

ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



NIÊN LUẬN CHUYÊN NGÀNH

**XÂY DỰNG HỆ THỐNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ
VỚI JAVA SPRING VÀ REACTJS**

Sinh viên thực hiện: Nguyễn Hứa Phong Phú

Giáo viên hướng dẫn: Vũ Duy Linh

MSSV: C2000027

Ngành: Công nghệ thông tin

Chuyên ngành: Tin học ứng dụng

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

GIÁO VIÊN HƯỚNG DẪN

CHỮ KÝ

LỜI CẢM ƠN

Quá trình thực hiện niên luận cơ sở là giai đoạn đầu quan trọng trong quãng đời mỗi sinh viên khoa Công Nghệ Thông Tin – Truyền Thông. Niên luận cơ sở là tiền đề nhằm trang bị cho chúng em những kỹ năng nghiên cứu, những kiến thức quý báu.

Trước hết, em xin chân thành cảm ơn Trường Công Nghệ Thông Tin và Truyền Thông Đại học Cần Thơ đã tạo điều kiện thuận lợi cho chúng em học tập và thực hiện đề tài này.

Em xin chân thành cảm ơn quý Thầy, Cô Trường Công Nghệ Thông Tin và Truyền Thông Trường Đại học Cần Thơ đã tận tình chỉ dạy và trang bị cho em những kiến thức cần thiết trong suốt thời gian ngồi trên ghế giảng đường, lấy đó làm nền tảng cho em có thể hoàn thành được bài niên luận chuyên ngành này.

Em xin trân trọng cảm ơn thầy Vũ Duy Linh đã tận tình giúp đỡ, định hướng cách tư duy và cách làm việc. Đó là những góp ý hết sức quý báu không chỉ trong quá trình thực hiện niên luận mà còn là hành trang tiếp bước không thể nào quên cho em trong quá trình học tập và lập nghiệp sau này.

Và cuối cùng, xin gửi lời cảm ơn đến gia đình, bạn bè, những người luôn sẵn sàng sẻ chia và giúp đỡ trong học tập và cuộc sống. Mong rằng, chúng ta sẽ mãi mãi gắn bó với nhau.

Dù đã cố gắng hoàn thành đề tài trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Rất mong nhận được sự thông cảm, góp ý và tận tình chỉ bảo của quý Thầy Cô.

Lời cuối cùng, em xin kính chúc các thầy cô đặt biệt là thầy Vũ Duy Linh thật là nhiều sức khỏe, thành công và hạnh phúc.

MỤC LỤC

CHƯƠNG I : TỔNG QUAN.....	7
1. Mục tiêu đề tài	7
2. Yêu cầu của đề tài	8
2.1 Yêu cầu chung	8
2.2 Xây dựng website thương mại điện tử.....	8
2.3 Các chức năng chính	8
2.4 Đối tượng và phạm vi nghiên cứu	8
2.5 Phương pháp nghiên cứu.....	9
2.6 Nội dung nghiên cứu	9
CHƯƠNG II. CƠ SỞ NGHIÊN CỨU LÝ THUYẾT.....	10
1. Giới thiệu về API và RESTful API.....	10
2. Java.....	13
3. Spring	15
4. MongoDB	16
5. ReactJS	18
5.1. JSX	19
5.2. Virtual DOM.....	20
6. React-Hooks	24
7. Axios	25
CHƯƠNG III. NỘI DUNG	27
1. Yêu cầu tối thiểu của đề tài	27
2. Xây dựng chức năng website thương mại điện tử	27
3. Mô tả nghiệp vụ	28
3.1 Dành cho khách hàng nói chung	28
3.2 Dành cho quản trị viên	29
4. Yêu cầu hệ thống	29
4.1 Yêu cầu chức năng.....	29
4.2 Yêu cầu phi chức năng.....	30

5. Các sơ đồ	31
5.1 Use case diagram	31
5.2 Entity diagram (sơ đồ thực thể).....	35
5.3 Relationship diagram (sơ đồ quan hệ)	36
5.4 Sequence diagram (Sơ đồ tuần tự).....	47
5.5 Functional diagram (Sơ đồ chức năng).....	49
6. Kiểm thử	52
CHƯƠNG IV : KẾT LUẬN.....	57
1. Kết quả đạt được	57
1.1 Ưu điểm.....	57
2.2 Nhược điểm	57
2. Hướng phát triển	58

DANH SÁCH CÁC SƠ ĐỒ

Sơ đồ 1. Sơ đồ Use Case mức 0 của khách hàng.....	31
Sơ đồ 2. Sơ đồ Use Case mức 1 của khách hàng.....	32
Sơ đồ 3. Sơ đồ mức 0 của quản trị viên	33
Sơ đồ 4. Sơ đồ mức 1 của quản trị viên	34
Sơ đồ 5. Sơ đồ thực thể biểu hiện các thành phần của ứng dụng	35
Sơ đồ 6. Sơ đồ quan hệ biểu thị các mối liên hệ của các thành phần trong ứng dụng	36
Sơ đồ 7. Sơ đồ tuần tự thêm sản phẩm của khách hàng.....	47
Sơ đồ 8. Sơ đồ tuần tự mua hàng của khách hàng	48
Sơ đồ 9. Sơ đồ chức năng của khách hàng viếng thăm.....	49
Sơ đồ 10. Sơ đồ chức năng của khách hàng thành viên.....	50
Sơ đồ 11. Sơ đồ chức năng của quản trị viên	51

DANH SÁCH CÁC ẢNH

Hình ảnh 1. Logo Rest API	10
Hình ảnh 2. Miêu tả cách hoạt động của Rest API.....	12
Hình ảnh 3. Ví dụ gọi phương thức GET trong Postman.....	13
Hình ảnh 4. Logo Java.....	13
Hình ảnh 5. Logo Spring	15
Hình ảnh 6. Logo MongoDB	16
Hình ảnh 7. Logo ReactJS	18
Hình ảnh 8. Ví dụ so sánh giữa React Element và JSX	20
Hình ảnh 9. Ảnh so sánh giữa Real DOM và Virtual DOM	21
Hình ảnh 10. Ảnh miêu tả quá trình làm việc của Real DOM và Virtual DOM.....	22
Hình ảnh 11. Logo React - Hooks	24
Hình ảnh 12. Logo Axios	25
Hình ảnh 13. Hình ảnh so sánh giữa Axios và Fetch.....	26

DANH SÁCH CÁC BẢNG

Bảng 1. Nội dung nghiên cứu đề tài.....	9
Bảng 2. Bảng các phương thức cơ bản của Rest API	11
Bảng 3. Bảng so sánh các khái niệm cơ bản trong MongoDB với các sơ sở dữ liệu quan hệ khác.....	17
Bảng 4. Bảng liệt kê danh sách các thực thể	37
Bảng 5. Bảng đơn nhập hàng.....	38
Bảng 6. Bảng thương hiệu	38
Bảng 7. Bảng nhà cung cấp	39
Bảng 8. Bảng đơn nhập hàng.....	39
Bảng 9. Bảng loại sản phẩm	40
Bảng 10. Bảng sản phẩm.....	41
Bảng 11. Bảng các thẻ.....	42
Bảng 12. Bảng các thẻ thuộc sản phẩm.....	42
Bảng 13. Bảng chi tiết hóa đơn.....	43

Bảng 14. Bảng quản trị viên	43
Bảng 15. Bảng bình luận	44
Bảng 16. Bảng khách hàng	45
Bảng 17. Bảng hóa đơn	46

CHƯƠNG I : TỔNG QUAN

1. MỤC TIÊU ĐỀ TÀI

Những tiến bộ to lớn về công nghệ hiện đại, đặc biệt là công nghệ thông tin trong những thập niên cuối của thế kỷ 20 đã tạo ra bước ngoặt mới cho sự phát triển kinh tế xã hội toàn cầu. Chính vì điều đó, một phương thức thương mại mới đã xuất hiện và phát triển nhanh chóng, đó là thương mại điện tử.

Thương mại điện tử chính là một công cụ hiện đại sử dụng mạng Internet giúp cho các doanh nghiệp có thể thâm nhập và nhanh chóng phát triển vào thị trường thế giới, thu thập thông tin nhanh hơn, nhiều hơn và chính xác hơn.

Với thương mại điện tử, các doanh nghiệp cũng có thể đưa các thông tin về sản phẩm của mình đến các đối tượng khách hàng tiềm năng lớn khác nhau ở mọi nơi trên thế giới với chi phí thấp hơn nhiều so với các phương pháp truyền thống.

Trong xu thế phát triển mạnh mẽ của thương mại điện tử trên thế giới, các doanh nghiệp Việt Nam đặc biệt là các doanh nghiệp vừa và nhỏ, bộ phận chiếm tỷ lệ lớn trong tổng số các doanh nghiệp Việt Nam, cũng đã bước đầu nhận thức được ích lợi và tầm quan trọng của việc ứng dụng thương mại điện tử.

Tuy nhiên, do còn nhiều hạn chế trong nhận thức của bản thân các doanh nghiệp cũng như các điều kiện cơ sở hạ tầng, việc ứng dụng thương mại điện tử trong các doanh nghiệp vừa và nhỏ ở Việt Nam có thể nói mới ở mức độ sơ khởi.

Vì thế, các doanh nghiệp vừa và nhỏ cần xây dựng cho mình một chiến lược kinh doanh phù hợp với điều kiện thực tế hiện nay nhằm tiếp cận nhiều hơn nữa với thương mại điện tử để có thể khai thác tối đa lợi ích mà phương thức kinh doanh này đem lại.

Ngày nay, các cửa hàng trực tuyến xuất hiện càng nhiều. Cửa hàng trực tuyến sẽ đem lại nhiều lợi ích về cho người sử dụng. Đối với doanh nghiệp, tiết kiệm được không gian trưng bày sản phẩm, tiết kiệm nguồn nhân lực, tiết kiệm chi phí mặt bằng...

Đối với khách hàng, tiết kiệm được thời gian đi lại, lựa chọn được những sản phẩm ưa thích... Đây cũng là lý do để em chọn đề tài: **“Xây dựng website thương mại điện tử về thời trang”** làm đề tài môn học “Niên luận chuyên ngành”.

Chính vì vậy, việc xây dựng website P&P SHOP được xây dựng để nắm bắt xu hướng thời đại, cũng là để doanh nghiệp tiếp cận với số lượng khách hàng mới, giúp cho các sản phẩm tiếp cận được với nhiều khách hàng, đem lại doanh thu lớn cho doanh nghiệp.

Không những thế, đây cũng là một cơ hội tốt để có thể trao đổi học hỏi những điều mới từ những kiến thức đã học và sẽ học để đủ điều kiện nâng cấp bản thân.

2. YÊU CẦU CỦA ĐỀ TÀI

2.1 Yêu cầu chung

- Tìm hiểu về cách thức hoạt động kinh doanh thương mại.
- Tìm hiểu về các giải pháp để xây dựng một website thương mại điện tử như thế nào.
- Xây dựng một website mang tính thương mại hóa.
- Xây dựng một trang web với đầy đủ các chức năng để phục vụ cho Quản Trị Viên nói riêng và Khách Hàng nói chung.
- Đòi hỏi một môi trường thân thiện với người tiêu dùng.

2.2 Xây dựng website thương mại điện tử.

- Website thương mại điện tử có một số chức năng như sau:
 - + Đối với khách hàng: Xem danh sách các sản phẩm, Xem thông tin chi tiết một sản phẩm, Tìm kiếm sản phẩm, Đưa sản phẩm vào giỏ hàng, Đặt mua, Thanh toán, Góp ý, Liên hệ,...
 - + Đối với các quản trị viên: Quản lý sản phẩm, quản lý nhân viên, Quản lý đơn hàng, Quản lý khách hàng, Quản lý các bình luận...

2.3 Các chức năng chính

Website thời trang được xây dựng với tiêu chí phục vụ cho tất cả mọi người, trong lĩnh vực thời trang với mong muốn đáp ứng đầy đủ các mặt hàng, các chức năng của một website bán hàng trực tuyến.

Website bao gồm các loại người dùng chính: Khách hàng viếng thăm, Khách hàng thành viên, Các quản trị viên:

- + Đối với Khách hàng viếng thăm: Xem danh sách sản phẩm, Xem chi tiết sản phẩm, Đưa sản phẩm vào giỏ hàng, cập nhật giỏ hàng, Tìm kiếm, Đăng Ký...
- + Đối với Khách hàng thành viên: Ngoài những chức năng như khách hàng viếng thăm còn có thêm một số chức năng như Đặt mua, Xem danh sách đơn hàng, Quản lý đơn hàng, Thanh Toán, Đăng nhập...
- + Đối với các Quản trị viên: Được xem như là một khách hàng thành viên và còn có chức năng quản lý tùy thuộc vào chức năng như: Quản lý sản phẩm, đơn hàng, khách hàng, quản lý thông tin khác.

2.4 Đối tượng và phạm vi nghiên cứu

Website thời trang được xây dựng với tiêu chí phục vụ cho tất cả mọi người có đam mê về thời trang với mong muốn đáp ứng đầy đủ các mặt hàng, các chức năng của một website bán hàng trực tuyến.

Nghiên cứu và tham khảo những trang web về thời trang nổi bật hiện nay để lấy đó làm nền tảng mở rộng kiến thức.

Nắm bắt được những điểm nổi bật của công nghệ về thời trang hiện nay, những sản phẩm tốt và tiên quyết để làm nên thương hiệu riêng của trang web từ đó phát triển duy trì trang web một cách thuận lợi.

2.5 Phương pháp nghiên cứu

Nội dung nghiên cứu khá phổ biến và đầy đủ, do đó tài liệu nghiên cứu rất nhiều nên không tránh khỏi tình trạng không biết bắt đầu từ đâu và như thế nào nhưng chủ yếu là tìm hiểu thông tin tài liệu thông qua các video có nhiều lượt xem và đầy đủ nội dung, sách báo và các bài viết bổ ích đáp ứng đủ tiêu chí trên mạng.

Từ đó, vận dụng các kiến thức đã tiếp thu được vào đề tài. Tương ứng với mỗi thành phần của hệ thống sẽ có những phương pháp nghiên cứu khác nhau.

2.6 Nội dung nghiên cứu

STT	Đối tượng	Nội dung
1	API và RESTful API	<ul style="list-style-type: none"> - Tìm hiểu về Rest và API - Restful API là gì ? - Các phương thức cơ bản ? - Nguyên tắc hoạt động của Restful API
2	Java	<ul style="list-style-type: none"> - Tìm hiểu về Java - Các tính năng của Java
3	Spring Framework	<ul style="list-style-type: none"> - Tìm hiểu Spring Framework - Hệ sinh thái của Spring Framework
4	MongoDB	<ul style="list-style-type: none"> - Tìm hiểu hệ quản trị cơ sở dữ liệu NOSQL – MongoDB - So sánh một số khái niệm trong MongoDB và các cơ sở dữ liệu quan hệ (RDBMS) khác - Khi nào nên sử dụng MongoDB
5	ReactJS	<ul style="list-style-type: none"> - Tìm hiểu ReactJS - JSX trong ReactJS - Ví dụ về JSX - Virtual Dom là như thế nào? - Cách ReactJs sử dụng Virtual Dom
6	React Hooks	<ul style="list-style-type: none"> - React Hooks là gì? - Tại sao lại sử dụng React Hooks
7	Axios	<ul style="list-style-type: none"> - Tìm hiểu về Axios - Tại sao nên sử dụng Axios thay vì Fetch

Bảng 1: Nội dung nghiên cứu đề tài

CHƯƠNG II. CƠ SỞ NGHIÊN CỨU LÝ THUYẾT

1. GIỚI THIỆU VỀ API VÀ RESTFUL API



Hình ảnh 1: Logo Rest API

Nguồn ảnh: <https://cdn.vockan.com/2022/wp-content/uploads/2023/03/02194049/restapi2-1.png.webp>

REST (REpresentational State Transfer) được đưa ra vào năm 2000, trong luận văn tiến sĩ của Roy Thomas Fielding (đồng sáng lập giao thức HTTP). Nó là một dạng chuyển đổi cấu trúc dữ liệu, là một phong cách kiến trúc cho việc thiết kế các ứng dụng có kết nối. Nó sử dụng HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

Vì một phần mềm chứa rất nhiều logic phức tạp, nên người ta tìm cách chia nhỏ nó ra thành nhiều phần, mỗi phần này tạm gọi là một component. Mỗi component sẽ có tính độc lập cao, ít phụ thuộc hoặc có thể không phụ thuộc vào các thành phần khác. Tuy là có tính độc lập cao, nhưng để có thể kết nối được với nhau mà một phần mềm hoàn chỉnh, buộc chúng vẫn phải tuân theo một hoặc một số chuẩn nào đó. Thì mỗi cái chuẩn đó được gọi là một giao diện lập trình ứng dụng – hay chính là một API.

RESTful API (hay còn gọi là REST API) là một giao diện lập trình ứng dụng (API hay web API) tuân theo các ràng buộc của kiểu kiến trúc REST, cho phép tương tác với các dịch vụ web RESTful.

Hay nói đơn giản, RESTful API là một tiêu chuẩn được dùng trong việc thiết kế API dành cho các ứng dụng web (thiết kế Web Services) để hỗ trợ cho việc quản lý các resource. REST là viết tắt của cụm từ Representational state transfer (ứng dụng chuyển đổi cấu trúc dữ liệu), được tạo ra lần đầu bởi nhà khoa học máy tính Roy Fielding ở những năm 2000.. REST API không được xem là một công nghệ, nó là một giải pháp để tạo ra các ứng dụng web services thay thế cho các kiểu khác như SOAP, WSDL (Web Service Definition Language),...

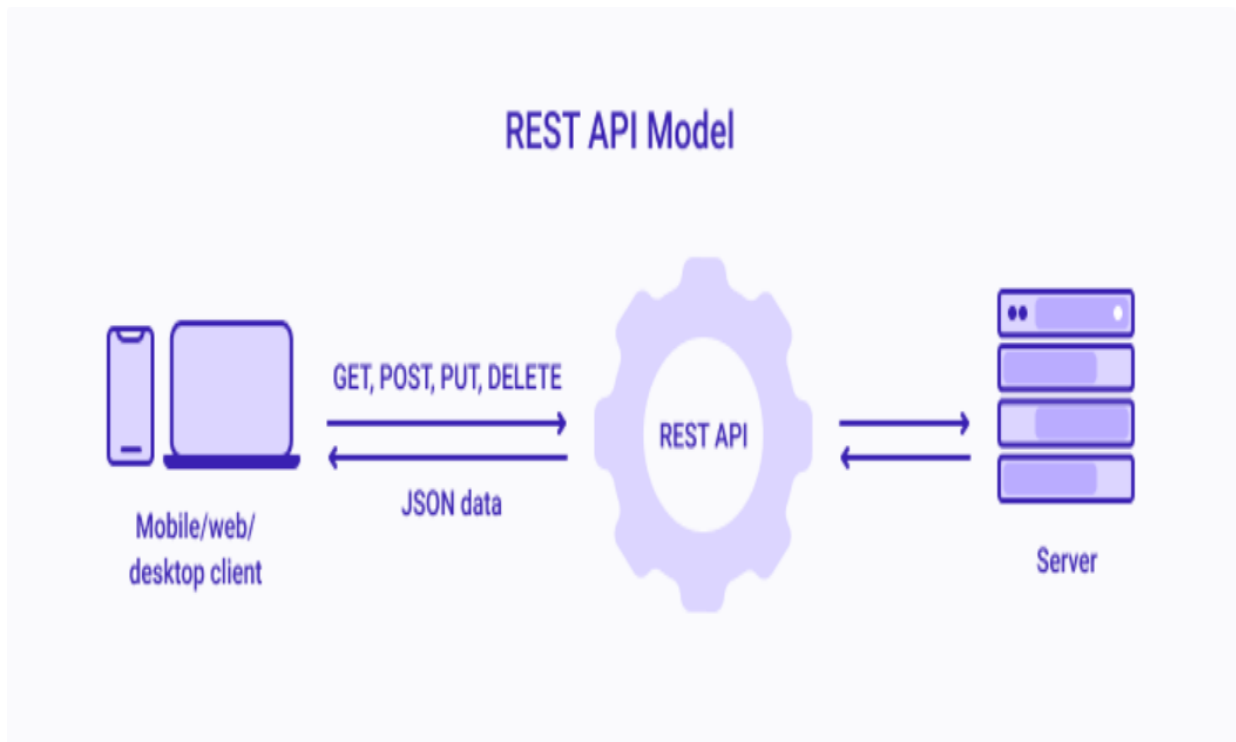
Các phương thức HTTP	Tác vụ
GET	Lấy dữ liệu
POST	Tạo mới dữ liệu
PUT	Cập nhật dữ liệu (toàn bộ)
PATCH	Cập nhật dữ liệu (một phần)
DELETE	Xóa dữ liệu

Bảng 2: Bảng phương thức cơ bản của Rest API

Các tác vụ đọc, tạo, cập nhật, xóa được gọi là CRUD service (Create, Read, Update, Delete). Mỗi tác vụ trên phải được thông qua địa chỉ URI (Uniform Resource Identifier) kèm theo phương thức payload (có thể có hoặc không, thường được định dạng kiểu XML hoặc JSON).

RESTful API sử dụng giao thức stateless (là một giao thức truyền thông không sử dụng session) và theo chuẩn nên hệ thống sẽ nhanh, đáng tin cậy và có thể mở rộng dễ dàng. Thông thường, RESTful API sẽ xác thực người dùng khi gửi yêu cầu đối với những tác vụ nguy hiểm như cập nhật hoặc xóa dữ liệu chỉ cho phép người quản trị.

RESTful API hoạt động như thế nào?



Hình ảnh 2: Miêu tả cách hoạt động của Rest API

Nguồn ảnh: <https://webkul.com/blog/wp-content/uploads/2022/08/rest2.png>

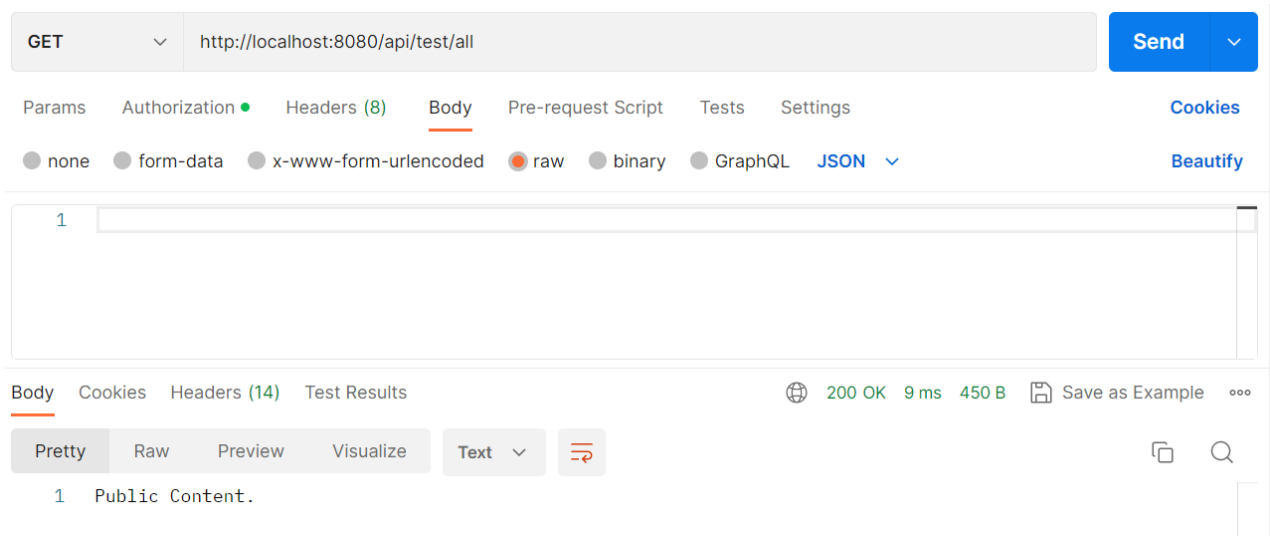
Bước 1: Client gửi một yêu cầu đến máy chủ. Client làm theo tài liệu API để định dạng yêu cầu theo cách mà máy chủ hiểu được.

Bước 2: Máy chủ xác thực và xác nhận máy khách có quyền đưa ra yêu cầu đó.

Bước 3: Máy chủ nhận yêu cầu và xử lý trong nội bộ.

Bước 4: Máy chủ trả về một phản hồi đến client. Phản hồi chứa thông tin cho client biết liệu yêu cầu có thành công hay không. Phản hồi cũng bao gồm bất kỳ thông tin nào mà client yêu cầu.

⇒ Chi tiết về phản hồi và yêu cầu API REST sẽ khác nhau đôi chút tùy thuộc vào cách các nhà phát triển API thiết kế API.



Hình ảnh 3: Ví dụ gọi phương thức GET trong Postman

2. JAVA



Hình ảnh 4: Logo Java

Nguồn ảnh: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQV0PsENAJMI2ahFush6VdOOKdq920rnB0WcQ&usqp=CAU>

Java được biết đến là một ngôn ngữ bậc cao, hướng đối tượng giúp bảo mật mạnh mẽ, và còn được định nghĩa là một Platform. Java được phát triển bởi Sun Microsystems, do James Gosling khởi xướng và ra mắt năm 1995. Java hoạt động trên rất nhiều nền tảng như Windows, Mac và các phiên bản khác nhau của UNIX.

Java chính là ngôn ngữ lập trình có mục đích, hướng đối tượng được phát triển cho môi trường phân tán và phát triển phần mềm. Ngôn ngữ này độc lập với nền tảng, không có giới hạn đối với bất kỳ phần cứng hay hệ điều hành cụ thể nào. Nó cung cấp cho người dùng cơ sở để có thể “viết một lần, chạy ở mọi nơi”.

Các tính năng của Java:

- **Hướng đối tượng:** Trong Java, tất cả đều là một Object. Java có thể mở rộng và bảo trì dễ dàng bởi nó được xây dựng dựa trên mô hình Object.
- **Nền tảng độc lập:** Khi được biên dịch, Java không được biên dịch thành ngôn ngữ máy trên nền tảng cụ thể mà thay vào mã byte – một nền tảng độc lập. Mã byte này được thông dịch từ máy ảo (JVM) trên một nền tảng nào đó mà nó đang chạy.
- **Đơn giản:** Java được thiết kế đơn giản, dễ học. Chỉ hiểu khái niệm cơ bản về OOP Java, để trở thành master về java rất dễ.
- **Bảo mật:** Tính năng an toàn của Java cho phép phát triển các hệ thống không virus, không giả mạo, việc xác thực dựa trên mã hoá khóa công khai.
- **Kiến trúc – trung lập:** Trình biên dịch của Java tạo ra các định dạng tệp đối tượng kiến trúc trung lập, khiến mã biên dịch được thực thi trên nhiều bộ vi xử lý thông qua hệ điều hành Java.
- **Portable:** Một loại kiến trúc trung lập, phụ thuộc vào việc thực hiện, là những đặc điểm chính khi nói về Portable của Java. Các trình biên dịch tại Java được viết bằng ANSI C với ranh giới Portable gọn gàng, gọi là Subset POSIX, bạn có thể mang Byte Code của Java lên bất cứ một nền tảng nào.
- **Mạnh mẽ:** Java luôn nỗ lực loại trừ các tình huống dễ bị lỗi thông qua việc kiểm tra lỗi tại thời điểm biên dịch và kiểm tra lỗi tại runtime.
- **Đa luồng:** Giúp tạo ra các chương trình thực hiện cùng lúc nhiều tác vụ, cho phép các nhà phát triển xây dựng ứng dụng tương tác chạy trơn tru hơn.
- **Thông dịch:** Các mã byte Java được dịch trực tiếp tới các máy tính gốc và không được lưu trữ ở bất cứ đâu.
- **Hiệu năng cao:** Khi sử dụng trình biên dịch Just-In-Time, Java cho phép thực hiện hiệu năng cao.
- **Phân tán:** Java được sử dụng thiết kế cho môi trường phân tán của Internet.
- **Năng động:** Java năng động hơn C hoặc C++ do được thiết kế với mục đích thích ứng môi trường đang phát triển. Các chương trình Java sở hữu lượng lớn thông tin tại runtime có thể được sử dụng để xác minh hay giải quyết các truy cập vào các đối tượng tại runtime.

3. SPRING



Hình ảnh 4: Logo Spring framework

Nguồn ảnh: <https://viblo.asia/p/sping-oi-minh-hoc-tu-dau-the-phan-1-tong-quan-ve-spring-framework-XL6lARbN5ek>

Spring Framework là gì?

- Spring là một trong những framework dựa trên ngôn ngữ lập trình Java. Nó được cung cấp một mô hình lập trình và cấu hình mạnh mẽ để phát triển những ứng dụng doanh nghiệp.
- Việc tạo ra Spring framework được thực hiện bởi mong muốn đơn giản hóa việc phát triển ứng dụng trên nền tảng công nghệ Java EE (Java Enterprise Edition) phổ biến của Oracle, vốn rất phức tạp và khó sử dụng vào thời đó.
- Spring framework sở hữu những ưu điểm nổi trội hướng tới những nghiệp vụ chuyên biệt của một khung phần mềm. Tuy nhiên, điều thực sự làm Spring Framework trở nên có giá trị là hệ sinh thái Spring Ecosystem phát triển mạnh mẽ trong những năm qua và vẫn đang được hỗ trợ để phát triển nhiều hơn trong tương lai.

Hệ sinh thái của Spring Framework:

Từ cấu hình đến bảo mật, từ phát triển ứng dụng web (web apps) đến dữ liệu lớn (big data), bất kể nhu cầu cơ sở hạ tầng của ứng dụng là gì, đều có các Spring Project giúp bạn xây dựng nó.

Spring được thiết kế theo mô-đun. Vậy nên lập trình viên có thể bắt đầu với quy mô nhỏ và chỉ sử dụng những module họ cần. Đến khi mở rộng hệ thống, họ cũng dễ dàng sử dụng các module anh em trong hệ sinh thái.

Mặc dù có một danh sách dài các module hỗ trợ trong hệ sinh thái của Spring framework và nó liên tục thay đổi, nhưng có một số module đáng được nhắc tới mà bạn nên biết:

- **Spring Boot:** Đưa ra một bộ thư viện để xây dựng ứng dụng Spring nhanh chóng và đáng tin cậy.
- **Spring Cloud:** Cung cấp một tập hợp các công cụ hỗ trợ các hệ thống phân tán, hữu ích để xây dựng và triển khai các hệ thống microservice.
- **Spring Security:** Bảo vệ ứng dụng của bạn với việc triển khai xác thực và ủy quyền có thể mở rộng.
- **Spring Data:** Cung cấp một cách truy cập dữ liệu nhất quán để truy cập tới các cơ sở dữ liệu quan hệ SQL và NoSQL và hơn thế nữa.
- **Spring REST Docs:** Cho phép lập trình viên mô tả các API RESTful bằng các tài liệu tự viết với các tài liệu chính xác được tạo tự động.

=> Ngoài ra, hệ sinh thái của Spring Framework cũng có rất nhiều bên thứ 3 hỗ trợ. Nhờ vậy, việc sử dụng Spring Framework cũng rất tiện lợi và dễ dàng.

4. MONGODB



Hình ảnh 5: Logo MongoDB

Nguồn ảnh: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRctFfD0UngqW-cec6Q1Ex4EKE3ngMD-TTdfq&usqp=CAU>

MongoDB lần đầu ra đời bởi MongoDB Inc., tại thời điểm đó là thế hệ 10, vào tháng Mười năm 2007, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine. Sau đó nó đã được chuyển thành nguồn mở từ năm 2009.

MongoDB là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

So sánh một số khái niệm trong MongoDB và các cơ sở dữ liệu quan hệ (RDBMS) khác:

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (mặc định là _id)

Bảng 3: Bảng so sánh các khái niệm cơ bản trong MongoDB với các cơ sở dữ liệu quan hệ khác

=> Trong đó ta thấy key _id là do hệ thống tự tạo ra một field như vậy làm khóa chính.

Khi nào nên sử dụng MongoDB:

MongoDB thật sự rất hot nhưng không phải lúc nào ta sử dụng nó cũng tốt, có những trường hợp không nên sử dụng và nên sử dụng, chi tiết thế nào thì chúng ta cùng thảo luận nhé.

- Nếu website của bạn có tính chất INSERT cao, bởi vì mặc định MongoDB có sẵn cơ chế ghi với tốc độ cao và an toàn.

- Website của bạn ở dạng thời gian thực nhiều, nghĩa là nhiều người thao tác với ứng dụng. Nếu trong quá trình load bị lỗi tại một điểm nào đó thì nó sẽ bỏ qua phần đó nên sẽ an toàn.

- Website bạn có nhiều dữ liệu quá, giả sử web bạn có đến 10 triệu records thì đó là cơn ác mộng với MYSQL. Bởi vì MongoDB có khả năng tìm kiếm thông tin liên quan cũng khá nhanh nên trường hợp này nên dùng nó.

- Máy chủ không có hệ quản trị CSDL, trường hợp này thường bạn sẽ sử dụng SQLITE hoặc là MongoDB.

5. REACTJS



Hình ảnh 6: Logo ReactJS

Nguồn ảnh: <https://t3h.com.vn/tin-tuc/install-reactjs>

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,... Giả sử bạn đang lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm ưng ý rồi và nhấn vào nút “Thêm vào giỏ hàng”, thì việc tiếp theo mà bạn phải làm đó là thêm sản phẩm được chọn vào giỏ hàng và hiển thị lại sản phẩm đó khi user vào xem => xử lý tương tác.

Trước khi có ReactJS, lập trình viên thường gặp rất nhiều khó khăn trong việc sử dụng “vanilla JavaScript”(JavaScript thuần) và JQuery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển ứng dụng sẽ lâu hơn và xuất hiện nhiều bug, rủi ro hơn. Vì vậy vào năm 2011, Jordan Walke – một nhân viên của Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.

Hơn nữa, để tăng tốc quá trình phát triển và giảm thiểu những rủi ro có thể xảy ra trong khi coding, React còn cung cấp cho chúng ta khả năng Reusable Code (tái sử dụng code) bằng cách đưa ra 2 khái niệm quan trọng bao gồm:

- JSX.
- Virtual DOM.

Để hiểu rõ hơn về ReactJS và tại sao bạn nên sử dụng nó, chúng ta cùng nhau tìm hiểu 2 khái niệm trên để xem chúng thực sự làm việc như thế nào.

5.1. JSX

JSX (JavaScript extension) là một React extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản. Và kể từ lúc ReactJS browser hỗ trợ toàn bộ những trình duyệt Web hiện đại, bạn có thể tự tin sử dụng JSX trên bất kỳ trình duyệt nào mà bạn đang làm việc.

Tại sao nên sử dụng JSX cho trang Web?

Thứ nhất, JSX với cú pháp gần giống XML, cấu trúc cây khi biểu thị các attributes, điều đó giúp ta dễ dàng định nghĩa, quản lý được các component phức tạp, thay vì việc phải định nghĩa và gọi ra nhiều hàm hoặc object trong javascript. Khi nhìn vào cấu trúc đó cũng dễ dàng đọc hiểu được ý nghĩa của các component. Code JSX ngắn hơn, dễ hiểu hơn code JS.

Thứ 2, JSX không làm thay đổi ngữ nghĩa của Javascript.

Thứ 3, với cách viết gần với các thẻ HTML, nó giúp những developers thông thường (ví dụ như các designer) có thể hiểu được một cách dễ dàng, từ đó có thể viết hoặc sửa code mà không gặp nhiều khó khăn.

Ví dụ dùng JSX như sau:

```
<div className="red">Children Text</div>;  
<MyCounter count={3 + 5} />;  
var gameScores = {  
  player1: 2,  
  player2: 5  
};  
<DashboardUnit data-index="2">
```

```

    <h1>Scores</h1>
    <Scoreboard className="results" scores={gameScores} />
  </DashboardUnit>;

```

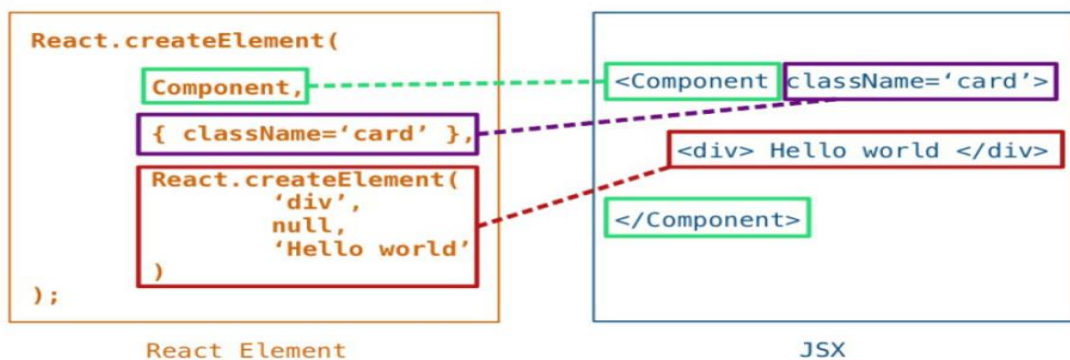
Code Javascript tương đương sẽ là:

```

    React.createElement("div", {className: "red"}, "Children
    Text");
    React.createElement(MyCounter, {count: 3 + 5});
    React.createElement(DashboardUnit, {"data-index": "2"},
    React.createElement("h1", null, "Scores"),
    React.createElement(Scoreboard, {className: "results",
    scores: gameScores}));

```

Ví dụ khác:



Hình ảnh 7: Ví dụ so sánh giữa React Element và JSX

Nguồn ảnh: <https://www.geeksforgeeks.org/react-jsx-in-depth/>

⇒ Có thể thấy là sử dụng JSX sẽ ngắn gọn hơn và sạch sẽ hơn rất nhiều khi sử dụng Javascript thuần

5.2. Virtual DOM

Trước tiên cùng tìm hiểu trước về Real DOM:

Real DOM hay là DOM thật. Đầu tiên DOM được hiểu là viết tắt của (Document Object Model). DOM sẽ đại diện cho giao diện người dùng trong ứng dụng của bạn. Mỗi khi có sự thay đổi về trạng thái của giao diện người dùng, DOM sẽ cập nhật để thể hiện sự thay đổi đó. Việc thường xuyên thao tác lên DOM sẽ gây giảm hiệu suất, làm ứng dụng của bạn chậm hơn.

Điều gì làm giảm hiệu suất của DOM:

DOM được biểu diễn dưới dạng cấu trúc dữ liệu cây (tree). Do đó các thay đổi và cập nhật đối với DOM diễn ra nhanh chóng. Nhưng sau mỗi lần thay đổi thì phần tử được cập nhật và các phần tử con của nó sẽ hiển thị lại để cập nhật giao diện người dùng. Việc render và hiển thị lại là nguyên nhân khiến cho ứng dụng của bạn chậm đi. Do đó, bạn càng có nhiều thành phần giao diện người dùng, thì các bản cập nhật DOM càng nhiều, vì chúng sẽ cần được hiển thị lại cho mỗi bản cập nhật DOM

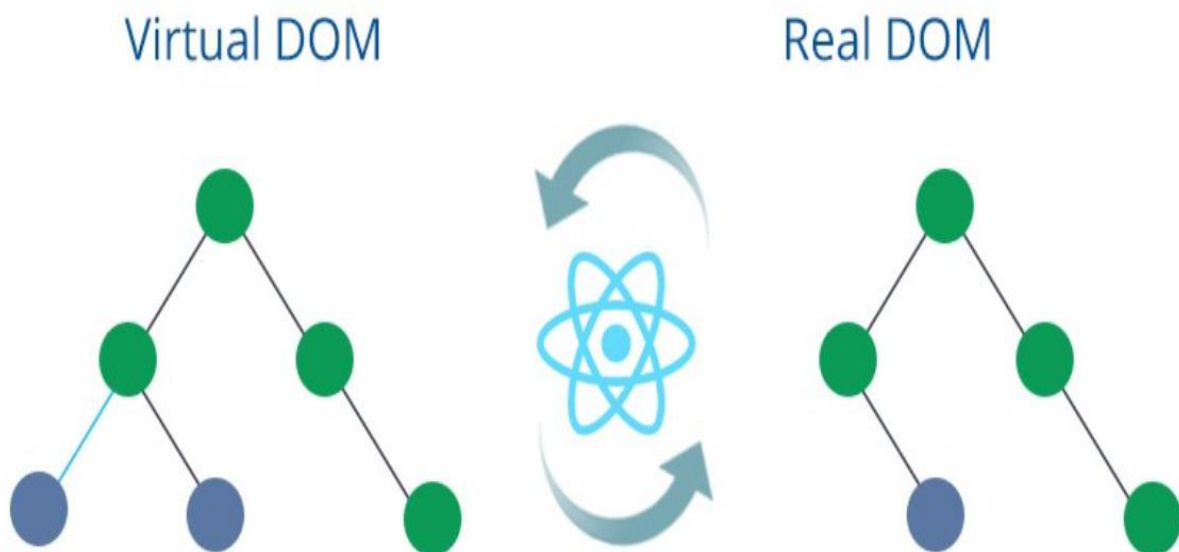
Cuối cùng là Virtual DOM:

Đó là lúc khái niệm virtual DOM (DOM ảo) xuất hiện và hoạt động tốt hơn đáng kể so với DOM thực. DOM ảo chỉ là một đại diện ảo của DOM. Mỗi khi trạng thái ứng dụng i thay đổi, DOM ảo được cập nhật thay vì DOM thực

Tại sao Virtual Dom (DOM ảo) lại nhanh hơn?

Khi các phần tử mới được thêm vào giao diện người dùng, một DOM ảo, được biểu thị dưới dạng cây sẽ được tạo. Mỗi phần tử là một nút trên cây này. Nếu trạng thái của bất kỳ phần tử nào trong số này thay đổi. Cây này sau đó được so sánh hoặc "diffing" với cây DOM ảo trước đó.

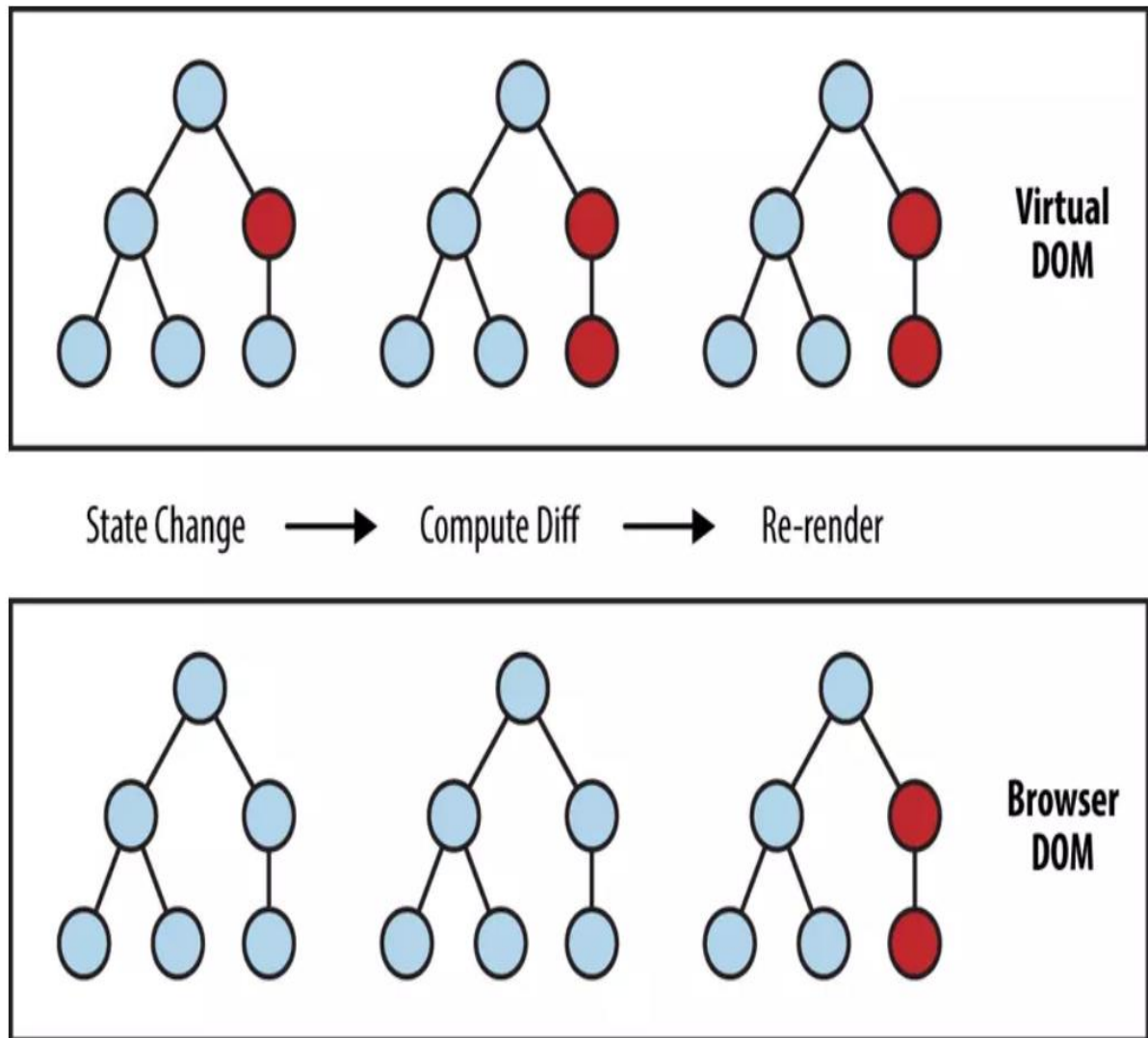
Khi điều này được thực hiện xong, DOM ảo sẽ tính toán phương pháp tốt nhất có thể để thực hiện những thay đổi này đối với DOM thực. Điều này đảm bảo rằng có các hoạt động tối thiểu trên DOM thực. Do đó, giảm chi phí hiệu suất của việc cập nhật DOM thực.



Hình ảnh 6: Ảnh so sánh giữa Real DOM và Virtual DOM

Nguồn ảnh: <https://blog.arrowwhitech.com/virtual-dom-its-definition-and-benefits-that-you-must-know/>

Hình ảnh bên dưới cho thấy cây DOM ảo và quá trình khác nhau:



Hình ảnh 8: Ảnh miêu tả quá trình làm việc của Real DOM và Virtual DOM

Nguồn ảnh: <https://dev.to/swarnaliroy94/introduction-to-react-real-dom-virtual-dom-363>

Các nút màu đỏ đại diện cho các nút đã thay đổi. Các nút này đại diện cho các phần tử giao diện người dùng đã bị thay đổi trạng thái. Sau đó sẽ tính toán sự khác biệt giữa phiên bản trước của cây DOM ảo và cây DOM ảo hiện tại. Toàn bộ cây con chính sau đó được render lại để cung cấp giao diện người dùng. Cây cập nhật này sau đó được cập nhật hàng loạt vào DOM thực.

Cách ReactJS sử dụng Virtual DOM:

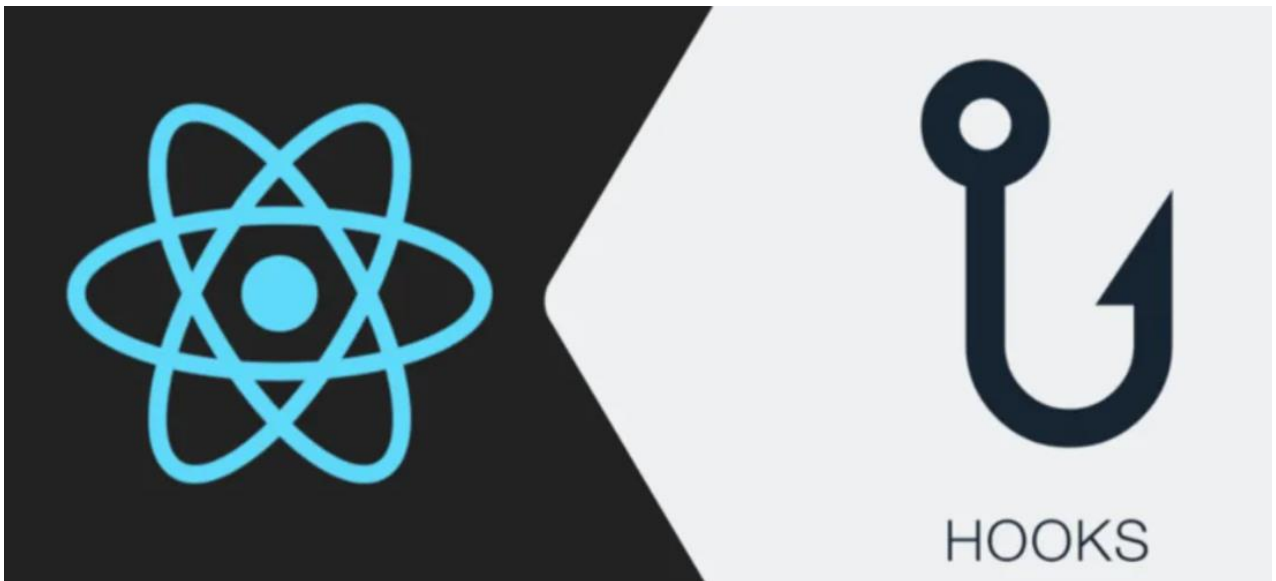
Trong React, mỗi phần giao diện người dùng là một thành phần (component) và mỗi thành phần có một trạng thái. React tuân theo mô hình có thể quan sát được và lắng nghe các thay đổi trạng thái. Khi trạng thái của một thành phần thay đổi, React sẽ cập nhật cây DOM ảo. Khi DOM ảo đã được cập nhật, React sau đó sẽ so sánh phiên bản hiện tại của DOM ảo với phiên bản trước của DOM ảo. Quá trình này được gọi là "diffing".

Khi React biết đối tượng DOM ảo nào đã thay đổi, thì React chỉ cập nhật các đối tượng đó trong DOM thực. Điều này làm cho hiệu suất tốt hơn nhiều so với thao tác trực tiếp với DOM thực. Do đó làm cho React nổi bật như một thư viện JavaScript hiệu suất cao.

Tóm lại:

- Thao tác DOM thường xuyên rất tốn kém và hiệu suất nặng.
- Khi thay đổi trạng thái xảy ra, DOM ảo được cập nhật và phiên bản trước đó và phiên bản hiện tại của DOM ảo được so sánh. Điều này được gọi là "diffing".
- Sau đó, DOM ảo sẽ gửi một bản cập nhật hàng loạt đến DOM thực để cập nhật giao diện người dùng.
- React sử dụng DOM ảo để nâng cao hiệu suất của nó.
- Nó sử dụng những gì có thể quan sát được để phát hiện những thay đổi trạng thái và chống đỡ.
- React sử dụng một thuật toán *DIFFING* hiệu quả để so sánh các phiên bản của DOM ảo.
- Sau đó, nó đảm bảo rằng các bản cập nhật theo đợt được gửi đến DOM thực để sơn lại hoặc kết xuất lại giao diện người dùng.

6. REACT – HOOKS



Hình ảnh 9: Logo React – Hooks

Nguồn ảnh: <https://t3h.com.vn/tin-tuc/reactjs-hook-va-nhung-dieu-ma-lap-trinh-vien-can-biet>

React hooks là tính năng mới, hứa hẹn rất nhiều thú vị được giới thiệu trong bản React 16.8. Với React hooks, bạn sẽ có trong tay nhiều công cụ hơn để phát triển ứng dụng web của mình.

Nếu bạn muốn sử dụng state hoặc các hàm liên quan đến lifecycle, thông thường bạn sẽ phải sử dụng `React.Component`. Với Hook thì khác, bạn hoàn toàn có thể sử dụng chúng trong các function component.

React Hooks gồm những hàm được thiết kế sẵn và mỗi hàm sẽ có một công dụng riêng các phương thức xử lý riêng. Điều này sẽ khiến các components gọn nhẹ hơn, giảm lượng code thừa thãi. Đặc biệt, các bạn có ít kinh nghiệm với Javascript sẽ dễ tiếp cận với React hơn.

Khi sử dụng hook có 2 điều cần lưu ý. Đầu tiên, chỉ có thể sử dụng hook trong functional component. Thứ hai, chỉ có thể sử dụng hook ở level cao nhất của function. Không thể sử dụng chúng trong các mệnh đề điều kiện, trong loops, hoặc trong các nested function.

Tầm quan trọng và phát triển của React Hooks trong tương lai:

Nếu đã từng làm việc với React các bạn đã từng nghe qua khái niệm Class Component và Functional Component và tại React Conf năm 2018 (React Conf là hội nghị Front-End lớn nhất tại Iran chuyên chia sẻ kiến thức về React đến với cộng đồng phát triển và yêu thích ReactJs) các diễn giả với nhiều năm kinh nghiệm làm việc với React đã cho biết rằng các dự định cho việc phát triển Web với React sau này khuyến nghị nên sử dụng Functional Component thay vì Class Component và đi kèm với React Hooks để code có thể sạch sẽ hơn, dễ hiểu hơn và với một cộng đồng React Hooks ngày càng lớn thì việc phát triển của nó cũng sẽ trở nên nhanh chóng hơn và mạnh mẽ hơn.

Nguồn React Conf 2018: <https://www.youtube.com/watch?v=dpw9EHDh2bM&t=1364s>

7. AXIOS



Hình ảnh 10: Logo Axios

Nguồn ảnh: https://commons.wikimedia.org/wiki/File:Axios_%28computer_library%29_logo.svg

Axios là một HTTP client được viết dựa trên Promises được dùng để hỗ trợ cho việc xây dựng các ứng dụng API từ đơn giản đến phức tạp và có thể được sử dụng cả ở trình duyệt.

Việc tạo ra một HTTP request dùng để fetch hay lưu dữ liệu là một trong những nhiệm vụ thường thấy mà một ứng dụng Javascript phía client cần phải làm khi muốn giao tiếp với phía server.

Tại sao nên sử dụng Axios thay vì Fetch ?

Axios vs Fetch

Hình ảnh 11: Hình ảnh so sánh giữa Axios và Fetch

Nguồn ảnh: <https://velog.velcdn.com/images/eunbinn/post/c8d8a1e4-4fa0-49df-bb86-f78a7818b7de/image.png>

Các browser hiện đại bây giờ thường đi kèm và xây dựng sẵn các tính năng Fetch API mới hơn, vậy tạo sao ta lại không dùng nó luôn cho xong? Rất nhiều những sự khác biệt giữa **Axios** và **Fetch** khiến rất nhiều người vẫn lựa chọn **Axios** thay vì **Fetch**.

Một trong những đặc điểm lớn ta có thể thấy đó là cách mà 2 thư viện xử lý với các HTTP error code. Khi sử dụng Fetch, nếu khi server trả về các mã lỗi 4xx hay 5xx, thì hàm catch() của bạn sẽ không được gọi đến và người lập trình viên sẽ có nhiệm vụ phải tự kiểm tra trạng thái của mã trả về để xác định xem liệu request đó có thành công hay không. Trong khi đó, **Axios** sẽ reject tất cả các promise của request nếu một trong các mã lỗi trên được trả về.

Một điểm khác biệt nho nhỏ khác, điều mà không ít các lập trình viên mới trong việc xây dựng API gặp phải đó là việc **Fetch** không tự động gửi trả cookies về cho server khi tạo một request. Ta sẽ cần phải truyền một cách trực tiếp các option để cho cookies có thể được include. Còn với **Axios** thì không phải lo về vấn đề này.

Sự khác biệt có thể được coi là vô cùng đáng chú ý đối với rất nhiều lập trình viên đó là khả năng cập nhật tiến độ của những lần uploads/downloads. Vì **Axios** được xây dựng dựa trên các XHR API cũ hơn, nên có thể khai báo các hàm callback cho onUploadProgress và onDownloadProgress để hiện thị phần trăm thành công tại giao diện cho app. Và cho đến hiện tại thì Fetch vẫn chưa hỗ trợ gì cho việc này.

Cuối cùng, **Axios** có thể được sử dụng ở cả browser và chính điều này đã tạo cơ hội cho việc chia sẻ code Javascript giữa các trình duyệt và phần back-end hay việc thực hiện render cho ứng dụng front-end của bạn ở phía server.

Một trong các cách thường thấy để cài đặt **Axios** đó là thông qua trình quản lý các npm package.

CHƯƠNG III. NỘI DUNG

1. YÊU CẦU TỐI THIỂU CỦA ĐỀ TÀI

- Tìm hiểu về cách thức hoạt động kinh doanh thương mại.
- Tìm hiểu về các giải pháp để xây dựng một website thương mại điện tử như thế nào.
- Xây dựng một website mang tính thương mại hóa.
- Xây dựng một trang web với đầy đủ các chức năng để phục vụ cho Quản Trị viên nói riêng và Khách Hàng nói chung.
- Đòi hỏi một môi trường thân thiện với người tiêu dùng.

2. XÂY DỰNG CHỨC NĂNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ

Website thương mại điện tử có một số chức năng như sau tương ứng với từng đối tượng tương ứng:

+ *Đối với khách hàng viếng thăm:*

- Xem danh sách sản phẩm
- Xem thông tin chi tiết từng sản phẩm
- Tìm kiếm sản phẩm
- Đưa sản phẩm vào giỏ hàng
- Cập nhật giỏ hàng
- Đăng ký thành viên

+ *Đối với khách hàng thành viên:*

- Xem danh sách sản phẩm
- Xem thông tin chi tiết từng sản phẩm
- Tìm kiếm sản phẩm
- Đưa sản phẩm vào giỏ hàng
- Cập nhật giỏ hàng
- Đặt mua
- Thanh toán
- Xem lịch sử mua hàng
- Bình luận sản phẩm

+ *Đối với quản trị:*

- Quản trị sản phẩm
- Quản trị đơn hàng
- Quản trị khách hàng
- Quản trị thành viên
- Quản trị các góp ý

3. MÔ TẢ NGHIỆP VỤ

3.1 Dành cho khách hàng nói chung

Website được xây dựng với mục tiêu là nơi trưng bày, giới thiệu các Sản Phẩm và các dịch vụ hỗ trợ Khách Hàng như: Góp Ý, Tìm Kiếm, Mua Hàng, ...

Mỗi Khách Hàng sẽ có các trường như: MaKhachHang, TenKhachHang, SDTKhachHang, DiaChiKhachHang, EmailKhachHang, TaiKhoan, MatKhai. Với MaKhachHang sẽ tạo nên sự riêng biệt độc nhất của từng Khách Hàng làm cho Khách Hàng không bị trùng lặp giúp cho phần thiết kế lấy thông tin Khách Hàng dễ dàng hơn. TaiKhoan và MatKhai của Khách Hàng giúp họ có thể đăng nhập vào trang web để mua hàng và các thuộc tính còn lại dùng để lưu thông tin của Khách Hàng vào Hóa Đơn tiện cho việc lấy thông tin và liên lạc.

Khách Hàng bắt đầu tìm kiếm những Sản Phẩm phù hợp để mua, Sản Phẩm sẽ có các thuộc tính riêng như TenSanPham, Gia, MoTa, SoLuong, TenThuongHieu, TenNhaCungCap, ... với MaSanPham là khóa chính. Với khóa chính riêng biệt như thế việc thao tác với các Sản Phẩm dễ dàng hơn. Khách Hàng có thể lọc các Sản Phẩm thông qua TenSanPham hoặc LoaiSanPham mình cần trên website, các Sản Phẩm được bố trí, bày bán theo từng LoaiSanPham, từng Gia hoặc đa dạng như Sản Phẩm mới (mới là Tag), Sản Phẩm bán chạy (bán chạy là Tag), ... giúp cho Khách Hàng dễ dàng tìm kiếm Sản Phẩm mình cần hơn. Khi Khách Hàng đã chọn được một Sản Phẩm ưng ý, Khách Hàng có thể nhấp chọn vào Sản Phẩm để xem chi tiết thông tin Sản Phẩm như: TenThuongHieu, TenNhaCungCap, MoTa, ...

Sau đó Khách Hàng có thể chọn nút “Thêm vào giỏ hàng” nếu quá trình mua sắm vẫn chưa kết thúc. Nếu muốn Thanh Toán chọn nút “Mua” hoặc nhấn nút “Giỏ Hàng”. Giỏ hàng đơn thuần là danh sách các Sản Phẩm mà Khách Hàng đã chọn bao gồm những thông tin như: TenSanPham, Gia, Số lượng, Thành tiền, ... Phân biệt từng Sản Phẩm riêng biệt qua MaSanPham.

Giỏ hàng cung cấp chức năng xóa các Sản Phẩm trong giỏ hàng, cập nhật số lượng Sản Phẩm muốn mua.

Ngay sau khi Khách Hàng đã chọn được các Sản Phẩm cần mua, có thể chọn nút “Đặt Mua”. Lúc này website sẽ lấy thông tin từ tài khoản của Khách Hàng (HoTenKhachHang, DiaChiKhachhang, EmailKhachHang, SDTKhachHang, ...) thông tin về yêu cầu của Khách (nếu có), sau cùng là thông tin về giỏ hàng (MaSanPham, TenSanPham, Gia, số lượng, thành tiền, ...). Sau khi đặt hàng các thông tin từ giỏ hàng sẽ được lưu vào Hóa Đơn, các Hóa Đơn sẽ có MaHoaDon riêng và các thông tin từ giỏ hàng mà Khách Hàng đã đặt. Nếu Khách Hàng chưa đăng nhập trước đó thì cần phải đăng nhập để có thể đặt hàng.

Sau khi Khách Hàng điền đầy đủ thông tin cần thiết, chọn nút “Đặt Hàng” để hoàn tất quá trình đặt hàng và chờ nhận hàng thôi.

3.2 Dành cho quản trị viên

Website cung cấp cho Quản Trị Viên (admin) toàn quyền quản lý website như: CRUD Sản Phẩm, Quản lý các đơn hàng, Quản lý Khách Hàng, Quản lý các Bình Luận,...

4. YÊU CẦU HỆ THỐNG

4.1 Yêu cầu chức năng

Miêu tả chi tiết các chức năng dành cho từng đối tượng chính:

Khách hàng viếng thăm:

- Xem danh sách các sản phẩm: Khi người dùng truy cập vào website hệ thống sẽ hiển thị tự động các danh sách Sản Phẩm một cách ngẫu nhiên. Bên cạnh đó người dùng có thể chọn các danh mục Sản Phẩm để có thể lựa chọn các loại Sản Phẩm phù hợp.
- Xem thông tin chi tiết từng sản phẩm: Sau khi tìm thấy Sản Phẩm phù hợp Khách Hàng có thể nhấn chọn vào Sản Phẩm đó để xem chi tiết của Sản Phẩm.
- Tìm kiếm sản phẩm: Nếu Khách Hàng muốn tìm kiếm bất kỳ một Sản Phẩm nào đó thì có thể nhập thông tin Sản Phẩm muốn tìm lên thanh công cụ tìm kiếm để thực hiện chức năng tìm kiếm. Sản Phẩm sẽ được hiển thị lên giao diện nếu có. Ngược lại sẽ hiển thị “Không tìm thấy kết quả”.
- Đưa các sản phẩm vào giỏ hàng: Sau khi khách hàng tìm thấy sản phẩm đáp ứng nhu cầu của mình thì có thể chọn đưa sản phẩm vào giỏ hàng.
- Cập nhật giỏ hàng: Khi người dùng đã lựa chọn một sản phẩm nhưng tìm được một sản phẩm hợp lý hơn thì có thể xóa sản phẩm đã có trong giỏ hàng vào chọn lại sản phẩm mới. Ngoài ra, khi cần mua sản phẩm đó với một số lượng khác thì người dùng cũng có thể cập nhật trong giỏ hàng.
- Đăng ký thành viên: Khi khách hàng viếng thăm muốn mua hàng thì có thể thực hiện chức năng đăng ký để trở thành thành viên của website và thực hiện các chức năng mua hàng.

Khách hàng thành viên: Ngoài có những các chức năng của khách hàng viếng thăm, khách hàng thành viên còn có những chức năng sau:

- Đặt mua: Sau khi xem thông tin sản phẩm đúng với nhu cầu thì người dùng có thể chọn đặt mua để đưa sản phẩm vào giỏ hàng và có thể thanh toán.

- Thanh toán sản phẩm: Khi đã chọn Sản Phẩm và quyết định chắc chắn mua chúng, Khách Hàng có thể chọn chức năng thanh toán để thực hiện xác nhận đơn hàng cho mình. Muốn thực hiện này thì yêu cầu người dùng phải đăng nhập vào hệ thống. Trong quá trình thực hiện chức năng thanh toán thì người dùng có thể thấy được danh sách sản phẩm mà mình đặt mua trong giỏ hàng. Khách hàng cần điền một số thông tin mô tả nếu có. Sau khi thực hiện xong các bước thì người dùng xác nhận đơn hàng.

- Xem lịch sử mua hàng: Khách hàng có thể xem lại các Sản Phẩm mà đã từng mua trên website.

- Bình luận sản phẩm: Chức năng này cho phép Khách Hàng đánh giá các Sản Phẩm mình đã từng mua trên website.

Quản trị viên website: Có tất cả các chức năng của Khách Hàng Thành Viên và có thêm chức năng quản lý khác như:

- Quản lý Sản Phẩm, Đơn Hàng, Khách Hàng và một số thông tin khác. Để thực hiện chức năng quản lý thì yêu cầu các Quản Trị Viên phải đăng nhập hệ thống. Quản Trị Viên chọn mục mình cần thực hiện quản lý như Sản Phẩm. Trong quản lý Sản Phẩm có xem, thêm, sửa, xóa Sản Phẩm. Chọn thao tác cần thực hiện, điền đầy đủ thông tin trong quá trình thực hiện rồi xác nhận thao tác.

4.2 Yêu cầu phi chức năng

- Dung lượng website phải vừa phải, tốc độ truy xuất cao, hạn chế downtime không cần thiết. Ngoài ra, cần đảm bảo chế độ bảo mật cho trang web nói riêng và thông tin Khách Hàng nói chung, không chấp nhận sai sót.

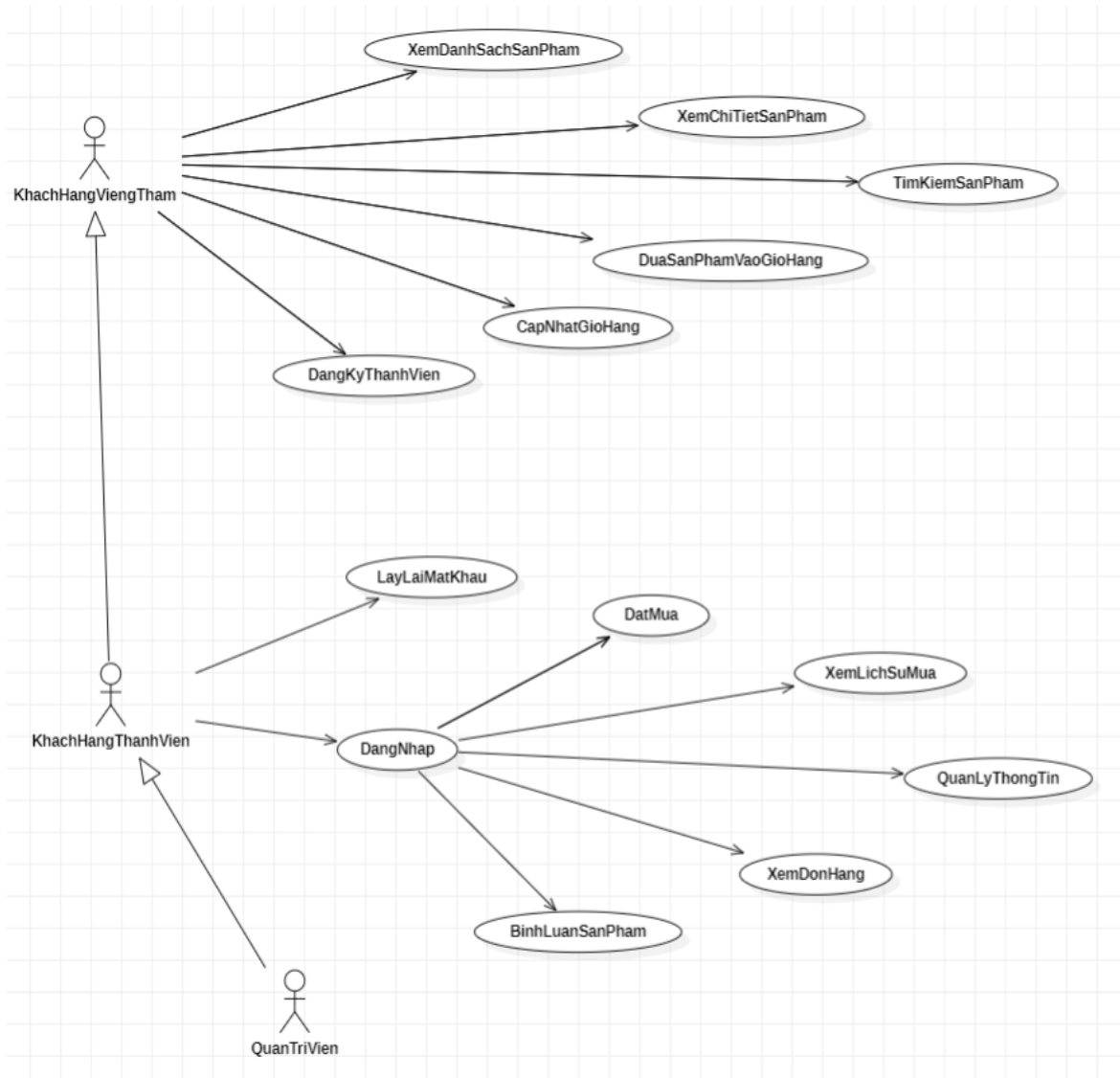
- Cơ sở dữ liệu phải hoạt động tốt khi hệ thống đang làm việc.

- Yêu cầu các bước chuyển tiếp phải rõ ràng, rành mạch, không rắc rối.

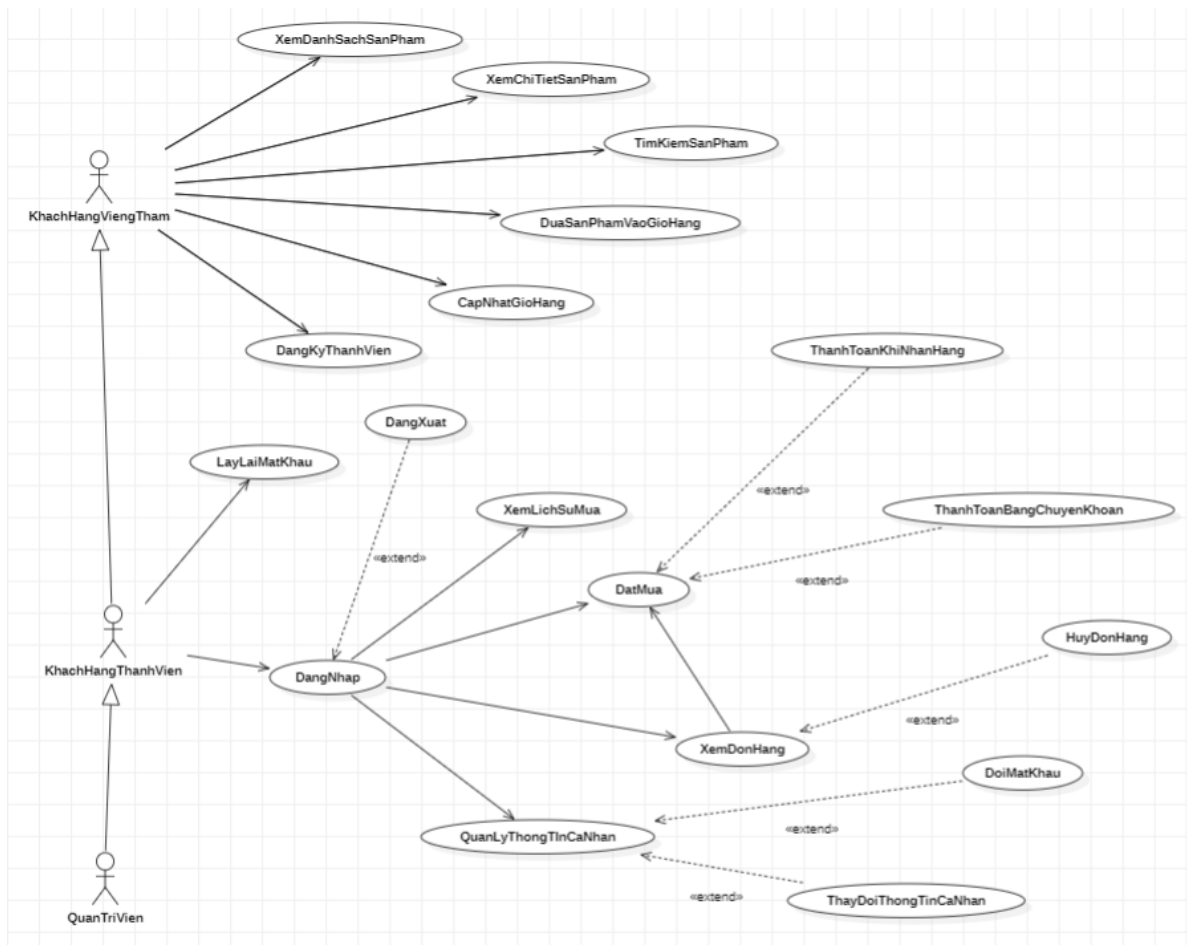
- Môi trường, hình ảnh, màu sắc ánh sáng thân thiện, dễ chịu đối với Khách Hàng và cả Quản Trị Viên.

5. CÁC SƠ ĐỒ

5.1 Use case diagram

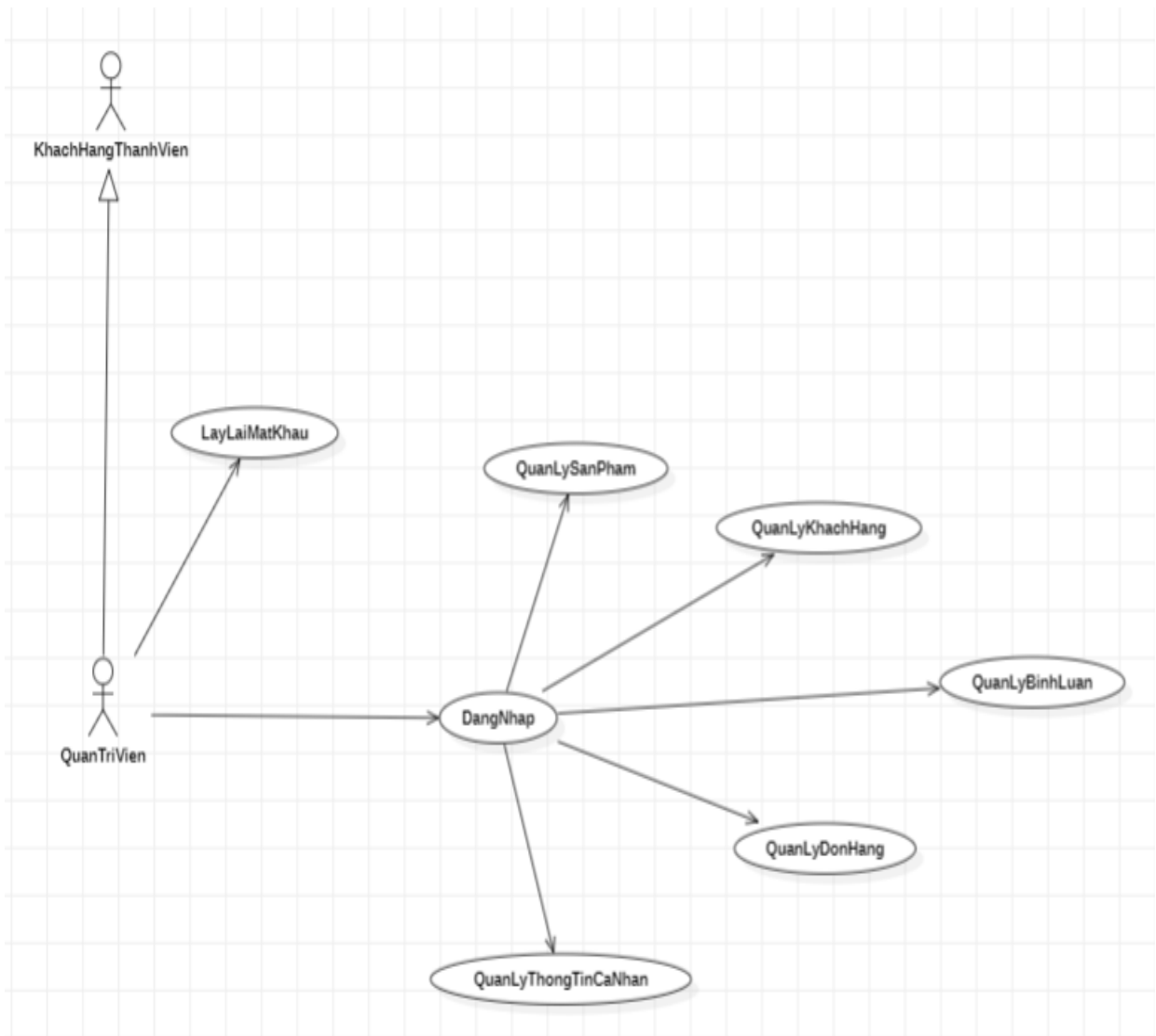


Sơ đồ 1: Sơ đồ Use Case mức 0 của khách hàng

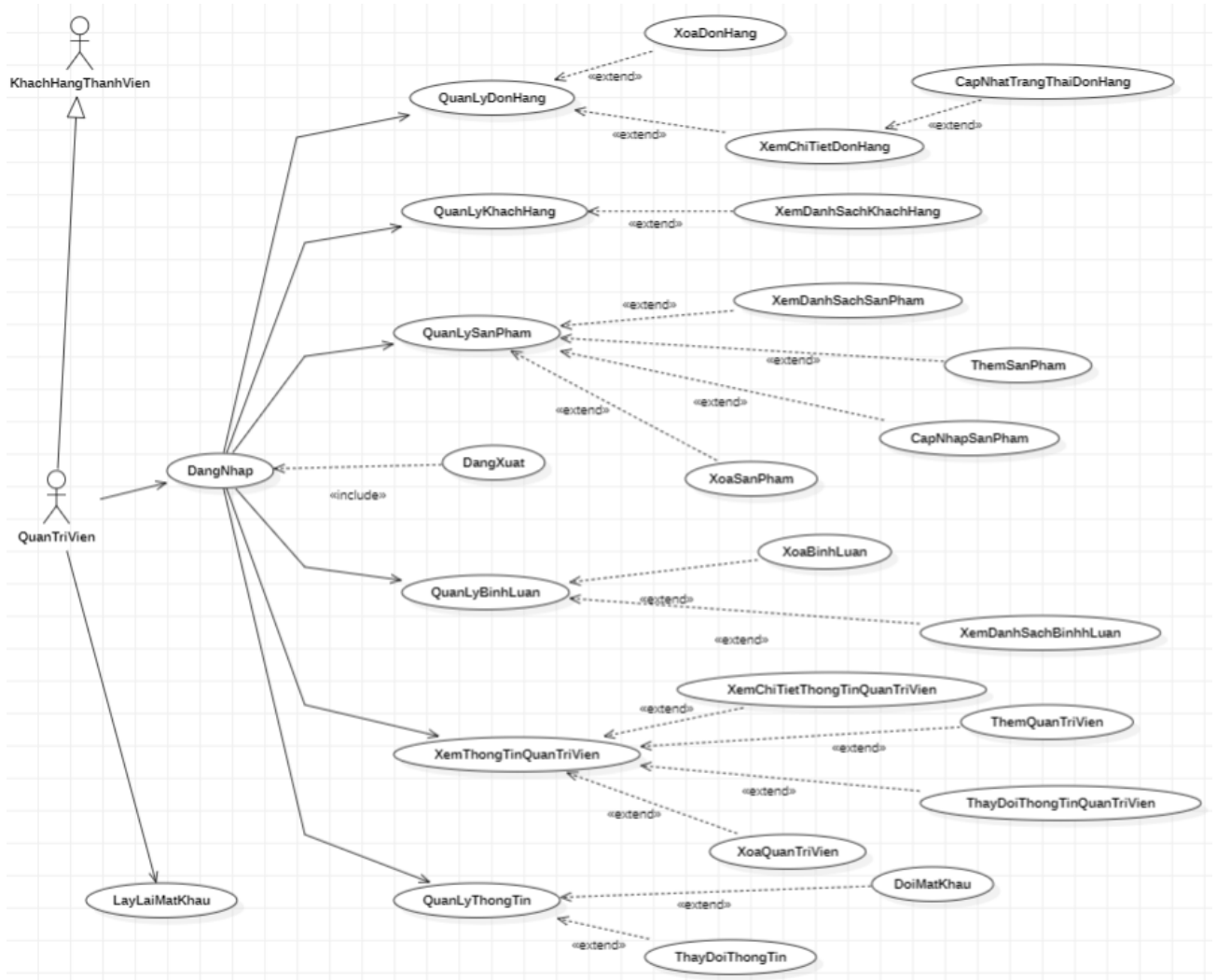


Sơ đồ 2: Sơ đồ Use Case mức 1 của khách hàng

- Người dùng khách (khách hàng không có tài khoản) có thể thực hiện các chức năng sau: Xem danh sách các sản phẩm, xem chi tiết các sản phẩm, tìm kiếm sản phẩm, đưa sản phẩm vào giỏ hàng, cập nhật giỏ hàng và đăng ký thành viên.
- Người dùng thành viên (khách hàng thành viên) có thể thực hiện các chức năng như sau: Lấy lại mật khẩu, đăng nhập, đăng ký, đăng xuất, xem lịch sử mua, mua hàng, thay đổi thông tin các nhân, hủy đơn hàng, xem đơn hàng, xem danh sách các sản phẩm, xem chi tiết các sản phẩm, tìm kiếm sản phẩm, đưa sản phẩm vào giỏ hàng, cập nhật giỏ hàng.



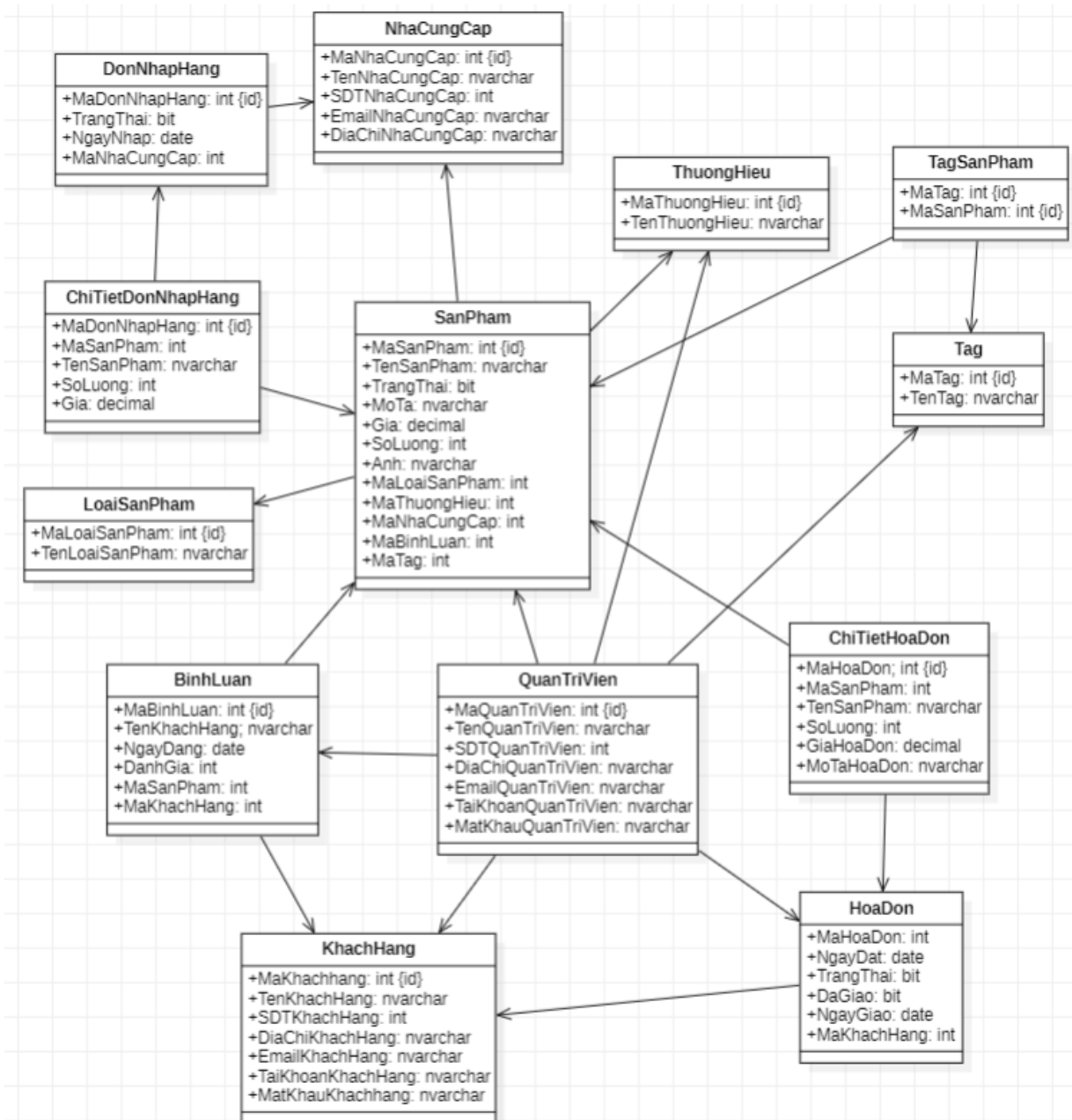
Sơ đồ 3: Sơ đồ mức 0 của quản trị viên



Sơ đồ 4: Sơ đồ mức 1 của quản trị viên

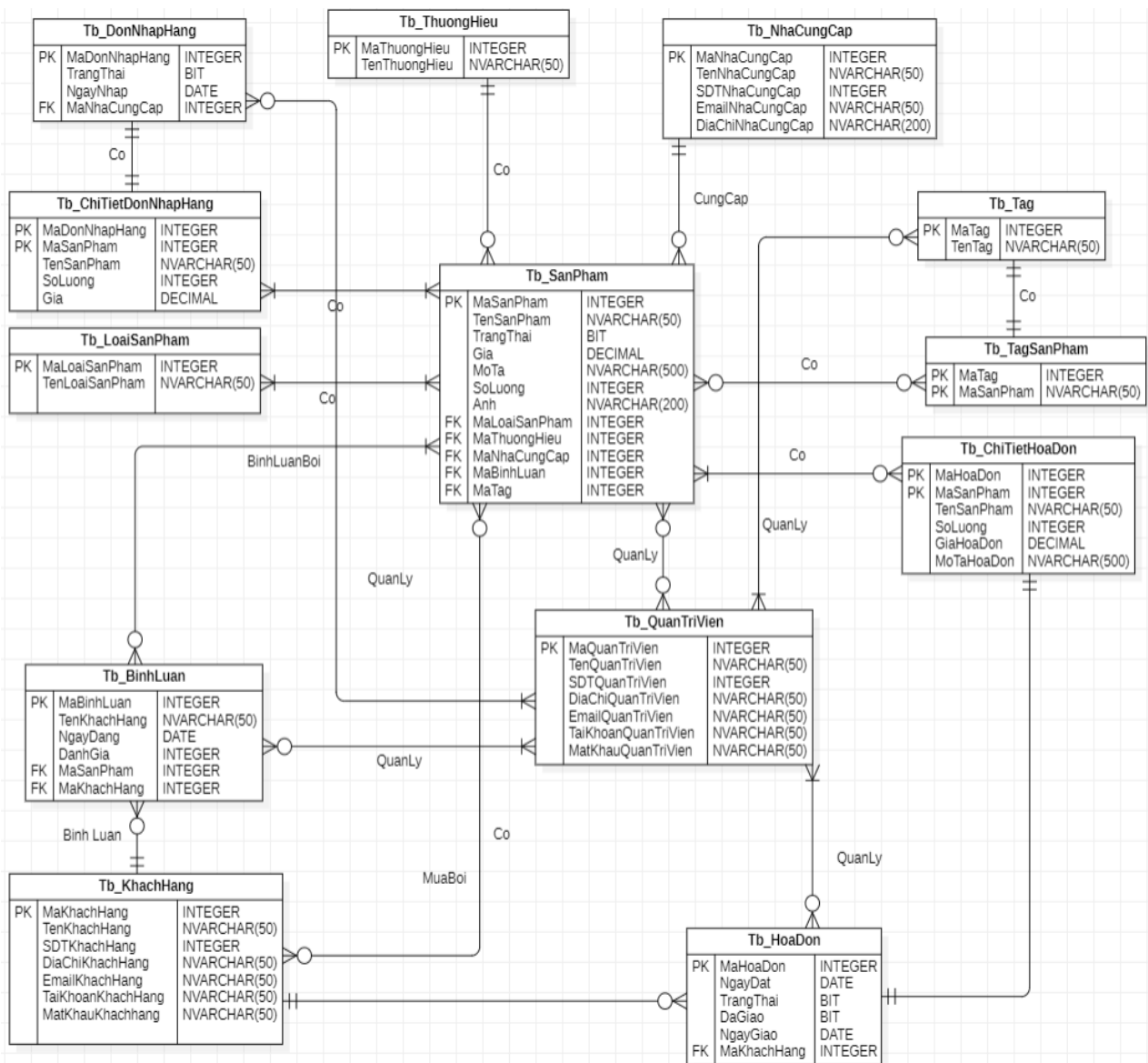
- Quản trị viên có thể thực hiện các chức năng sau: Lấy lại mật khẩu, đăng nhập, đăng xuất, quản lý đơn hàng (xóa đơn hàng, xem chi tiết đơn hàng, cập nhật đơn hàng), quản lý khách hàng (xem thông tin các khách hàng), quản lý sản phẩm (xem danh sách các sản phẩm, thêm sản phẩm, cập nhật sản phẩm, xóa sản phẩm), quản lý các bình luận (xem các bình luận, xóa bình luận), quản lý các quản trị viên (xem thông tin chi tiết, ngưng cấp quyền cho các quản trị viên, thay đổi thông tin, thêm quản trị viên mới), quản lý thông tin cá nhân (thay đổi thông tin cá nhân, đổi mật khẩu).

5.2 Entity Diagram (sơ đồ thực thể)



Sơ đồ 5: Sơ đồ thực thể biểu hiện các thành phần của ứng dụng

5.3 Relationship Diagram (sơ đồ quan hệ)



Sơ đồ 6: Sơ đồ quan hệ biểu thị các mối liên hệ của các thành phần trong ứng dụng

Danh sách các bảng thực thể:

STT	Tên bảng	Diễn giải
1	Tb_DonNhapHang	Đơn nhập hàng
2	Tb_ThuongHieu	Thương hiệu
3	Tb_NhaCungCap	Nhà cung cấp
4	Tb_ChiTietDonNhapHang	Chi tiết đơn nhập hàng
5	Tb_LoaiSanPham	Loại sản phẩm
6	Tb_SanPham	Sản phẩm
7	Tb_Tag	Các loại thẻ
8	Tb_TagSanPham	Các thẻ trên sản phẩm
9	Tb_ChiTietHoaDon	Chi tiết hóa đơn
10	Tb_QuanTriVien	Quản trị viên
11	Tb_BinhLuan	Bình luận
12	Tb_KhachHang	Khách hàng
13	Tb_HoaDon	Hóa đơn

Bảng 5: Bảng liệt kê danh sách các thực thể

Bảng đơn nhập hàng: dùng để lưu thông tin và cập nhật thông tin số lượng từng loại sản phẩm.

ĐƠN NHẬP HÀNG				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaDonNhapHang	INTEGER	Khóa chính	Mã đơn nhập hàng
2	TrangThai	BIT		Trạng thái của đơn hàng
3	NgayNhap	DATE		Ngày nhập hàng
4	MaNhaCungCap	INTEGER	Khóa ngoại	Mã nhà cung cấp

Bảng 6: Bảng đơn nhập hàng

- Đơn Nhập Hàng quan hệ một nhiều với Quản Trị Viên
- Mỗi Đơn Nhập Hàng có một Chi Tiết Đơn Nhập Hàng dùng để liệt kê chi tiết thông tin của Đơn Nhập Hàng đó

Bảng thương hiệu: dùng để quản lý các thương hiệu cho từng sản phẩm.

THƯƠNG HIỆU				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaThuongHieu	INTEGER	Khóa chính	Mã thương hiệu
2	TenThuongHieu	NVARCHAR (50)		Tên thương hiệu

Bảng 7: Bảng thương hiệu

- Thương Hiệu có quan hệ một nhiều với Sản Phẩm

Bảng nhà cung cấp: dùng để quản lý các nhà cung cấp nhập hàng cho từng sản phẩm.

NHÀ CUNG CẤP				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaNhaCungCap	INTEGER	Khóa chính	Mã nhà cung cấp
2	TenNhaCungCap	NVARCHAR (50)		Tên nhà cung cấp
3	SDTNhaCungCap	INTEGER		Số điện thoại nhà cung cấp
4	EmailNhaCungCap	NVARCHAR (50)		Địa chỉ email nhà cung cấp
5	DiaChiNhaCungCap	NVARCHAR (200)		Địa chỉ nhà cung cấp

Bảng 8 : Bảng nhà cung cấp

- Nhà Cung Cấp có thể có quan hệ một nhiều với Sản Phẩm

Bảng chi tiết đơn nhập hàng: hiển thị chi tiết thông tin đơn nhập hàng của các sản phẩm.

CHI TIẾT ĐƠN NHẬP HÀNG				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaDonNhapHang	INTEGER	Khóa chính	Mã đơn nhập hàng
2	MaSanPham	INTEGER	Khóa chính	Mã sản phẩm
3	TenSanPham	NVARCHAR (50)		Tên sản phẩm
4	SoLuong	INTEGER		Số lượng sản phẩm
5	Gia	DECIMAL		Giá của tổng hóa đơn

Bảng 9 : Bảng đơn nhập hàng

- Mỗi Chi Tiết Đơn Nhập Hàng thuộc một Đơn Nhập Hàng duy nhất dùng để miêu tả các thông tin khác chi tiết hơn cho Đơn Nhập Hàng
- Một Chi Tiết Đơn Nhập Hàng có ít nhất một Sản Phẩm trong đó, nếu không có Sản Phẩm sẽ không tạo thành một Đơn Hàng

Bảng loại sản phẩm: quản lý các loại sản phẩm của sản phẩm.

LOẠI SẢN PHẨM				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaLoaiSanPham	INTEGER	Khóa chính	Mã loại sản phẩm
2	TenLoaiSanPham	NVARCHAR (50)		Tên loại sản phẩm

Bảng 10 : Bảng loại sản phẩm

- Mỗi Loại Sản Phẩm có ít nhất trong một Sản Phẩm hay trong nhiều Sản Phẩm

Bảng sản phẩm: quản lý tất cả các sản phẩm của cửa hàng.

SẢN PHẨM				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaSanPham	INTEGER	Khóa chính	Mã sản phẩm
2	TenSanPham	NVARCHAR (50)		Tên sản phẩm
3	TrangThai	BIT		Trạng thái của sản phẩm
4	Gia	NVARCHAR (50)		Địa chỉ email nhà cung cấp
5	MoTa	NVARCHAR (200)		Địa chỉ nhà cung cấp
6	Anh	NVARCHAR (50)		Ảnh của sản phẩm
7	MaLoaiSanPham	INTEGER	Khóa ngoại	Mã loại sản phẩm
8	MaThuongHieu	INTEGER	Khóa ngoại	Mã thương hiệu
9	MaNhaCungCap	INTEGER	Khóa ngoại	Mã nhà cung cấp
10	MaBinhLuan	INTEGER	Khóa ngoại	Mã bình luận
11	MaTag	INTEGER	Khóa ngoại	Mã tag
12	SoLuong	INTEGER		Số lượng tồn

Bảng 11 : Bảng sản phẩm

- Một Sản Phẩm sẽ có quan hệ một – một với Thương Hiệu và quan hệ một – một với Nhà Cung Cấp
- Một Sản Phẩm có thể không có các Tag hoặc có nhiều Tag

- Mỗi Sản Phẩm có thể không có trong Chi Tiết Hóa Đơn nào hoặc có trong nhiều Chi Tiết Hóa Đơn
- Sản Phẩm có thể không được quản lý bởi một Quản Trị Viên hoặc nhiều hơn một Quản Trị Viên
- Một Sản Phẩm không bị Khách Hàng nào mua hoặc đã bị mua bởi nhiều Khách Hàng
- Một Sản Phẩm không được bình luận bởi Khách Hàng nào hoặc đã được bình luận bởi nhiều Khách Hàng
- Một Sản Phẩm có ít nhất một Loại Sản Phẩm và có ít nhất một Sản Phẩm có trong Chi Tiết Đơn Nhập Hàng

Bảng tag: quản lý các thẻ thông tin thẻ và thêm, xóa hay sửa thẻ.

TAG				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaTag	INTEGER	Khóa chính	Mã thẻ
2	TenTag	NVARCHAR (50)		Tên thẻ

Bảng 12 : Bảng các thẻ

- Mỗi Tag sẽ không có trong Sản Phẩm nào hoặc có ít nhất trong một Sản Phẩm và được quản lý bởi ít nhất một Quản Trị Viên

Bảng tag sản phẩm: quản lý các thẻ thông tin thẻ thuộc trong các sản phẩm

TAG SẢN PHẨM				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaTag	INTEGER	Khóa chính	Mã thẻ
2	MaSanPham	INTEGER	Khóa chính	Mã sản phẩm

Bảng 13 : Bảng các thẻ thuộc sản phẩm

- Mỗi Tag sẽ không có trong Sản Phẩm nào hoặc có ít nhất trong một Sản Phẩm

Bảng chi tiết hóa đơn: dùng để hiển thị thông tin chi tiết của một hóa đơn.

CHI TIẾT HÓA ĐƠN				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaHoaDon	INTEGER	Khóa chính	Mã hóa đơn
2	MaSanPham	INTEGER	Khóa chính	Mã sản phẩm
3	TenSanPham	NVARCHAR (50)		TenSanPham
4	Soluong	INT		Số lượng
5	GiaHoaDon	DECIMAL		Giá của hóa đơn
6	MoTa	NVARCHAR (500)		Mô tả về hóa đơn

Bảng 14 : Bảng chi tiết hóa đơn

- Chi Tiết Hóa Đơn sẽ có ít nhất một Sản Phẩm hay nhiều Sản Phẩm và chỉ thuộc duy nhất trong một Hóa Đơn

Bảng quản trị viên: dùng để lưu thông tin và quản lý thông tin quản trị viên.

QUẢN TRỊ VIÊN				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaQuanTriVien	INTEGER	Khóa chính	Mã quản trị viên
2	TenQuanTriVien	NVARCHAR (50)		Tên quản trị viên
3	SDTQuanTriVien	INTEGER		Số điện thoại quản trị viên
4	DiaChiQuanTriVien	NVARCHAR (50)		Địa chỉ quản trị viên
5	EmailQuanTriVien	NVARCHAR (50)		Email quản trị viên
6	TaiKhoanQuanTriVien	NVARCHAR (500)		Tài khoản quản trị viên
7	MatKhauQuanTriVien	NVARCHAR (50)		Mật khẩu quản trị viên

Bảng 15 : Bảng quản trị viên

- Mỗi Quản Trị Viên có thể không quản lý bất kỳ Sản Phẩm nào hay quản lý nhiều Sản Phẩm
- Mỗi Quản Trị Viên quản lý một hoặc nhiều Tag hoặc không quản lý Tag nào
- Quản Trị Viên quản lý nhiều Hóa Đơn cùng lúc hoặc có thể không quản lý Hóa Đơn nào
- Quản Trị Viên quản lý nhiều các Bình Luận của Khách Hàng hoặc không quản lý Bình Luận nào
- Quản Trị Viên quản lý không hoặc nhiều Đơn Nhập Hàng

Bảng bình luận: dùng để quản lý các bình luận của các khách hàng dành cho từng sản phẩm.

BÌNH LUẬN				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaBinhLuan	INTEGER	Khóa chính	Mã bình luận
2	TenKhachHang	NVARCHAR (50)		Tên khách hàng
3	NgayDang	DATE		Ngày đăng
4	DanhGia	INTEGER		Đánh giá
5	MaSanPham	INTEGER	Khóa ngoại	Mã sản phẩm
6	MaKhachHang	INTEGER	Khóa ngoại	Mã khách hàng

Bảng 16: Bảng bình luận

- Một Bình Luận được bình luận bởi một Khách Hàng hoặc không có Khách Hàng nào bình luận về Sản Phẩm đó cả
- Bình Luận được quản lý bởi ít nhất một Quản Trị Viên hoặc nhiều Quản Trị Viên quản lý Bình Luận đó

Bảng khách hàng: dùng để lưu thông tin và quản lý thông tin của khách hàng.

KHÁCH HÀNG				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaKhachHang	INTEGER	Khóa chính	Mã khách hàng
2	TenKhachHang	NVARCHAR (50)		Tên khách hàng
3	SDTKhachHang	INTEGER		Số điện thoại khách hàng
4	DiaChiKhachHang	NVARCHAR (50)		Địa chỉ khách hàng
5	EmailKhachHang	NVARCHAR (50)		Email khách hàng
6	TaiKhoanKhachHang	NVARCHAR (500)		Tài khoản khách hàng
7	MatKauKhachHang	NVARCHAR (50)		Mật khẩu khách hàng

Bảng 17 : Bảng khách hàng

- Khách Hàng có thể không bình luận Sản Phẩm nào hoặc bình luận nhiều Sản Phẩm cùng lúc
- Mỗi Khách Hàng có thể không mua Sản Phẩm nào hoặc mua bởi nhiều Sản Phẩm cùng lúc

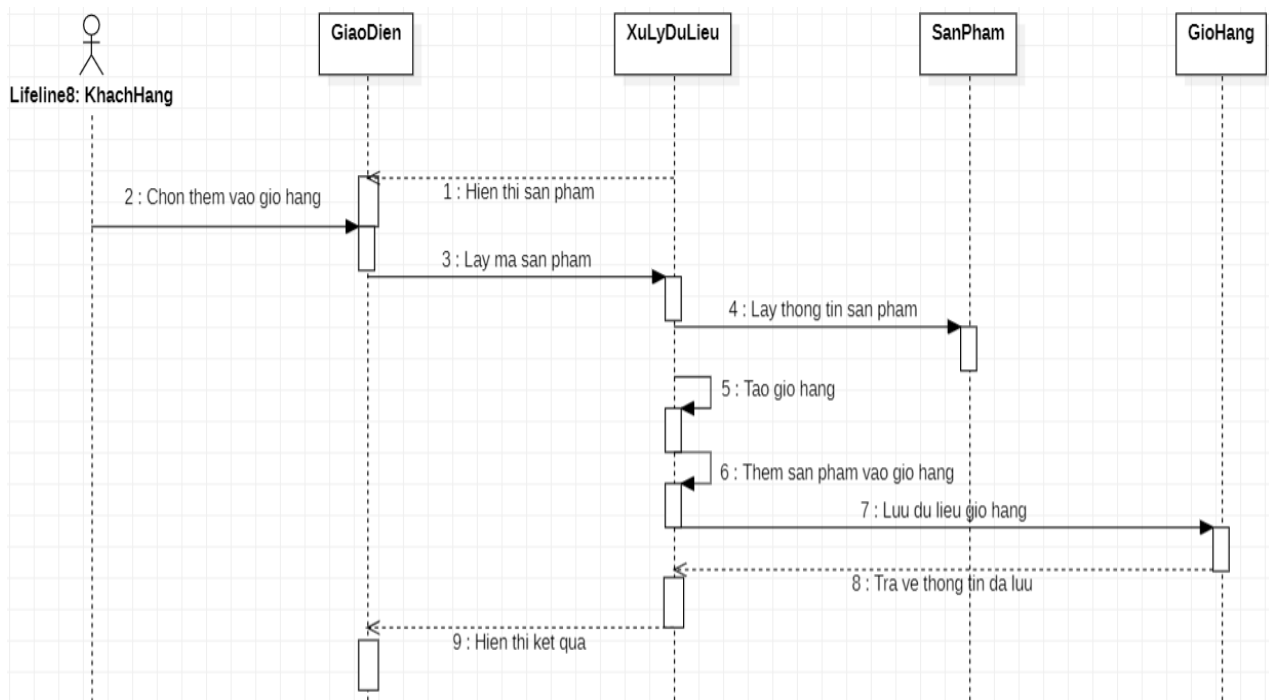
Bảng hóa đơn: dùng để lưu lại thông tin về các hóa đơn mà những khách hàng đã đặt.

HÓA ĐƠN				
STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaHoaDon	INTEGER	Khóa chính	Mã hóa đơn
2	NgayDat	DATE		Ngày đặt
3	TrangThai	BIT		Trạng thái của hóa đơn
4	DaGiao	BIT		Đã giao hay chưa
5	NgayGiao	DATE		Ngày giao
6	MaKhachHang	INTEGER	Khóa ngoại	Mã khách hàng đã đặt

Bảng 18: Bảng hóa đơn

- Một Hóa Đơn chỉ có duy nhất một Khách Hàng và có duy nhất một Chi Tiết Hóa Đơn miêu tả về Hóa Đơn đó
- Một Hóa Đơn được quản lý một Quản Trị Viên hoặc không bởi một Quản Trị Viên nào cả

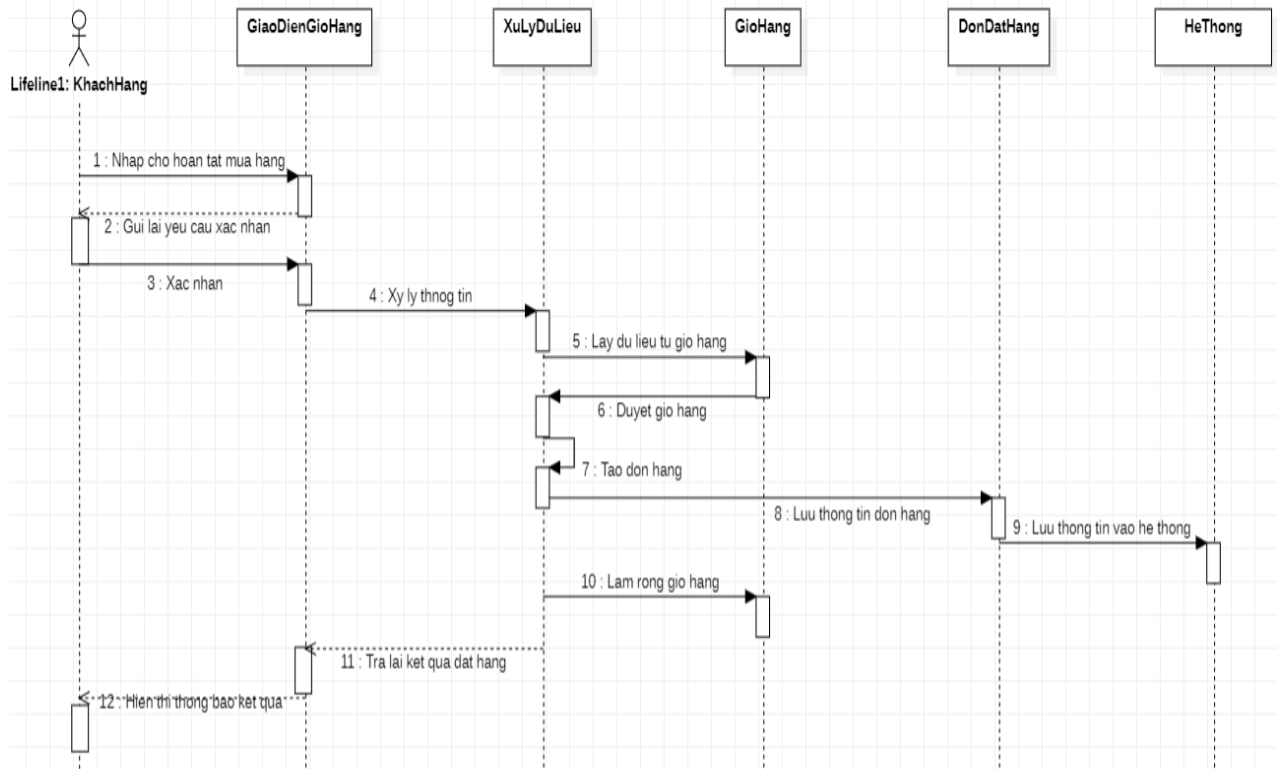
5.4 Sequence diagram (Sơ đồ tuần tự)



Sơ đồ 7: Sơ đồ tuần tự thêm sản phẩm của khách hàng

Miêu tả chi tiết các trình tự thêm sản phẩm vào giỏ hàng:

- Sau khi trang web xử lý dữ liệu sẽ hiện ra giao diện sản phẩm.
- Khách hàng có thể lựa chọn tùy ý các sản phẩm yêu thích và thêm vào giỏ hàng.
- Web sẽ lấy id của sản phẩm và tìm sản phẩm đó theo id và lưu nó vào LocalStorage tạm thời.
- Tạo giỏ hàng mới và lấy dữ liệu đã lưu trong LocalStorage ra giỏ hàng.
- Sau khi khách hàng hoàn tất quá trình mua hàng, click vào giỏ hàng hàng sẽ nhận được các sản phẩm đã chọn từ trước.



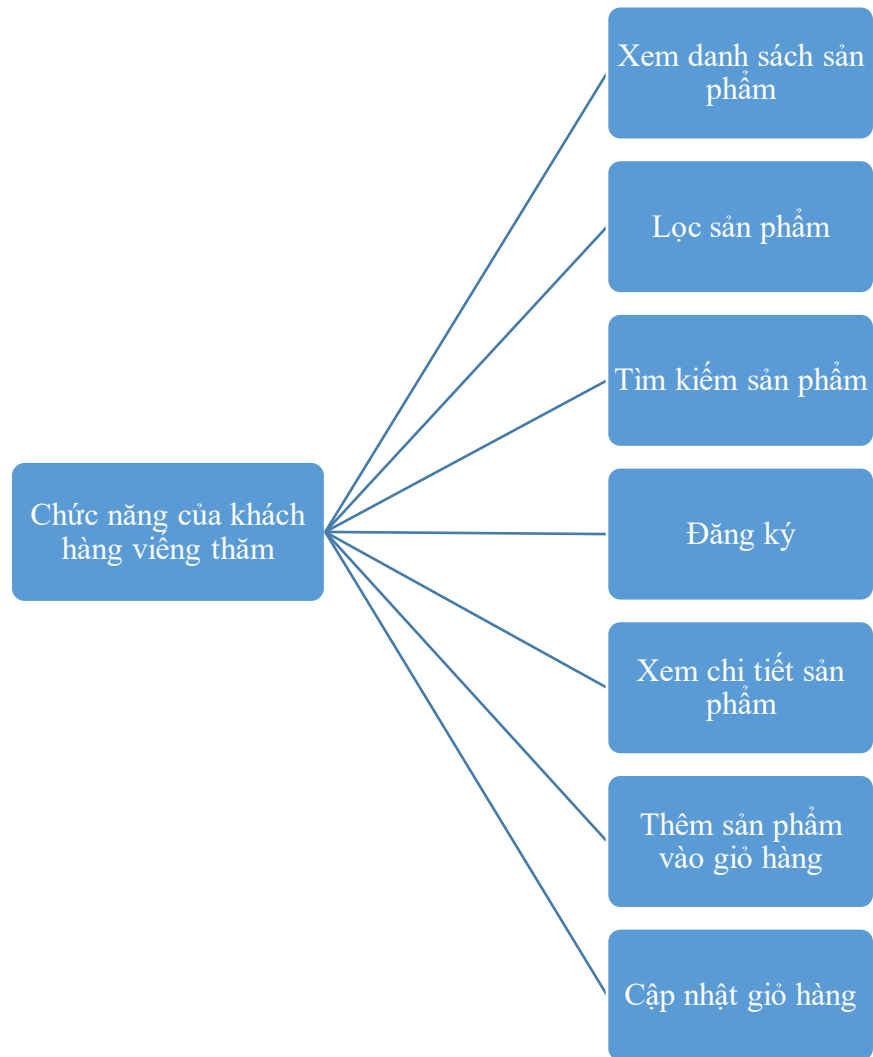
Sơ đồ 8: Sơ đồ tuần tự mua hàng của khách hàng

Miêu tả chi tiết trình tự mua hàng của khách hàng

- Sau quá trình chọn lựa sản phẩm khách hàng chọn Đặt Mua
- Trang web sẽ gửi một yêu cầu xác nhận có đồng ý mua hàng hay không, chọn đồng ý xác nhận
- Lúc này trang web sẽ xử lý thông tin lấy dữ liệu các sản phẩm và số lượng
- Tiếp theo sẽ tạo đơn hàng và lưu vào hệ thống với thông tin gồm: tất cả sản phẩm trong giỏ hàng và số lượng cho các sản phẩm đó, thông tin khách hàng mua, tổng tiền của hóa đơn và thời gian đặt
- Hệ thống sẽ làm rỗng lại giỏ hàng (tức làm rỗng lại Local Storage) sau khi lưu thông tin đơn hàng thành công
- Trả lại kết quả đơn hàng khách hàng đã đặt mua gồm các thông tin như: các sản phẩm đã đặt, số lượng, tổng tiền của đơn hàng, ngày đặt và trạng thái của đơn hàng (được xác nhận hay chưa)

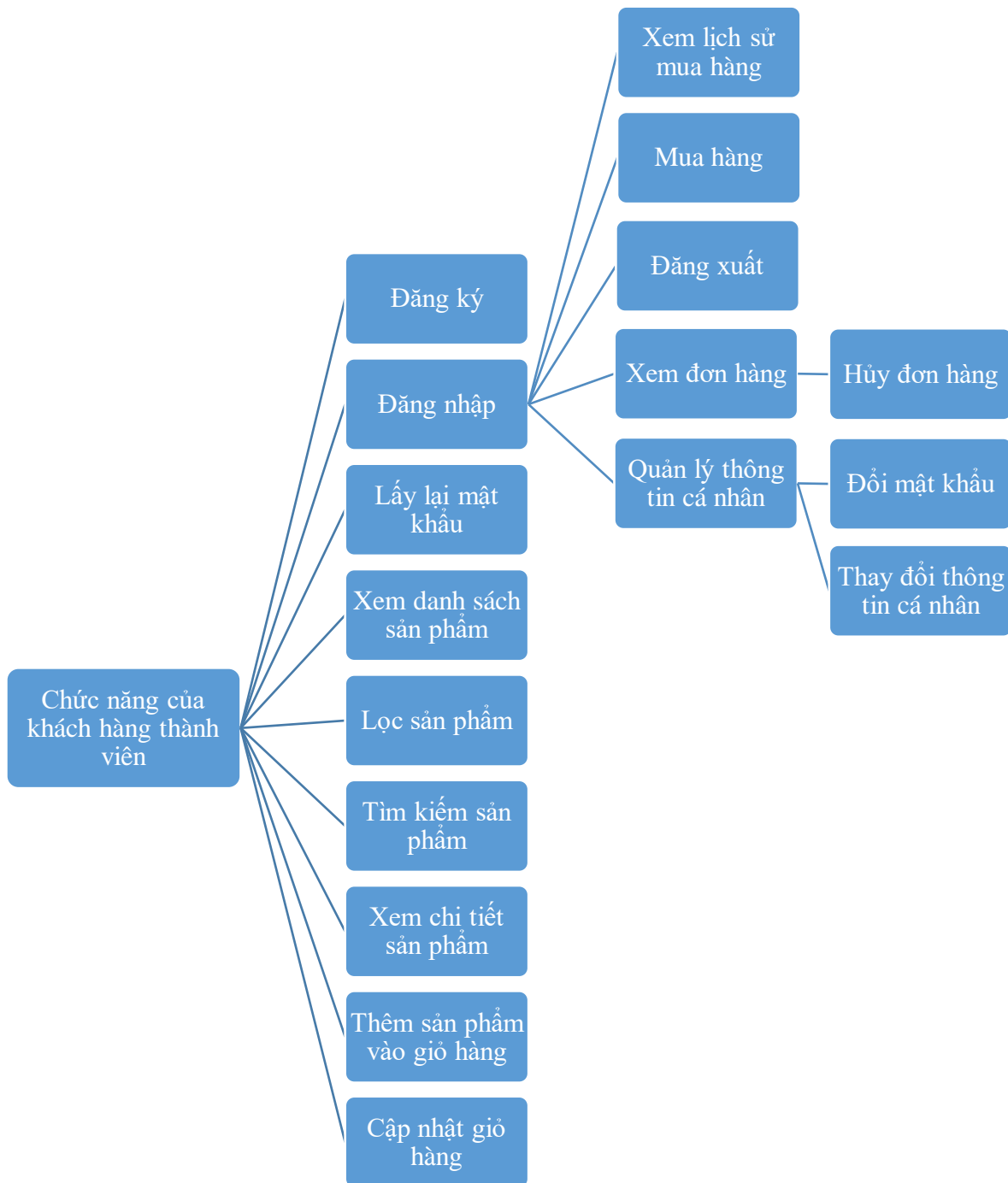
5.5 Functional diagram (Sơ đồ chức năng)

Sơ đồ chức năng của khách hàng viếng thăm:



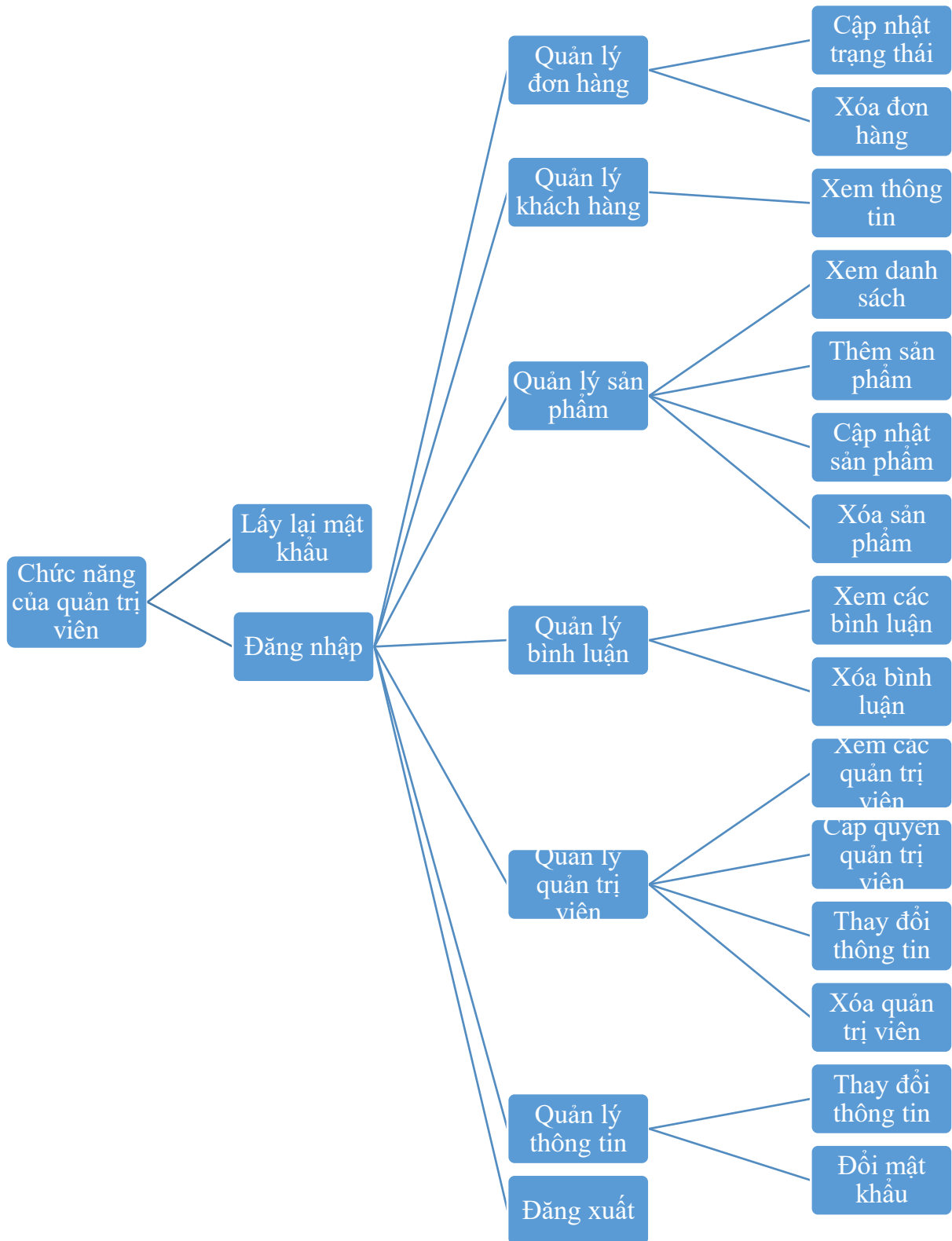
Sơ đồ 9: Sơ đồ chức năng của khách hàng viếng thăm

Sơ đồ chức năng của khách hàng thành viên:



Sơ đồ 10: Sơ đồ chức năng của khách hàng thành viên

Sơ đồ chức năng của quản trị viên:



Sơ đồ 11: Sơ đồ chức năng của quản trị viên


6. KIỂM THỬ

Kiểm thử quá trình mua hàng của khách hàng thành viên:

URL: <http://localhost:3000/>

P&P SHOP


[ĐĂNG NHẬP](#) [ĐĂNG KÝ](#)



SUMMER SALE


Đừng chạy theo xu hướng. Đừng khiến bản thân lệ thuộc vào thời trang. Hãy để chính mình là người quyết định bản thân sẽ mặc gì cũng như sẽ sống ra sao.

[MUA NGAY](#)




SHIRT STYLE!

[MUA NGAY](#)



JEAN LOVE

[MUA NGAY](#)



LIGHT JACKETS

[MUA NGAY](#)

P&P

Đến với P&P bạn sẽ có một trải nghiệm tuyệt vời về thời trang, tận hưởng không khí gian thoải mái với những sản phẩm mới được cập nhật liên tục và mới mẻ với giá cả phải chăng.

[f](#) [i](#) [t](#) [p](#)


Useful Links

[Home](#)
[Men Fashion](#)
[Accessories](#)
[Order Tracking](#)
[Wishlist](#)

Contact

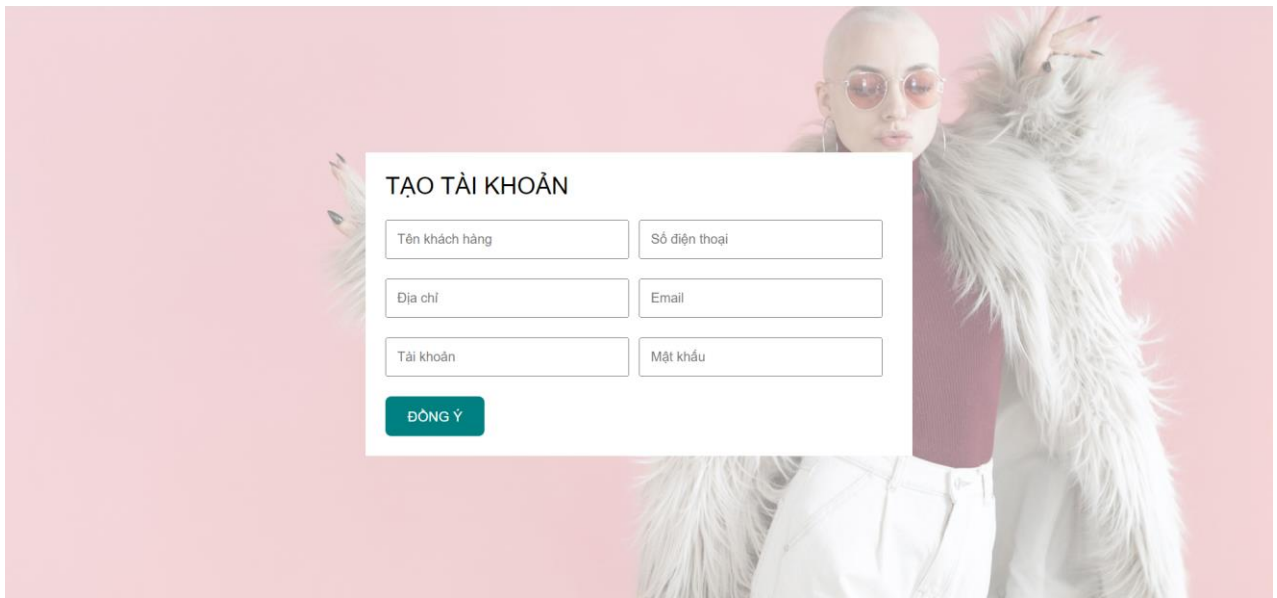
[Chat](#)
[Woman Fashion](#)
[My Account](#)
[Wishlist](#)
[Terms](#)

150 Huỳnh Cương, An Cư Ward, Ninh Kiều District, Cần Thơ City
0907 113 113
P&P@gmail.com



- Giao diện là trang Landing Page với mục đích sơ lược về thông tin của Website
- Click **ĐĂNG KÝ** góc trên bên phải màn hình để tiến hành đăng ký tài khoản

URL: <http://localhost:3000/register>

A registration form titled "TẠO TÀI KHOẢN" (Create Account) is displayed over a background image of a person with a shaved head wearing sunglasses and a white fur coat. The form contains six input fields: "Tên khách hàng" (Customer Name), "Số điện thoại" (Phone Number), "Địa chỉ" (Address), "Email", "Tài khoản" (Username), and "Mật khẩu" (Password). A green "ĐỒNG Ý" (I Agree) button is located at the bottom left of the form.

TẠO TÀI KHOẢN

Tên khách hàng

Số điện thoại

Địa chỉ

Email

Tài khoản

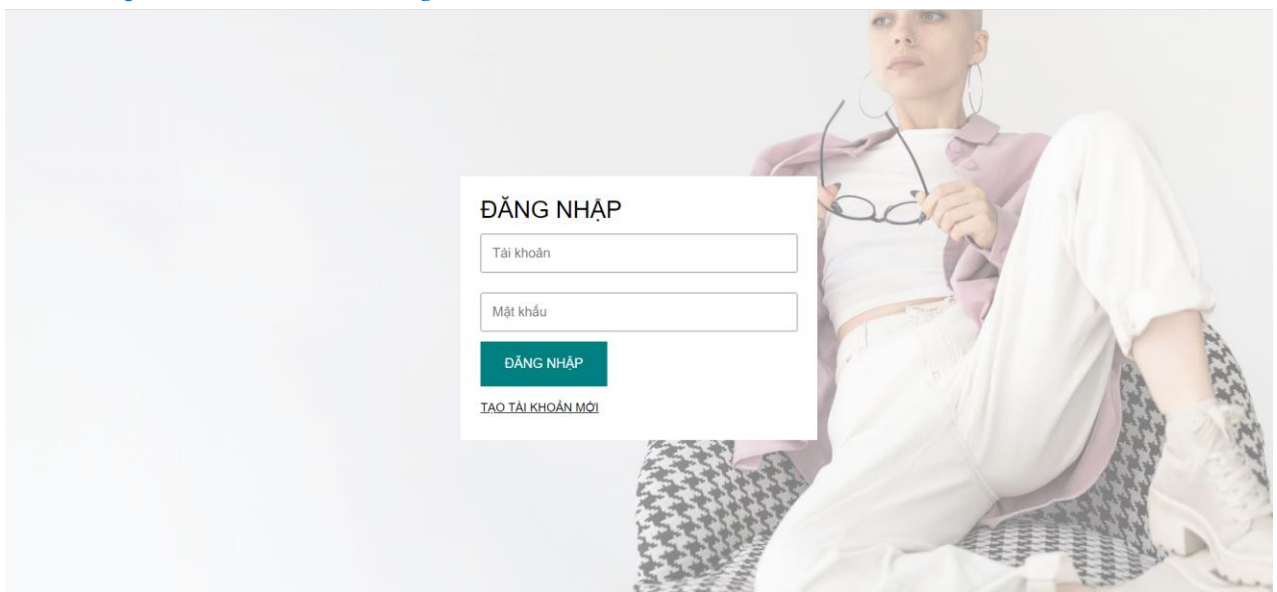
Mật khẩu

ĐỒNG Ý

- Giao diện để đăng ký thành viên của Website
- Sau khi điền chính xác các thông tin sẽ được chuyển đến trang đăng nhập thông tin vừa đăng ký

Lưu ý: Điền đầy đủ thông tin và chính xác nếu không sẽ hiện thông báo lỗi

URL: <http://localhost:3000/login>

A login form titled "ĐĂNG NHẬP" (Login) is displayed over a background image of a person with a shaved head wearing a purple jacket and white pants. The form contains two input fields: "Tài khoản" (Username) and "Mật khẩu" (Password). A green "ĐĂNG NHẬP" (Login) button is located below the fields. A link "TẠO TÀI KHOẢN MỚI" (Create New Account) is located at the bottom of the form.

ĐĂNG NHẬP

Tài khoản

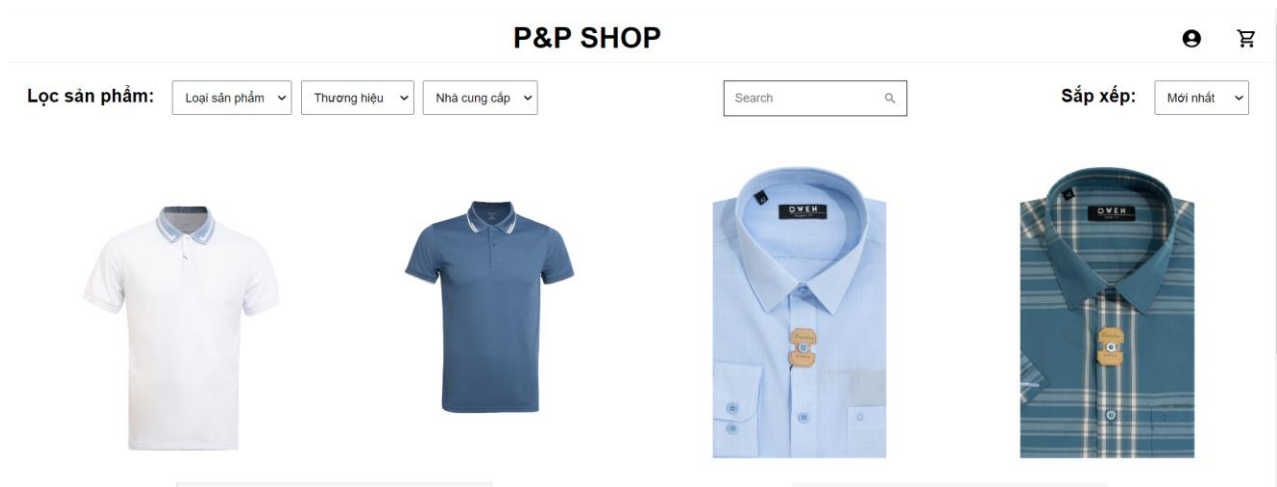
Mật khẩu


ĐĂNG NHẬP

[TẠO TÀI KHOẢN MỚI](#)

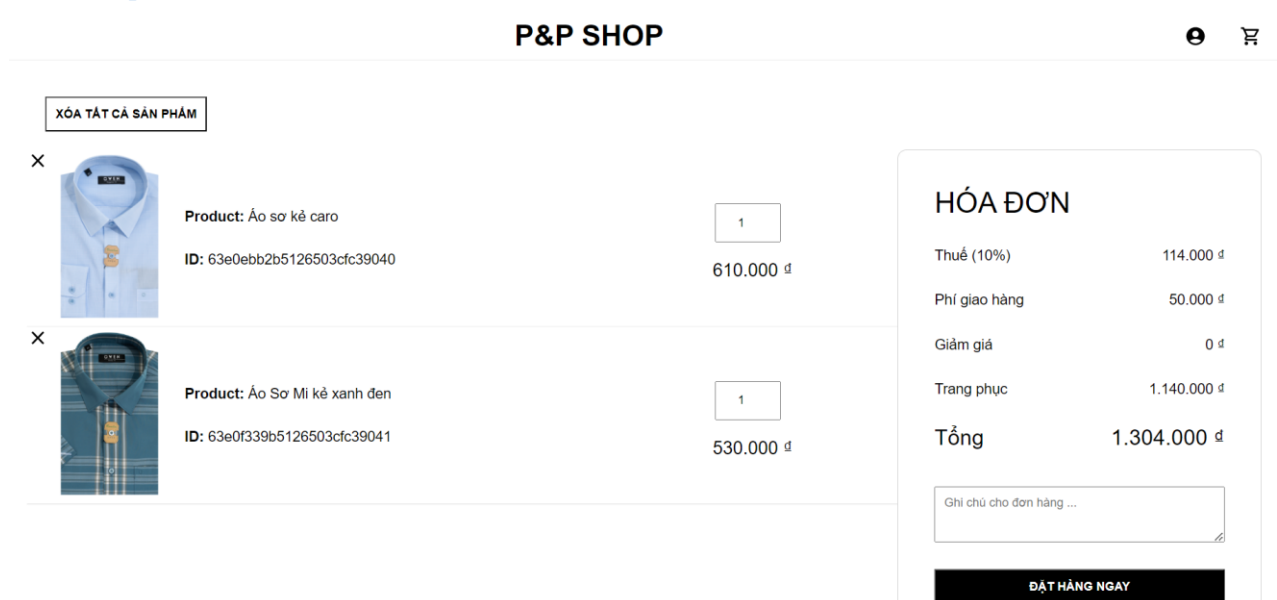
- Điền thông tin vừa đăng ký vào và đã thành công trở thành khách hàng thành viên của cửa hàng

URL: <http://localhost:3000/productlist>



- Tại trang danh sách các sản phẩm chọn sản phẩm để mua hàng
- Sau khi chọn xong click  để xem giỏ hàng

URL: <http://localhost:3000/cart>



- Tại đây có thể tùy chỉnh số lượng muốn mua sau đó click **Đặt Hàng Ngay**
- Từ đó có thể xem lịch sử mua hàng và trạng thái của đơn hàng trên góc phải màn hình

URL: <http://localhost:3000/historybought>

P&P SHOP				
Ngày mua	Tổng tiền	Trạng thái	Đã giao	Hành động
09/05/2023	1.304.000 đ	Nhận đơn	Chưa giao hàng	→ ☰
07/05/2023	4.395.000 đ	Nhận đơn	Chưa giao hàng	→ ☰

- Trạng thái **Nhận đơn** cho biết đơn hàng đã được chuẩn bị hay chưa và trạng thái **Chưa giao hàng** cho biết rằng đơn hàng đã được giao hay chưa.

Kiểm thử quá trình quản lý đơn hàng của các quản trị viên:

- Đầu tiên, để có thể quản lý được các đơn hàng thì cần được quyền admin phân quyền sau khi đăng ký một tài khoản thành viên
- Sau đó đăng nhập sẽ hiển thị giao diện của các quản trị viên, tùy thuộc vào quyền gì mà giao diện quản lý sẽ khác nhau
- Click chọn **Hóa Đơn** trên thanh **Menu** để đến với các sản phẩm mà khách hàng đã đặt mua

URL: <http://localhost:3000/bill>

P&P SHOP

☰

Thông tin cá nhân

>

☰

Thư viện ảnh

>

☰

Sản Phẩm

🔍

Hóa Đơn

📦

Đơn Nhập Hàng

💬

Bình Luận

💳

Thẻ

🏆

Thương Hiệu

📦

Loại Sản Phẩm

🏠

Nhà cung cấp

👤

Khách Hàng

👤

Quản Trị Viên

📊

Thống kê doanh thu

Lọc sản phẩm:


Chưa giao hàng

▼

Ngày đặt	Tổng tiền	Trạng thái	Đã giao	Hành động
09/05/2023	1.304.000 đ	Nhận đơn	Chưa giao hàng	↔️ 🗑️
07/05/2023	4.395.000 đ	Nhận đơn	Chưa giao hàng	↔️ 🗑️
27/04/2023	3.724.000 đ	Nhận đơn	Chưa giao hàng	↔️ 🗑️
27/04/2023	1.304.000 đ	Nhận đơn	Chưa giao hàng	↔️ 🗑️
22/04/2023	633.000 đ	Nhận đơn	Chưa giao hàng	↔️ 🗑️
21/04/2023	1.304.000 đ	Nhận đơn	Chưa giao hàng	↔️ 🗑️
21/04/2023	1.304.000 đ	Nhận đơn	Chưa giao hàng	↔️ 🗑️
21/04/2023	2.558.000 đ	Nhận đơn	Chưa giao hàng	↔️ 🗑️
28/03/2023	721.000 đ	Đã nhận đơn	Chưa giao hàng	↔️ 🗑️

- Click chọn **Nhận đơn** nếu đang chuẩn bị hàng cho Khách Hàng để Khách Hàng biết mình đã nhận được đơn
- Sau khi chuẩn bị các Sản Phẩm đóng gói xong chuyển hàng đi giao, Khách hàng nhận được hàng sẽ click **Chưa giao hàng** để đổi trạng thái thành **Đã giao hàng**


URL: <http://localhost:3000/detailbill/645a029d3d0f2436626a54d2>



Tên sản phẩm: Áo sơ kẻ caro

1

610.000 đ



Tên sản phẩm: Áo Sơ Mì kẻ xanh đen

1

530.000 đ

HÓA ĐƠN

Khách hàng :Trà Sữa Nâu

Số điện thoại :0785378493

Email :trasuanau@gmail.com

Địa chỉ :180 Huỳnh Cương

Tổng1.304.000 đ

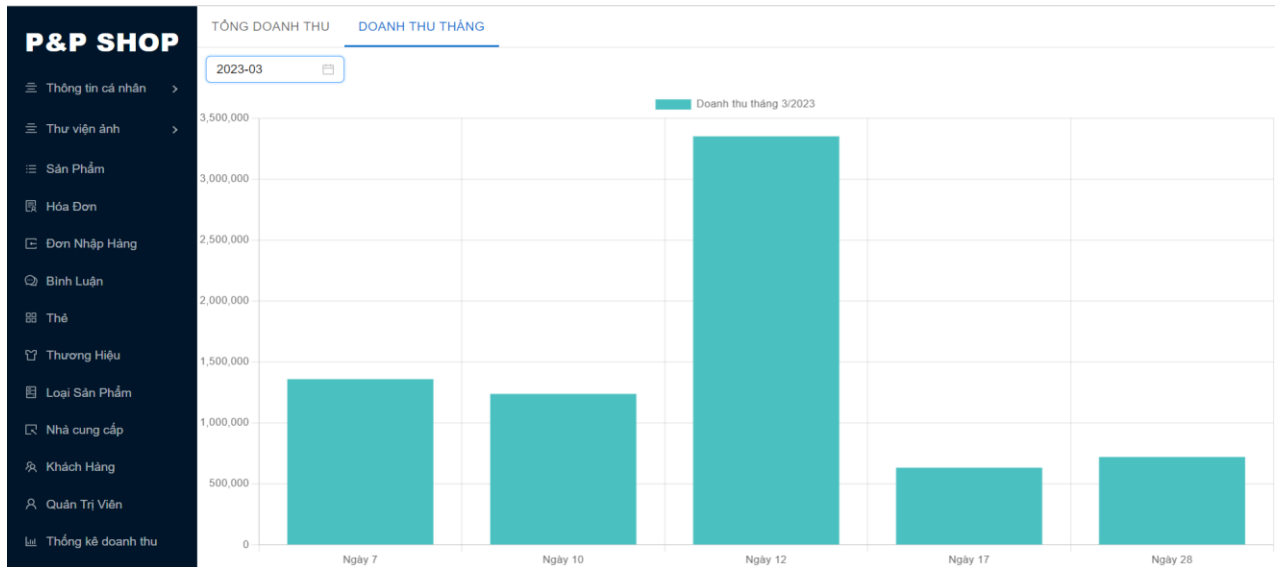
Ghi chú cho đơn hàng ...

- Click chọn ➡ để xem chi tiết hóa đơn gồm: sản phẩm, số lượng, thông tin khách hàng,...
- Tiếp theo, trở về trang quản trị click chọn **Thống kê** để thống kê doanh thu buôn bán các Sản Phẩm theo ngày, tháng hoặc năm

URL: <http://localhost:3000/adstatictical>

P&P SHOP	TỔNG DOANH THU		DOANH THU THÁNG	
	Doanh thu theo:		Select date	
☰ Thông tin cá nhân > ☰ Thư viện ảnh > ☰ Sản Phẩm 📄 Hóa Đơn 📦 Đơn Nhập Hàng 💬 Bình Luận 👤 Thẻ 🏷️ Thương Hiệu 📁 Loại Sản Phẩm 🏠 Nhà cung cấp 👤 Khách Hàng 👤 Quản Trị Viên 📊 Thống kê doanh thu	Ngày đặt	Tổng tiền	Trạng thái	
	09/05/2023	1.304.000 đ	➡	🗑️
	07/05/2023	4.395.000 đ	➡	🗑️
	27/04/2023	3.724.000 đ	➡	🗑️
	27/04/2023	721.000 đ	➡	🗑️
	27/04/2023	1.304.000 đ	➡	🗑️
	22/04/2023	633.000 đ	➡	🗑️
	21/04/2023	1.304.000 đ	➡	🗑️
	21/04/2023	1.304.000 đ	➡	🗑️
	21/04/2023	1.304.000 đ	➡	🗑️

- Tại mục **Tổng doanh thu** có thể xem và lọc doanh thu theo ngày và một khoảng thời gian nhất định và có thể xem, xóa hóa đơn



- Tại mục **Doanh thu tháng** có thể xem doanh thu tháng theo từng ngày theo dạng biểu đồ để giúp quan sát sự tăng trưởng kinh tế trong buôn bán hơn

IV: KẾT LUẬN

1. KẾT QUẢ ĐẠT ĐƯỢC.

- Tìm hiểu về cách thức hoạt động kinh doanh thương mại.
- Tìm hiểu về các giải pháp để xây dựng một website thương mại điện tử như thế nào.
- Xây dựng một website mang tính thương mại hóa.
- Xây dựng một trang web với đầy đủ các chức năng để phục vụ cho Quản Trị viên nói riêng và Khách Hàng nói chung.
- Đòi hỏi một môi trường thân thiện với người tiêu dùng.

1.1 Ưu điểm.

- Giới thiệu sản phẩm đến khách hàng. Đối với khách hàng: cho phép thực hiện các chức năng xem hàng hóa, góp ý.
- Sau khi mua sản phẩm, đặt mua sản phẩm và thanh toán bằng nhiều cách khác nhau.
- Khách hàng có thể thay đổi mật khẩu và mật khẩu thanh toán trong thông tin cá nhân. Khách hàng có thể xem thông tin đơn hàng của mình.
- Hoàn thành hầu hết các chức năng cơ bản của một trang web bán hàng.
- Xây dựng trên mô hình 3 layer dễ dàng quản lý việc thay đổi cấu trúc sau này.
- Website áp dụng giao thức bảo mật SSL nên thông tin đã bảo mật hơn.

1.2 Nhược điểm.

Về dữ liệu .

- + Dữ liệu chưa được đầy đủ.

Về giao diện

- + Giao diện chưa sinh động
- + Giao diện cứng, không thể thay đổi template cho giao diện.

2. HƯỚNG PHÁT TRIỂN

Với những thiếu sót như phân tích trên, để phần mềm ngày càng hoàn thiện, phục vụ tốt cho nhu cầu của người dùng thì nên có những cập nhật sau đây:

- ⇒ Thiết kế trang web đẹp hơn, tiện dụng hơn, có nhiều chức năng và sinh động hơn.
- ⇒ Xây dựng thêm một diễn đàn trên trang web để khách hàng có thể trao đổi bình luận về sản phẩm. Thiết kế thêm vị trí để cập nhật giá cả thị trường. Ngăn ngừa tối đa các trường hợp đột nhập dữ liệu hệ thống. Chỉnh sửa dữ liệu cho chính xác với thực tế và sẽ nâng cấp dần Website để đáp ứng được những nhu cầu của khách hàng trong thời đại mới.
- ⇒ Cần được thiết kế Website chuẩn cho các thiết bị khác như điện thoại, máy tính bảng,... bằng Responsive Web Design, vì hầu như điện thoại ngày càng phát triển và được sử dụng rộng rãi hơn từ đó dễ dàng tiếp cận được nhiều khách hàng hơn trong tương lai.
- ⇒ Hiện tại, Website còn tuân theo kiểu giao hàng truyền thống bằng tiền mặt, cần được tích hợp cổng thanh toán trực tuyến như: VNPay, PayPal, Momo,... để thuận tiện cho việc buôn bán và phát triển sau này.
- ⇒ Ngoài ra, không ngừng tìm tòi, phát triển và tích hợp cho Web bằng những công nghệ mới nhưng vẫn không kém phần ổn định để Website có thể hạn chế thời gian cập nhật và sửa chữa.
- ⇒ Cuối cùng, để đạt được hiệu quả tối đa tiếp cận khách hàng và giảm thiểu chi phí phát triển Web mới, cần nâng cấp Website hiện tại thành một chuỗi cửa hàng thay vì một cửa hàng hơn lẻ. Để thiết kế một lần nhưng dùng cho nhiều doanh nghiệp hơn sau này.

-----HẾT-----