sách category. 	<p>Người dùng thực hiện thao tác xóa brandra khỏi danh sách brand có trong kho hàng.</p> <p>Input: brand id (Nhấn vào button DELETE).</p> <p>Test:</p> <ul style="list-style-type: none"> - Người dùng ấn vào button DELETE, trang web hiển thị dialog thông báo người dùng chắc chắn muốn xóa không. - Khi người dùng ấn vào nút "Ok", Truy vấn dữ liệu, xóa đi brand có id đó <p>Output:</p> <p>Thông báo xác nhận: "Bạn có chắc chắn muốn xóa brand này không ?"+ Người dùng xác nhận xóa -> Dữ liệu về sản phẩm vừa xóa sẽ được xóa khỏi cơ sở dữ liệu. -> Redirect người dùng tới trang Hiển thị danh sách brand.</p>

	<p>+ Người dùng xác nhận không xóa -> Dữ liệu về brand vẫn còn trên cơ sở dữ liệu.</p>
<p>4. Thêm brand</p> <p>- Người dùng có thể thêm thông tin brand bằng cách nhấn vào button NEW BRAND. Hệ thống sẽ tự động hiện 1 trang tạo brand mới cho phép người dùng nhập thông tin brand muốn tạo. Hệ thống sẽ gửi thông báo lỗi nếu có lỗi xảy ra trong quá trình cập nhật thông tin brand.</p>	<p>- Sau khi hệ thống đã chuyển sang trang tạo brand mới, người dùng tiến hành nhập thông tin brand cho brand mình muốn thêm vào kho hàng. Nhập thông tin brand-> thông tin hợp lệ -> hệ thống tiến hành thêm vào cơ sở dữ liệu.</p> <p>- Người dùng nhập thiếu thông tin cho các trường thông tin của brand-> thông báo lỗi nhập thiếu thông tin -> hệ thống yêu cầu người dùng nhập đầy đủ thông tin.</p> <p>- Người dùng nhập sai cú pháp, giá trị không hợp lệ(Ví dụ: id nhưng lại nhập bằng một string ...) -> yêu cầu người dùng nhập lại thông tin chính xác.</p> <p>Input: new brand</p> <p>Test:</p> <p>- Kiểm tra các trường có đầy đủ thông tin không</p> <p>- Kiểm tra các trường đã đúng thông tin chưa</p> <p>- Lưu vào cơ sở dữ liệu</p> <p>Output:</p> <p>Ok và redirect người dùng tới trang brand detail, hoặc thông báo lỗi tương ứng</p>

<p>5.Sửa brand</p> <p>Người dùng có thể sửa thông tin brand bằng cách nhấn vào button EDIT bên cột action. Hệ thống sẽ tự động chuyển đến trang /:id/edit của brand đó. Tại đó người dùng có thể sửa đổi thông tin brand. Hệ thống sẽ gửi thông báo lỗi nếu có lỗi xảy ra trong quá trình cập nhật thông tin brand.</p>	<p>Trong trường hợp người dùng cần thay đổi thông tin của các sản phẩm đã có trong kho hàng như: description, name...</p> <ul style="list-style-type: none"> - Người dùng nhập sai thông tin của brand(description, name ..)-> hệ thống yêu cầu người dùng nhập cho đúng. - Người dùng nhập sai cú pháp (ví dụ trong id có chứa ký tự đặc biệt ..)-> yêu cầu người dùng nhập lại. - Trong quá trình sửa lại thông tin của category, người dùng vô tình để trống một trường thông tin nào đó và ấn hoàn tất -> báo lỗi cho người dùng -> yêu cầu nhập đầy đủ tất cả thông tin. <p>Input: category</p> <p>Test:</p> <ul style="list-style-type: none"> - Kiểm tra các trường có đầy đủ thông tin không - Kiểm tra các trường đã đúng thông tin chưa - Update vào cơ sở dữ liệu <p>Output:</p> <p>Ok và redirect người dùng tới trang brand detail, hoặc thông báo lỗi tương ứng</p>
---	--

Danh sách các API cung cấp của vòng 3

GET	/api/brands	Thông tin tất cả các brand
GET	/api/brands/:id	Thông tin 1 brand
POST	/api/brands	Tạo brand mới

PUT	/api/brands/:id	Cập nhật brand
DELETE	/api/brands/:id	Xóa brand

5. AS Group Assignment 6

I. Identify website's functionalities that can be developed as a web service.

Xác định các chức năng của trang web có thể được phát triển như một dịch vụ web.

- Web service (dịch vụ web) là tập hợp các giao thức và tiêu chuẩn mở được sử dụng để trao đổi dữ liệu giữa các ứng dụng hoặc giữa các hệ thống.
- Có rất nhiều chức năng của trang web có thể được phát triển như là 1 web service. Ví dụ: quản lý sản phẩm, quản lý khách hàng, thanh toán, quản lý quảng cáo, chăm sóc khách hàng, quản lý kho hàng,....
- Hiện tại nhóm đang phát triển chức năng quản lý sản phẩm. Quản lý sản phẩm có thể phát triển thành web service, và cung cấp cho client hay các nhóm khác một số các chức năng như: thêm, sửa, xóa, hiển thị thông tin products, thêm, sửa, xóa, hiển thị category, thêm, sửa, xóa, hiển thị brand.

1. Develop such a web service.

Nhóm đã deploy website lên heroku. Client có thể truy cập.

Link: <https://nguyenvd27-ltct-demo.herokuapp.com/products/>

Demo 1 số chức năng:

1. Lấy thông tin tất cả sản phẩm.

(<https://nguyenvd27-ltct-demo.herokuapp.com/api/products/>)

```

"result": "ok",
"data": [
  {
    "id": 8,
    "name": "Áo khoác bomber kaki màu bò 999",
    "price": 999000,
    "description": "chất lượng siêu đỉnh",
    "image": "https://firebasestorage.googleapis.com/v0/b/image-1c367.appspot.com/o/ao-khoac-bomber-kaki-nau-bo-ak193-4918-p-s-t.jpg?alt=media&token=aeee7973-bdb7-41a2-97d6-6b7ac276f68d",
    "category": "Áo khoác",
    "brand": "ADIDAS",
    "sold_out": false,
    "created_at": "2019-11-19T11:40:11.512Z",
    "updated_at": "2019-11-19T11:40:41.806Z"
  },
  {
    "id": 7,
    "name": "Áo Khoác Da Nâu AK209",
    "price": 390000,
    "description": "demo 12",
    "image": "https://firebasestorage.googleapis.com/v0/b/image-1c367.appspot.com/o/ao-khoac-da-nau-ak209-6252-p-s-t.jpg?alt=media&token=b08b97e2-3509-4cd7-bb07-c443671e46f6",
    "category": "Áo khoác",
    "brand": "nguyenvd",
    "sold_out": false,
    "created_at": "2019-11-19T11:39:05.019Z",
    "updated_at": "2019-11-19T11:39:05.019Z"
  },
  {
    "id": 6,
    "name": "Áo khoác nữ xanh đen",
    "price": 390000,
    "description": "hàng Việt Nam chất lượng trung quốc",
    "image": "https://firebasestorage.googleapis.com/v0/b/image-1c367.appspot.com/o/ao-khoac-ni-xanh-den-ak176-2187-p.jpg?alt=media&token=d84981d4-5af8-4850-a66b-11fcb88ccff",
    "category": "Áo khoác",
    "brand": "ADIDAS",
    "sold_out": true,
    "created_at": "2019-11-19T11:38:09.777Z",
    "updated_at": "2019-11-19T11:38:09.777Z"
  },
  {
    "id": 5,
    "name": "Áo thun Cà Sầu Vàng",
    "price": 300000,
    "description": "hàng Việt Nam chất lượng trung quốc",
    "image": "https://firebasestorage.googleapis.com/v0/b/image-1c367.appspot.com/o/ao-thun-ca-sau-vang-at558-1482-p.jpg?alt=media&token=37d14631-f22a-4e0f-ba16-3a01c9f76359",
    "category": "Áo thun",
    "brand": "nguyenvd",
    "sold_out": true,
    "created_at": "2019-11-19T11:37:30.714Z",
    "updated_at": "2019-11-19T11:37:30.714Z"
  },
  {
    "id": 4,
    "name": "Áo thun Cổ Tròn Cam",
    "price": 300000,
    "description": "hàng Việt Nam chất lượng trung quốc",
    "image": "https://firebasestorage.googleapis.com/v0/b/image-1c367.appspot.com/o/ao-thun-co-tron-can-at596-3379-p.jpg?alt=media&token=a2103f8b-d661-469e-8032-9233bec5bdf4",
    "category": "Áo thun",
    "brand": "nguyenvd",
    "sold_out": false,
    "created_at": "2019-11-19T11:36:36.454Z",
    "updated_at": "2019-11-19T11:36:36.454Z"
  }
]

```

2. Lấy thông tin chi tiết 1 sản phẩm

(<https://nguyenvd27-ltct-demo.herokuapp.com/api/products/8>)

```

1 {
2   "result": "ok",
3   "data": [
4     {
5       "id": 8,
6       "name": "Áo khoác bomber kaki màu bò 999",
7       "price": 999000,
8       "description": "chất lượng siêu đỉnh",
9       "image": "https://firebasestorage.googleapis.com/v0/b/image-1c367.appspot.com/o/ao-khoac-bomber-kaki-nau-bo-ak193-4918-p-s-t.jpg?alt=media&token=aeee7973-bdb7-41a2-97d6-6b7ac276f68d",
10      "category": "Áo khoác",
11      "brand": "ADIDAS",
12      "sold_out": false,
13      "created_at": "2019-11-19T11:40:11.512Z",
14      "updated_at": "2019-11-19T11:40:41.806Z"
15    }
16  ],
17   "message": "query product by id successfully"
18 }

```

3. Lấy thông tin tất cả categories

(<https://nguyenvd27-ltct-demo.herokuapp.com/api/categories/>)

```

1 {
2   "result": "ok",
3   "data": [
4     {
5       "id": 6,
6       "name": "Váy nữ",
7       "description": "lalal",
8       "created_at": "2019-11-19T11:31:00.383Z",
9       "updated_at": "2019-11-19T11:31:00.383Z"
10    },
11    {
12      "id": 5,
13      "name": "Váy nam",
14      "description": "demo",
15      "created_at": "2019-11-19T11:30:44.804Z",
16      "updated_at": "2019-11-19T11:30:44.804Z"
17    },
18    {
19      "id": 4,
20      "name": "Quần jean",
21      "description": "demo loto",
22      "created_at": "2019-11-19T11:30:27.278Z",
23      "updated_at": "2019-11-19T11:30:27.278Z"
24    },
25    {
26      "id": 3,
27      "name": "Áo khoác",
28      "description": "demo",
29      "created_at": "2019-11-19T11:30:10.641Z",
30      "updated_at": "2019-11-19T11:30:10.641Z"
31    },
32    {
33      "id": 2,
34      "name": "Áo sơ mi",
35      "description": "demo 12",
36      "created_at": "2019-11-19T11:29:55.618Z",
37      "updated_at": "2019-11-19T11:29:55.618Z"
38    },
39    {
40      "id": 1,
41      "name": "Áo thun",
42      "description": "demo category",
43      "created_at": "2019-11-19T11:29:37.987Z",
44      "updated_at": "2019-11-19T11:31:15.522Z"
45    }
46  ],
47   "message": "Query list of categories successfully"
48 }

```

4. Lấy thông tin tất cả brands

(<https://nguyenvd27-ltct-demo.herokuapp.com/api/brands/>)

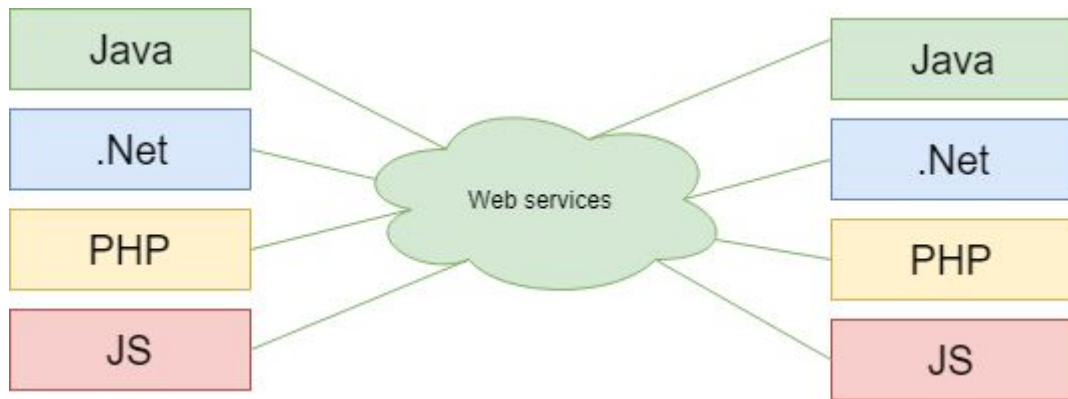
```

1 {
2   "result": "ok",
3   "data": [
4     {
5       "id": 5,
6       "name": "manhvd",
7       "description": "demo",
8       "created_at": "2019-11-19T11:32:38.092Z",
9       "updated_at": "2019-11-19T11:32:38.092Z"
10    },
11    {
12      "id": 4,
13      "name": "nguyenvd",
14      "description": "vda",
15      "created_at": "2019-11-19T11:32:24.065Z",
16      "updated_at": "2019-11-19T11:32:24.065Z"
17    },
18    {
19      "id": 3,
20      "name": "nike",
21      "description": "demo",
22      "created_at": "2019-11-19T11:32:07.398Z",
23      "updated_at": "2019-11-19T11:32:07.398Z"
24    },
25    {
26      "id": 2,
27      "name": "ADIDAS",
28      "description": "demo 12",
29      "created_at": "2019-11-19T11:31:53.318Z",
30      "updated_at": "2019-11-19T11:31:53.318Z"
31    },
32    {
33      "id": 1,
34      "name": "vu duc nguyen",
35      "description": "demo",
36      "created_at": "2019-11-19T11:31:40.925Z",
37      "updated_at": "2019-11-19T11:31:40.925Z"
38    }
39  ],
40   "message": "Query list of Brands successfully"
41 }

```

II. Use agile methods to integrate this web service to the existing version of website.

- Web service (dịch vụ web) là tập hợp các giao thức và tiêu chuẩn mở được sử dụng để trao đổi dữ liệu giữa các ứng dụng hoặc giữa các hệ thống.
- Các ứng dụng phần mềm được viết bằng các ngôn ngữ lập trình khác nhau hoặc chạy trên các nền tảng khác nhau, chúng có thể sử dụng các web service để trao đổi dữ liệu qua lại theo cách tương tự như liên lạc giữa các quá trình trên một máy tính.



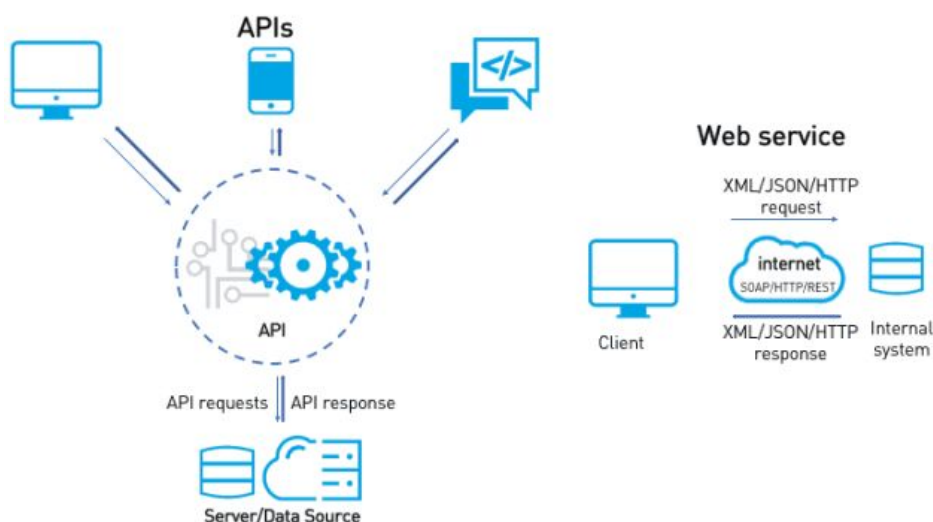
III. Web service hoạt động như thế nào?

Một ứng dụng WS bao gồm 2 thành phần: Client và Server giao tiếp với nhau qua giao thức HTTP.

+ Client gửi yêu cầu qua các lời gọi hàm thông qua HTTP Request đến Server

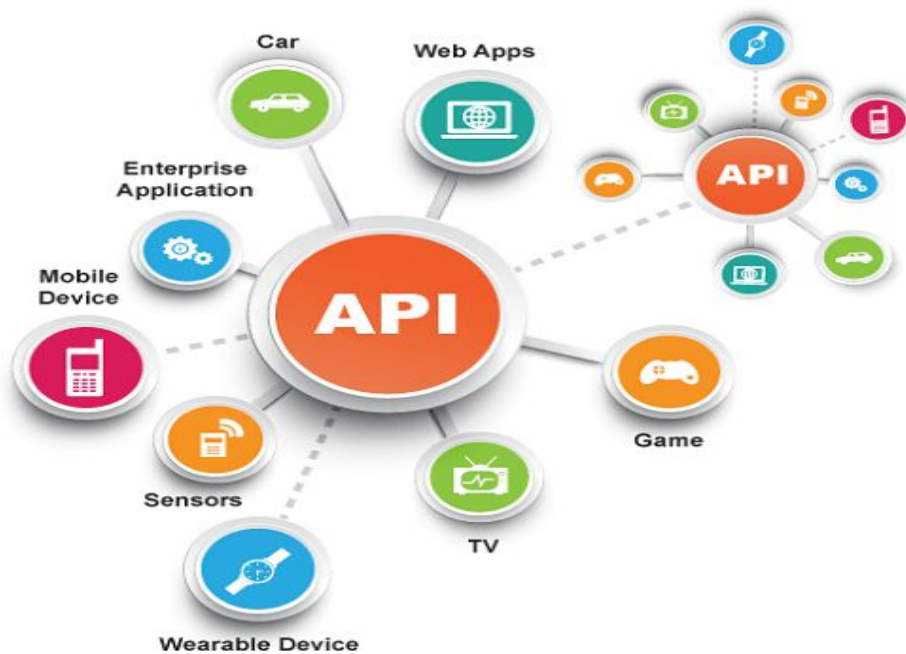
+ Server gửi các kết quả được thực thi các ở hàm thông qua HTTP Request

- REST là một kiểu kiến trúc phần mềm xác định một tập các ràng buộc sẽ được sử dụng để tạo các web services. web services phù hợp với kiến trúc REST hoặc RESTful web services, cung cấp khả năng tương tác giữa các hệ thống máy tính trên Internet. Các dịch vụ web RESTful cho phép các hệ thống yêu cầu truy cập và thao tác tài nguyên web bằng cách sử dụng một tập hợp thống nhất và được xác định trước của REST. RESTful web services rất nhẹ, có khả năng mở rộng cao và có thể bảo trì và được sử dụng rất phổ biến để tạo API cho các ứng dụng dựa trên web. REST là một phong cách kiến trúc, không phải là một giao thức.



REST tuân thủ 4 nguyên tắc thiết kế cơ bản sau:

- Sử dụng các phương thức HTTP một cách rõ ràng.
- Phi trạng thái.
- Hiển thị cấu trúc thư mục như các URLs.
- Truyền tải JavaScript Object Notation (JSON), XML hoặc cả hai.



Các hành động của một webservice thông thường bao gồm tạo dữ liệu (**Create**), lấy dữ liệu (**Read**), cập nhập dữ liệu (**Update**) hoặc xóa dữ liệu (**Delete**). Các hành động này còn được gọi là **CRUD**.

Thiết lập một ánh xạ 1-1 giữa các hành động (CRUD) và các phương thức **HTTP**:

- **POST** : để tạo một tài nguyên trên Server.
- **GET** : để truy xuất một tài nguyên.
- **PUT** : để thay đổi trạng thái một tài nguyên hoặc để cập nhật nó.
- **DELETE** : để hủy bỏ hoặc xóa một tài nguyên

.Khi Client gửi một yêu cầu tới web service nó thường được truyền tải dưới dạng XML hoặc JSON và thông thường nhận về với hình thức tương tự. Đôi khi Client cũng có thể chỉ định kiểu dữ liệu nhận về mà nó mong muốn (JSON, hoặc XML,...), các chỉ định này được gọi là các kiểu MIME, nó được gửi kèm trên phần HEADER của request.

API Quản lý sản phẩm

1	Nhóm 3: Quản lý sản phẩm			
2				
3	API cung cấp			Thời điểm cung cấp (vòng 1, vòng 2, vòng 3)
4	GET	/api/products	Thông tin tất cả các sản phẩm đang có trong database	Vòng 1 (DONE)
5	GET	/api/products/:id	Thông tin 1 sản phẩm	
6	POST	/api/products	Tạo sản phẩm mới, thêm vào database	
7	PUT	/api/products/:id	Cập nhật sản phẩm	
8	DELETE	/api/products/:id	Xóa sản phẩm	
9	GET	/api/categories	Thông tin tất cả các category	Vòng 2 (DONE)
10	GET	/api/categories/:id	Thông tin 1 category	
11	POST	/api/categories	Tạo category mới	
12	PUT	/api/categories/:id	Cập nhật category	
13	DELETE	/api/categories/:id	Xóa category	
14	GET	/api/brands	Thông tin tất cả các brand	Vòng 3 (DONE)
15	GET	/api/brands/:id	Thông tin 1 brand	
16	POST	/api/brands	Tạo brand mới	
17	PUT	/api/brands/:id	Cập nhật brand	
18	DELETE	/api/brands/:id	Xóa brand	
19				

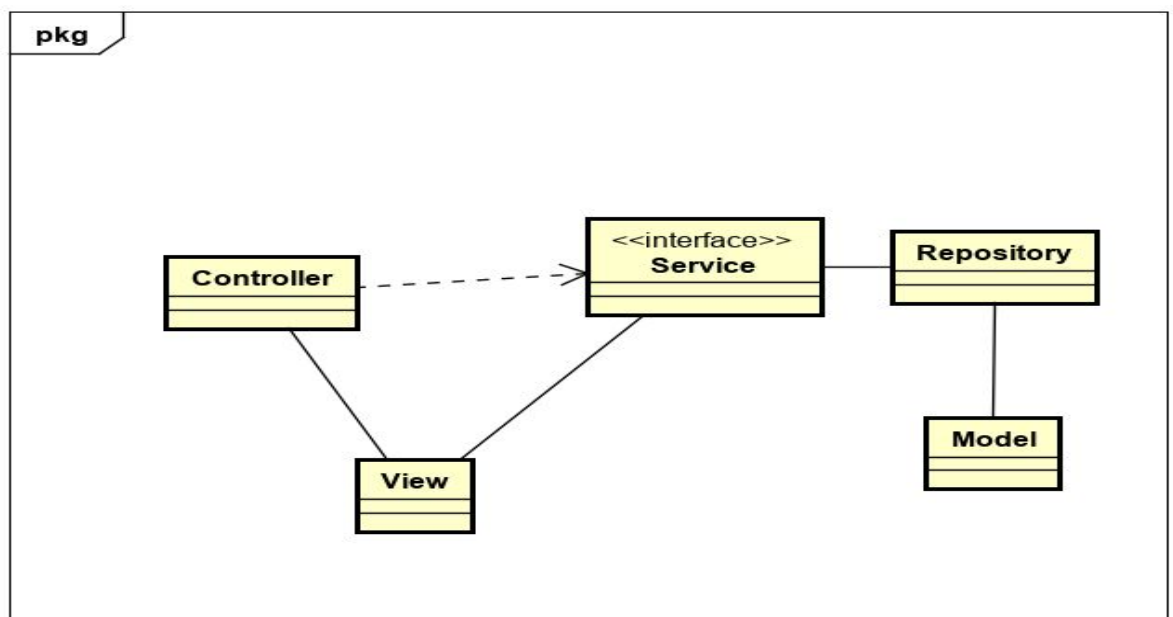
6. AS Group Assignment 7

I. Which patterns should be applied to which functionalities?

- Repository Design Pattern. Lí do:

- + Khi áp dụng design pattern vào trong code thì sẽ tách biệt giữa data access logic và business logic, cho phép người viết business logic có thể sử dụng data mà không cần phải quan tâm đến cấu trúc dữ liệu phía dưới.
- + Thực hiện điều này bằng cách thêm vào một tầng nằm giữa Business Logic và Data Source, gọi là các Repository. Các Repository này đóng vai trò như một kho chứa, nơi lưu trữ tất cả code liên quan đến việc truy cập dữ liệu, như việc mapping giữa thông tin hiển thị trên business logic với thông tin lưu trữ trong data source, hay các query truy vấn dữ liệu ...
- + Các class controller sẽ thông qua interface ở tầng service để giao tiếp với tầng model, qua đó giảm thiểu sự phụ thuộc đạt tiêu chí low coupling.

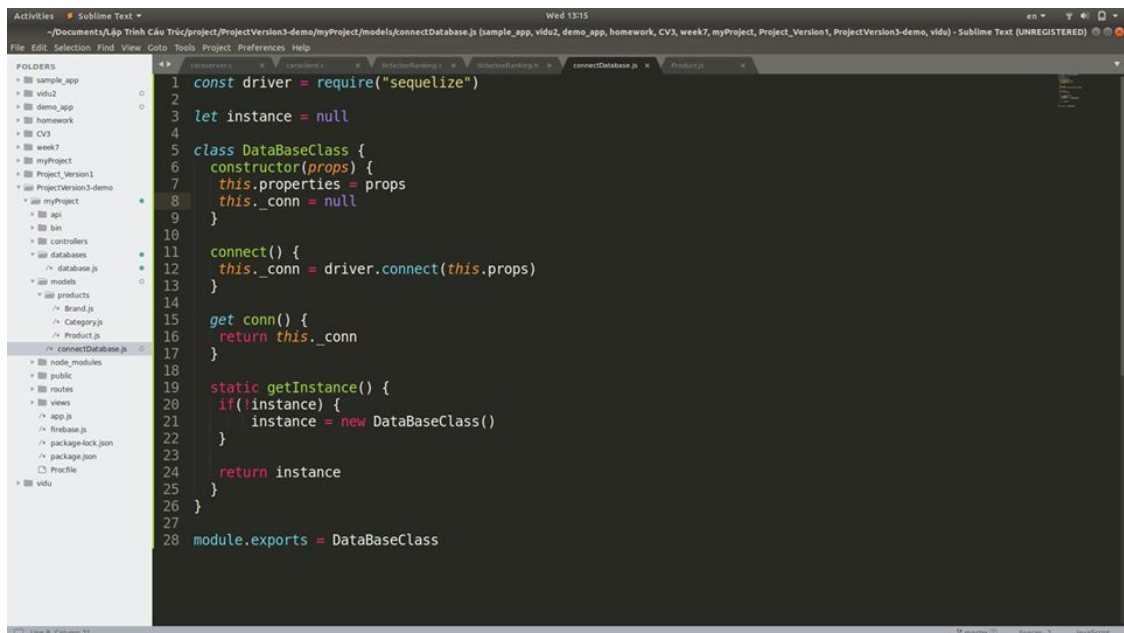
II. Illustrate by means of UML diagrams.



III. Justify by code:

- Design Pattern: Singleton pattern

Một Singleton chỉ cho phép cho một khởi tạo đơn, nhưng nhiều phiên bản của cùng một đối tượng. Singleton hạn chế clients tạo ra nhiều đối tượng, sau khi đối tượng đầu tiên được tạo ra, nó sẽ trả về các phiên bản của chính nó.



```
1 const driver = require("sequelize")
2
3 let instance = null
4
5 class DataBaseClass {
6   constructor(props) {
7     this.properties = props
8     this._conn = null
9   }
10
11   connect() {
12     this._conn = driver.connect(this.props)
13   }
14
15   get conn() {
16     return this._conn
17   }
18
19   static getInstance() {
20     if(!instance) {
21       instance = new DataBaseClass()
22     }
23
24     return instance
25   }
26 }
27
28 module.exports = DataBaseClass
```

- Design Pattern: Module Design Pattern

Là các mẫu thiết kế được sử dụng phổ biến nhất để giữ các mẫu đoạn mã độc lập với các thành phần khác. Điều này cung cấp khớp nối lỏng lẻo để hỗ trợ code có cấu trúc tốt hơn. Các module là các phần riêng biệt nên việc thêm bớt hay xóa bỏ tính năng, function cho module rất tiện.

- Lý do sử dụng: giúp code có cấu trúc tốt hơn, bảo trì và mở rộng code dễ dàng hơn.

```

1 // Models
2 const Product = require('../../models/products/Product');
3 const Category = require('../../models/products/Category');
4 const Brand = require('../../models/products/Brand');
5 const sequelize = require('../../databases/database').sequelize;
6
7 // show list of products
8 module.exports.index = async (req, res) => {
9
10 }
11
12 // page create product
13 module.exports.new = async (req, res) => {
14
15 }
16
17 // create product
18 module.exports.create = async (req, res) => {
19
20 }
21
22 // show 1 product
23 module.exports.show = async (req, res) => {
24
25 }
26
27 // page edit product
28 module.exports.edit = async (req, res) => {
29
30 }
31
32 // update product
33 module.exports.update = async (req, res) => {
34
35 }
36
37 // delete product
38 module.exports.delete = async (req, res) => {
39
40 }

```

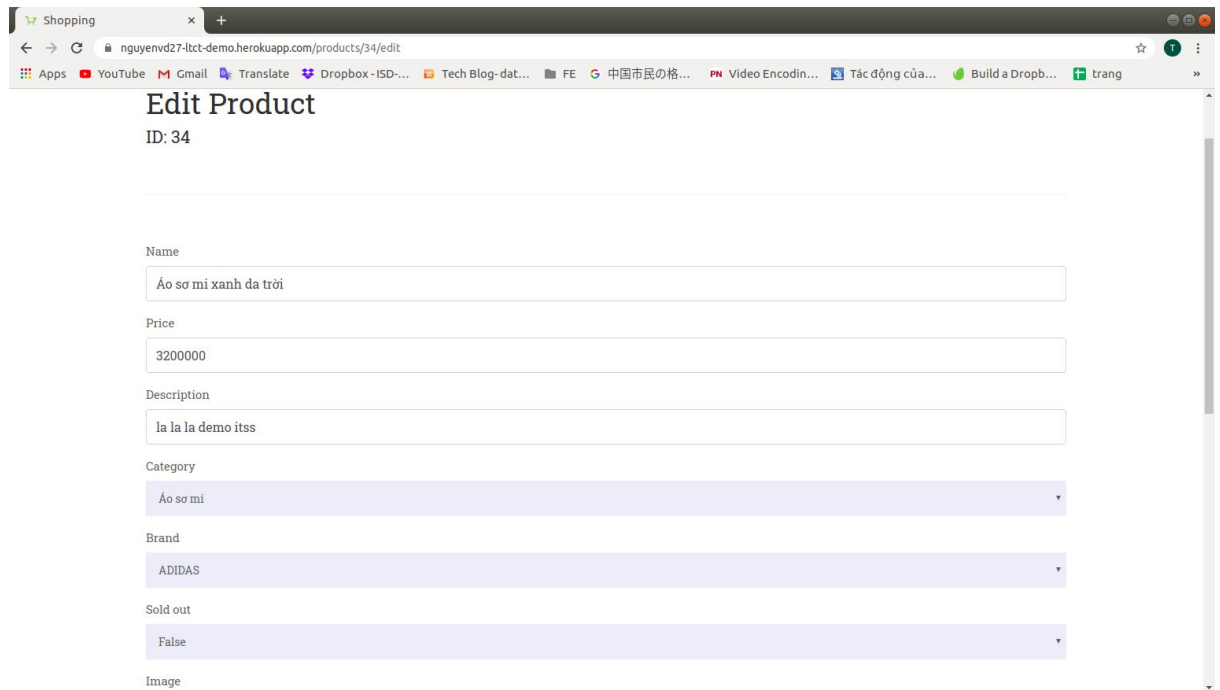
7. Hướng dẫn sử dụng

Sau khi admin đăng nhập thành công và chuyển đến module quản lý sản phẩm, admin sẽ xem được danh sách sản phẩm hiện có trong cơ sở dữ liệu.

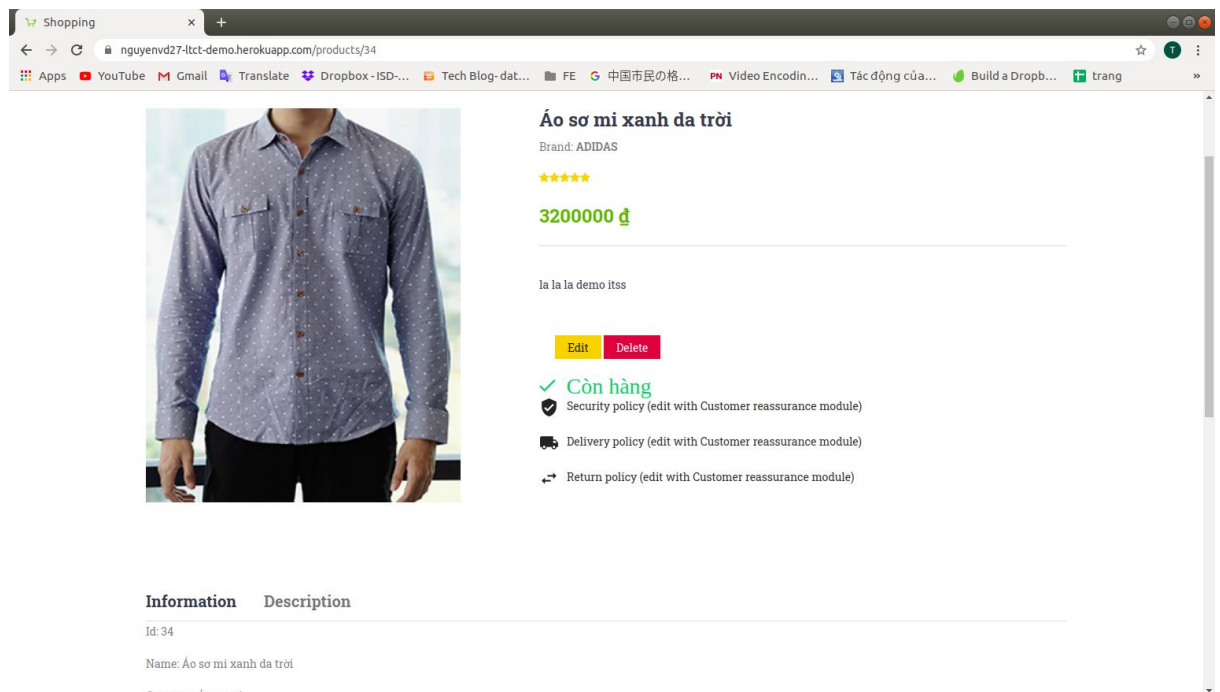
ID	Name	Image	Price	Category	Brand	Sold out	Edit	Delete
34	Áo sơ mi xanh da trời		3200000 đ	Áo sơ mi	ADIDAS	false	Edit	Delete
31	Áo sơ mi họa tiết xanh biển đậm		500000 đ	Áo sơ mi	nguyenvd	false	Edit	Delete
27	Áo thun tay dài xanh đen		999999 đ	Áo thun	ADIDAS	false	Edit	Delete
26	Áo sơ mi caro đỏ		399000 đ	Áo sơ mi	nike	false	Edit	Delete
25	Áo sơ mi sọc xanh xám		390000 đ	Áo sơ mi	nguyenvd	false	Edit	Delete

- Admin xóa sản phẩm bằng cách click vào button DELETE tương ứng ở mỗi sản phẩm.

- Click vào button EDIT bên cạnh sản phẩm để sửa thông tin sản phẩm



- Click vào ảnh của sản phẩm để xem chi tiết sản phẩm



- Tại trang hiển thị danh sách sản phẩm, click vào button NEW PRODUCT để tạo mới sản phẩm

Shopping

nguyenvd27-ltct-demo.herokuapp.com/products/new

Create Product

Name

Price

Description

Category [New Category](#)

Brand [New Brand](#)

Sold out

Image

- Danh sách Category

Shopping

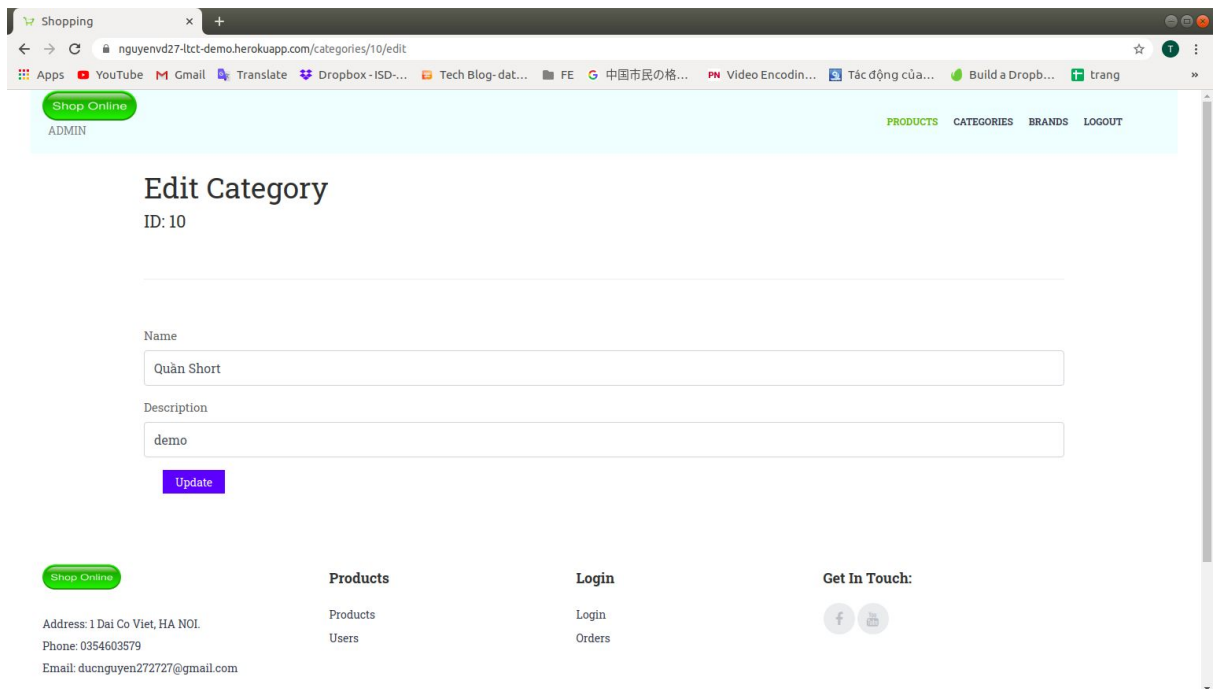
nguyenvd27-ltct-demo.herokuapp.com/categories/

Categories [New Category](#)

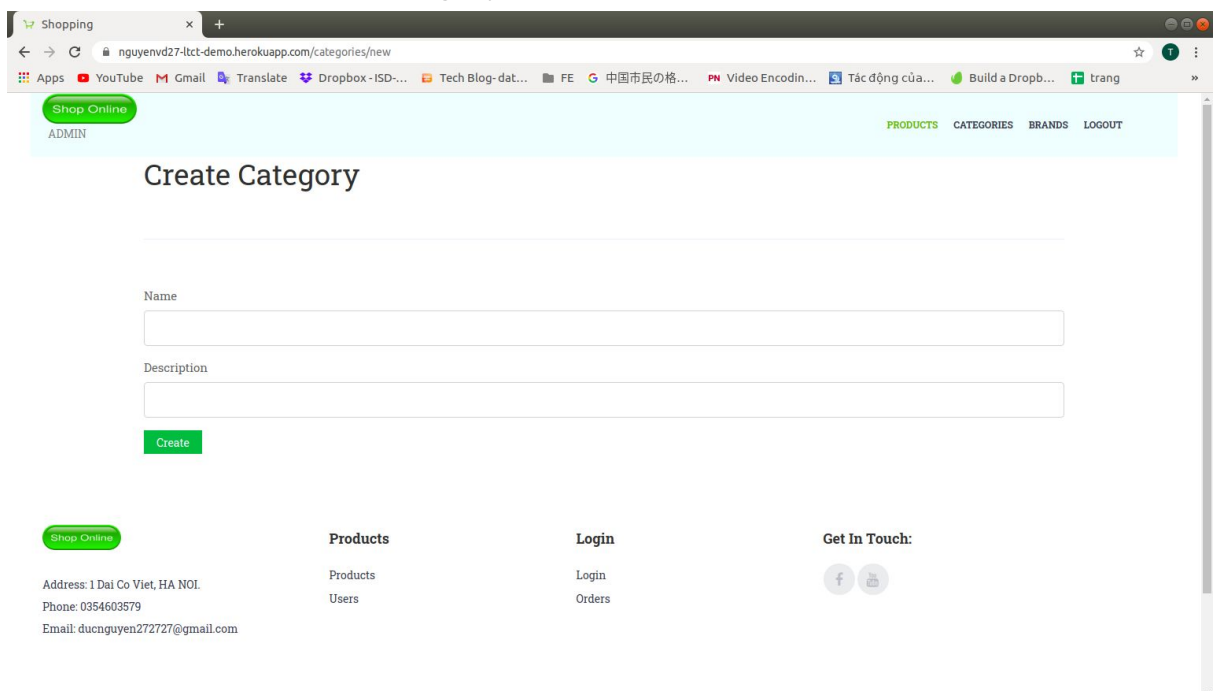
Total: 6

ID	Name	Description	Created At	Updated At	Edit	Delete
10	Quần Short	demo	Wed Dec 11 2019 16:00:03 GMT+0000 (Coordinated Universal Time)	Wed Dec 11 2019 16:00:03 GMT+0000 (Coordinated Universal Time)	Edit	Delete
5	Váy nam	demo	Tue Nov 19 2019 11:30:44 GMT+0000 (Coordinated Universal Time)	Tue Nov 19 2019 11:30:44 GMT+0000 (Coordinated Universal Time)	Edit	Delete
4	Quần jean	itss Cô Giang	Tue Nov 19 2019 11:30:27 GMT+0000 (Coordinated Universal Time)	Wed Dec 11 2019 15:38:57 GMT+0000 (Coordinated Universal Time)	Edit	Delete
3	Áo khoác	demo	Tue Nov 19 2019 11:30:10 GMT+0000 (Coordinated Universal Time)	Tue Dec 10 2019 15:43:47 GMT+0000 (Coordinated Universal Time)	Edit	Delete
2	Áo sơ mi	demo 12	Tue Nov 19 2019 11:29:55 GMT+0000 (Coordinated Universal Time)	Tue Nov 19 2019 11:29:55 GMT+0000 (Coordinated Universal Time)	Edit	Delete
1	Áo thun	demo category	Tue Nov 19 2019 11:29:37 GMT+0000 (Coordinated Universal Time)	Tue Nov 19 2019 11:31:15 GMT+0000 (Coordinated Universal Time)	Edit	Delete

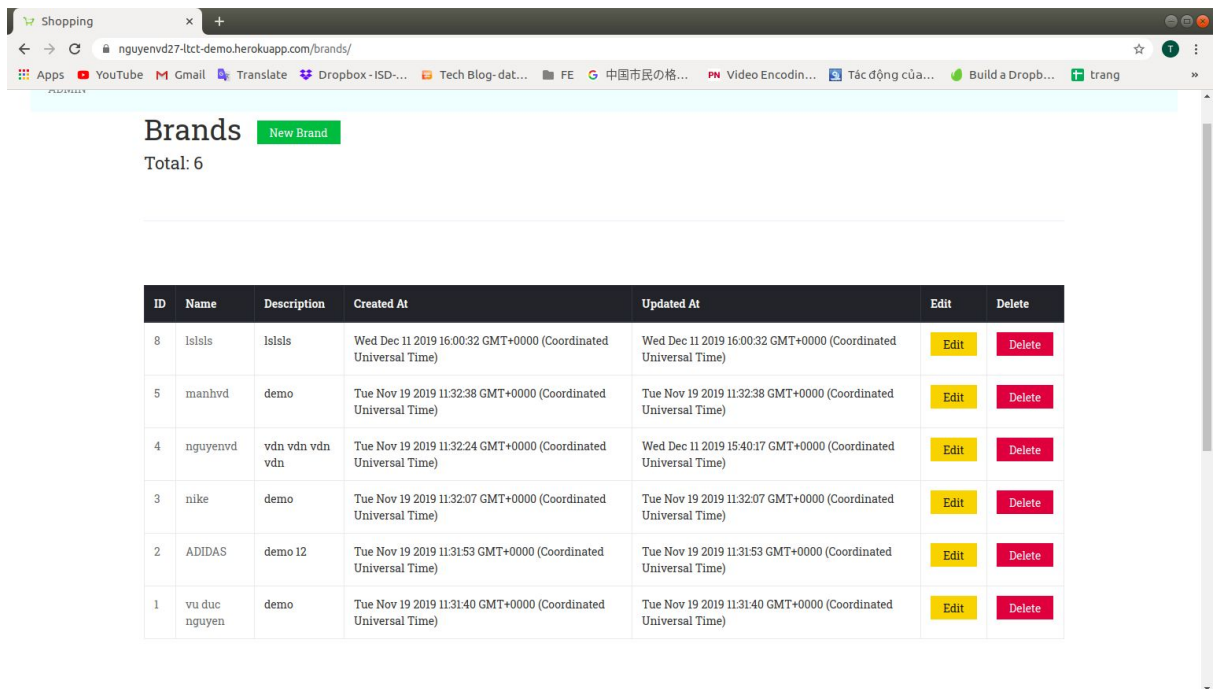
- Click vào button DELETE bên cạnh category để xóa category
- Click vào button EDIT bên cạnh category để sửa thông tin của category



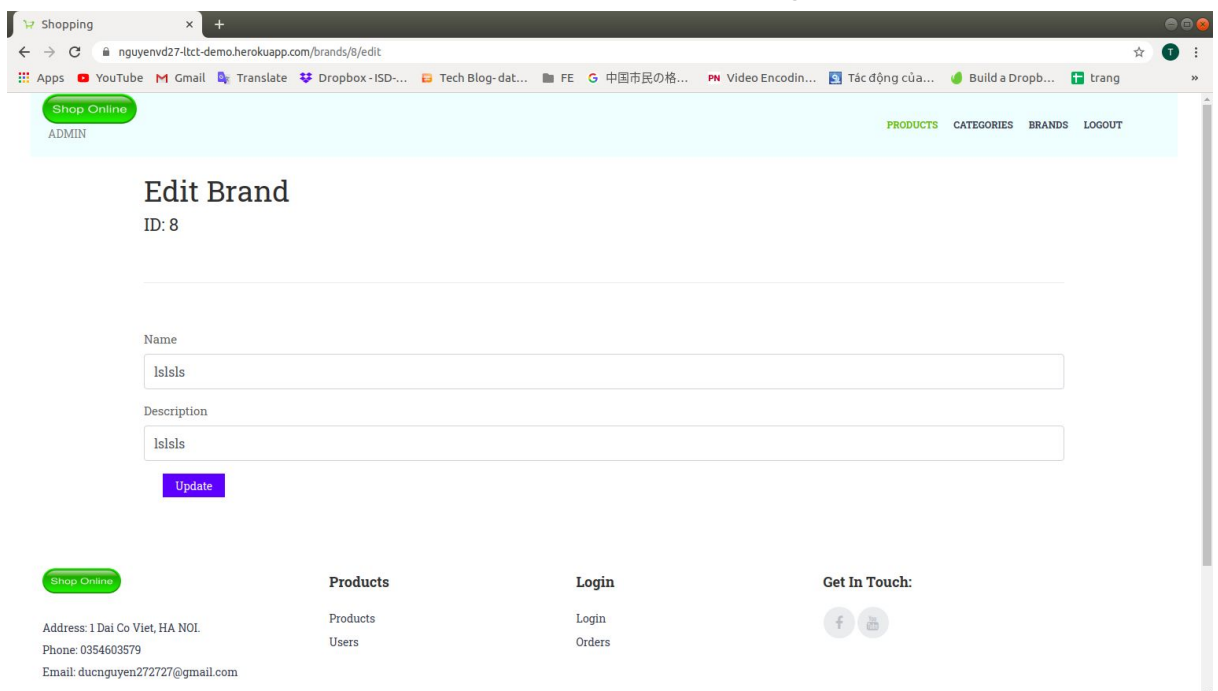
- Tại trang hiển thị danh sách category, click vào button NEW CATEGORY để tạo mới category



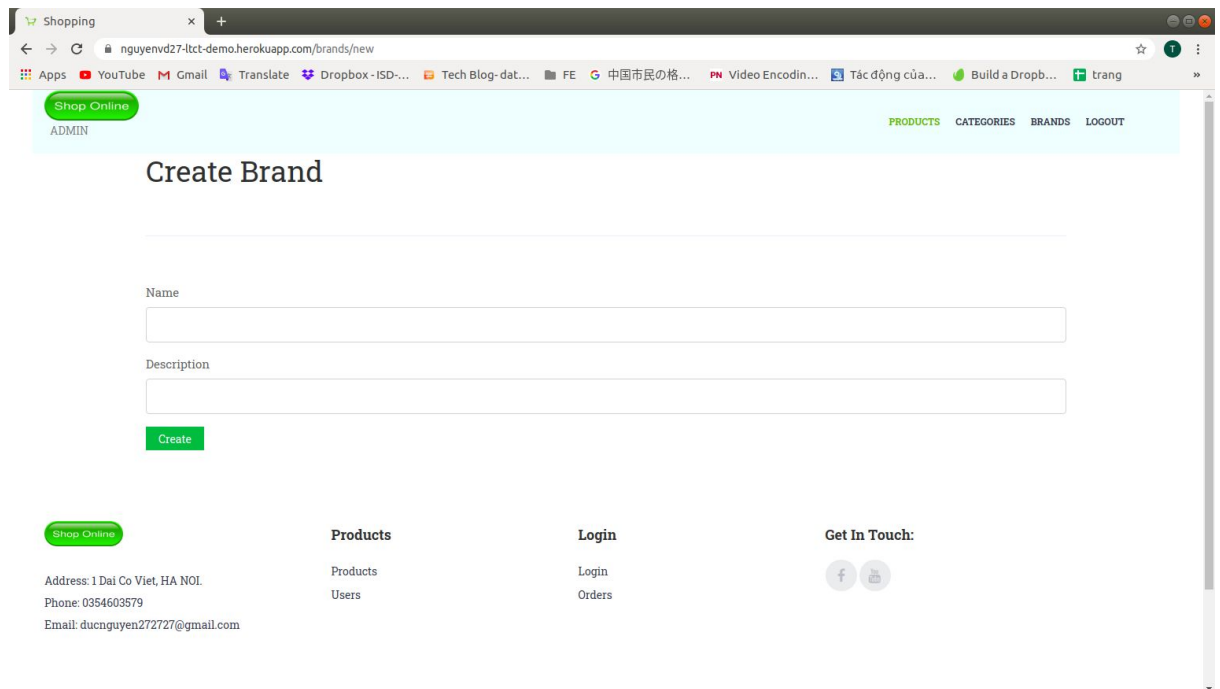
- Danh sách Brands



- Click vào button DELETE bên cạnh brand để xóa brand
- Click vào button EDIT bên cạnh brand để sửa thông tin của brand



- Tại trang hiển thị danh sách brand, click vào button NEW BRAND để tạo mới brand



8. Hướng dẫn cài đặt, triển khai, phát triển

8.1 Cài đặt và chạy project trên máy local (trên hệ điều hành Linux).

- Bước 1: Cài đặt nodejs và npm

Mở cửa sổ Terminal và gõ lần lượt các câu lệnh sau:

- sudo apt-get update
- sudo apt install nodejs

Để kiểm tra nodejs và npm đã có chưa sử dụng 2 câu lệnh sau:

node -v và npm -v

- Bước 2: Cài đặt hệ quản trị cơ sở dữ liệu postgresSQL

Mở cửa sổ Terminal và gõ lần lượt các câu lệnh sau:

- sudo apt update
- sudo apt install postgresql postgresql-contrib

- Bước 3: Clone repo trên github về với đường link sau:

https://github.com/nguyenvd27/structured_programming

- Bước 4: Di chuyển vào thư mục chứa project, mở Terminal lên và gõ npm install để cài đặt các gói cần thiết cho node và npm

- Bước 5: Chạy project trên máy local:

Mở Terminal lên và gõ npm start, project sẽ chạy ở cổng 3000 (có thể tùy chỉnh được)

Mở trình duyệt bất kỳ và gõ url: localhost:3000

Như vậy là đã hoàn thành xong cài đặt và chạy project trên local.

8.2 Phát triển project (trên hệ điều hành Linux).

Trong code của project có các comment code để dễ đọc hiểu, để phát triển tiếp project hãy đọc qua code.

Project được viết theo mô hình MVC, sử dụng một số design pattern, vì thế khi phát triển nên tuân theo MVC và các design pattern đó.

8.3 Hướng dẫn triển khai (deploy) trên heroku .

Bước 1: Tạo tài khoản trên heroku.

Bước 2: Tạo project trên heroku.

Bước 3: Thêm add-on postgre (hệ quản trị cơ sở dữ liệu) cho project vừa tạo trên heroku.

Bước 4: Trong thư mục project ở máy local, mở Terminal và gõ lần lượt các lệnh sau:

- heroku login (login vào heroku trên máy local)
- heroku git:remote -a "comment"

9. Phân công công việc

- Phân chia nhiệm vụ

STT	Nhiệm vụ	Công việc cụ thể		Phần trăm	Thành viên đảm nhiệm
1	Lập Trình (50%)	api	api phần products	10%	Vũ Đức Nguyễn
			api phần categories	5%	Vũ Đức Nguyễn
			api phần brands	5%	Vũ Đức Nguyễn
		giao diện	phần products	5%	Nguyễn Việt Hưng
			phần categories	5%	Lê Đức Anh
			phần brands	5%	Phùng Thị Trang
		Thiết kế cơ sở dữ liệu		5%	Nguyễn Việt Hưng
		Kết nối cơ sở dữ liệu với postgresSQL		5%	Lê Đức Anh
		Tái cấu trúc code (refactoring) +Triển khai (Deploy)		5%	Vũ Đức Nguyễn
2	Phân tích thiết kế (35%)	Xác định các design pattern		20%	Cả 4 cùng làm, mỗi người 5%
		Phân tích việc sử dụng SOLID		15%	Nguyễn Việt Hưng 5% Phùng Thị Trang 5% Lê Đức Anh 5%
3	Kiểm thử (10%)	Kiểm tra độ chính xác của api		5%	Phùng Thị Trang
		Kiểm tra dữ liệu đầu vào		5%	Nguyễn Việt Hưng
4	Báo cáo (5%)	Tổng hợp và viết báo cáo		5%	Phùng Thị Trang

- Đánh giá mức độ đóng góp

STT	Thành viên	Mức độ hoàn thành	Phần trăm đóng góp
1	Vũ Đức Nguyễn	100%	30%
2	Nguyễn Việt Hưng	100%	25%
3	Phùng Thị Trang	100%	25%
4	Lê Đức Anh	100%	20%

10. Kết luận

Trên đây là toàn bộ nội dung bài làm của chúng em. Qua các bài tập này, chúng em được hiểu sâu hơn về những lý thuyết mà cô dạy. Vì còn là sinh viên nên kiến thức còn hạn chế, chúng em rất mong nhận được sự đóng góp ý kiến của cô để bài làm tốt hơn nữa.

Qua phân hướng dẫn của cô trên lớp, nhiệm vụ phát triển module:

"Quản lý sản phẩm" của chúng em cũng được hoàn thành đầy đủ.

Link github: https://github.com/nguyenvd27/structured_programming

Link project: <https://nguyenvd27-ltct-demo.herokuapp.com/products/>

Link video demo: <https://www.youtube.com/watch?v=Bm6xpxaLFjA>

11. Tài liệu tham khảo

1. Slide bài giảng môn Lập Trình Cấu Trúc của cô Vũ Thị Hương Giang.
2. Trang chủ Expressjs (framework của nodejs): <https://expressjs.com/>
3. Tài liệu trên trang chủ Nodejs: <https://nodejs.org/en/docs/>