

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIA ĐỊNH
KHOA: CÔNG NGHỆ THÔNG TIN**



TIỂU LUẬN

**TÊN ĐỀ TÀI: WEBSITE GIỚI THIỆU, BÁN
THỜI TRANG NAM, NỮ TRỰC TUYẾN**

Khoa: CÔNG NGHỆ THÔNG TIN

Chuyên ngành: KỸ THUẬT PHẦN MỀM

Giảng viên hướng dẫn: Ths. LÊ HUỖNH PHƯỚC

Sinh viên thực hiện: PHAN TRUNG NAM

MSSV: 2109110261

Lớp: K15DCPM06

TP. Hồ Chí Minh, tháng 04 năm 2024

Khoa/Viện: Công Nghệ Thông Tin

NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN
TIỂU LUẬN MÔN: LẬP TRÌNH NODEJS

1. **Họ và tên sinh viên: Phan Trung Nam**
2. **Tên đề tài: Website giới thiệu, bán thời trang nam, nữ trực tuyến**
3. **Nhận xét:**

a) Những kết quả đạt được:

.....

.....

.....

.....

.....

b) Những hạn chế:

.....

.....

.....

.....

.....

4. **Điểm đánh giá (theo thang điểm 10, làm tròn đến 0.5):**

Sinh viên:

Điểm số: Điểm chữ:

TP. HCM, ngày ... tháng năm 2024

Giảng viên chấm thi

(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ƠN	4
CHƯƠNG 1: HIỆN TRẠNG VÀ YÊU CẦU	5
1.1 Hiện trạng	5
1.2 Yêu cầu	8
1.3 Các biểu mẫu thống kê	9
1.4. Mô hình hóa yêu cầu	11
CHƯƠNG 2: PHÂN TÍCH	19
2.1. Sơ đồ lớp (Class Diagram)	19
2.2. Sơ đồ tuần tự	19
2.3. Sơ đồ hoạt động (Activity Diagram)	23
2.4. Sơ đồ hoạt động (ERD)	24
CHƯƠNG 3: THIẾT KẾ	25
3.1. Kiến trúc phần mềm	25
3.2. Thiết kế CSDL	25
3.2.1 Giao diện chính của mongoDB compass	25
3.2.2. Table collection	25
3.2.3. Collection cho khóa lưu là sản phẩm	26
3.3. Thiết kế giao diện	26
3.3.1 Giao diện chính của sản phẩm	26
3.3.2 Giao diện trang chi tiết sản phẩm	27
3.3.2 Giao diện trang thêm mới sản phẩm	27
3.3.3 Giao diện trang quản lý sản phẩm	27
CHƯƠNG 4: CÀI ĐẶT KIỂM NGHIỆM	28
4.1. Cài đặt	28
4.1.1 Install expressjs	28
4.1.2 Install Nodemon	28
4.1.3 Install Morgan	29
4.1.4 Install Template engine (handlebars)	29
4.2. Các thử nghiệm	29

Kiểm thử đơn vị (Unit Testing)	29
Kiểm thử tích hợp (Integration Testing)	29
Kiểm thử đầu cuối (End-to-End Testing)	29
Kiểm thử hiệu suất (Performance Testing)	30
CHƯƠNG 5: TỔNG KẾT	31
5.1. Kết quả đạt được	31
5.2. Đánh giá ưu, khuyết điểm	31
5.2.1. Ưu điểm	31
5.2.2. Khuyết điểm	31
5.2.3. Hướng phát triển của đề tài	31
TÀI LIỆU THAM KHẢO	33

LỜI CẢM ƠN

Xin chân thành cảm ơn đến trường Đại Học Gia Định khoa công nghệ Thông tin đã đưa môn học “ Lập trình Nodejs ” vào chương trình giảng dạy. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn - Thầy Lê Huỳnh Phước đã dạy dỗ, truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học, em được học những kỹ năng lập trình, về ngôn ngữ tiếp cận và hiện đại của thế giới đó là javascript, tất cả mọi mặt về tư duy code, học được những kiến thức cơ bản về ngôn ngữ javascript và nodejs. Học tập ở thầy những bài học hay và có thể sẽ rất giúp ích em cho những chặn đường sau này.

Đây chắc chắn là những kiến thức quý báu, là hành trang để em có thể vững bước trên con đường em đã chọn, bộ môn “Lập trình nodejs” là môn học thú vị, vô cùng bổ ích và có tính thực tế rất cao. Cung cấp được những kiến thức gắn liền với nhu cầu thực tiễn của sinh viên. Em sẽ cố gắng hoàn thành thật tốt bài tiểu luận này, nhưng sẽ vẫn còn nhiều thiếu sót không mong muốn, kính mong thầy xem xét và góp ý để bài tiểu luận của em có thể hoàn thiện và chính chu hơn.

Một lần nữa em xin chân thành cảm ơn!

CHƯƠNG 1: HIỆN TRẠNG VÀ YÊU CẦU

1.1 Hiện trạng

1.1.1 Tính linh hoạt

- Một trong những lý do chính khiến Node.js được ưa chuộng là tính linh hoạt và dễ dàng mở rộng. Công nghệ này cho phép phát triển ứng dụng web đa nền tảng, trong đó web bán hàng thời trang không phải là một ngoại lệ.

- Node.js là một nền tảng JavaScript mã nguồn mở cho phép phát triển các ứng dụng mạng hiệu quả và có thể mở rộng. Nó được biết đến với tính linh hoạt, có thể được sử dụng để xây dựng nhiều loại ứng dụng khác nhau, từ các trang web đơn giản đến các ứng dụng web phức tạp và các ứng dụng thời gian thực.

- **Mô hình lập trình phi đồng bộ:** Node.js sử dụng mô hình lập trình phi đồng bộ, điều này có nghĩa là nó có thể xử lý nhiều yêu cầu cùng một lúc mà không bị chặn. Điều này làm cho Node.js lý tưởng cho việc phát triển các ứng dụng web có thể mở rộng và hiệu suất cao.

- **Khả năng mở rộng:** Node.js có thể dễ dàng mở rộng bằng cách thêm thêm máy chủ vào cụm. Điều này làm cho nó lý tưởng cho việc phát triển các ứng dụng cần phải xử lý một lượng lớn lưu lượng truy cập.

- **Cộng đồng lớn và năng động:** Node.js có một cộng đồng lớn và năng động bao gồm các nhà phát triển, doanh nghiệp và tổ chức. Điều này có nghĩa là có rất nhiều tài nguyên sẵn có để giúp bạn bắt đầu với Node.js và giải quyết các vấn đề mà bạn có thể gặp phải.

- **Hỗ trợ nhiều nền tảng:** Node.js chạy trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS và Linux. Điều này làm cho nó lý tưởng cho việc phát triển các ứng dụng cần phải triển khai trên nhiều môi trường khác nhau.

Hệ sinh thái phong phú: Node.js có một hệ sinh thái phong phú gồm các thư viện và công cụ của bên thứ ba có thể được sử dụng để xây dựng các ứng dụng. Điều này có nghĩa là bạn có thể tìm thấy các giải pháp cho hầu hết mọi vấn đề mà bạn có thể gặp phải khi phát triển ứng dụng Node.js.

1.1.2 Tính năng thời gian thực

- Trong môi trường web bán hàng thời trang, việc cập nhật sản phẩm, giá cả, và thông tin khuyến mãi là rất quan trọng. Node.js, với kiến trúc non-blocking và sự hỗ trợ của các thư viện như Socket.io, là một lựa chọn tốt cho các tính năng thời gian thực như thông báo mới, chat trực tuyến với khách hàng, vv.
- Node.js cung cấp một số tính năng mạnh mẽ hỗ trợ phát triển ứng dụng thời gian thực. Những tính năng này cho phép bạn xây dựng các ứng dụng có thể cập nhật dữ liệu cho người dùng theo thời gian thực, mà không cần phải tải lại trang, dưới đây là một số tính năng thời gian thực chính của Node.js:
- WebSockets: WebSockets là một giao thức mạng cho phép kết nối hai chiều thời gian thực giữa máy chủ web và trình duyệt web. Node.js cung cấp hỗ trợ tích hợp cho WebSockets, giúp bạn dễ dàng xây dựng các ứng dụng trò chuyện, bảng tin và các ứng dụng thời gian thực khác.
- Socket.IO: Socket.IO là một thư viện mã nguồn mở cho Socket.IO. Nó cung cấp một lớp trừu tượng hóa trên WebSockets, giúp bạn dễ dàng sử dụng các tính năng thời gian thực trong Node.js. Socket.IO hỗ trợ nhiều giao thức truyền tải khác nhau, bao gồm WebSockets, HTTP và Flash.
- Server-Sent Events (SSE): Server-Sent Events (SSE) là một API cho phép máy chủ web đẩy dữ liệu đến trình duyệt web mà không cần yêu cầu của trình duyệt. SSE là một lựa chọn tốt cho việc xây dựng các ứng dụng cần cập nhật dữ liệu cho người dùng thường xuyên, chẳng hạn như bảng tin chứng khoán hoặc ứng dụng theo dõi thể thao.
- WebRTC: WebRTC là một API cho phép trình duyệt web giao tiếp trực tiếp với nhau mà không cần máy chủ trung gian. WebRTC có thể được sử dụng để xây dựng các ứng dụng trò chuyện video, hội nghị truyền hình và các ứng dụng thời gian thực khác.

1.1.3 Khái niệm SSR và CSR

- SSR (Server-Side Rendering) và CSR (Client-Side Rendering) là hai phương pháp phổ biến để hiển thị nội dung trang web. Cả hai đều có những ưu điểm và nhược điểm riêng, và lựa chọn phương pháp nào tốt nhất cho một ứng dụng đều phụ thuộc vào nhu cầu cụ thể của ứng dụng đó.

- **SSR** hoạt động bằng cách máy chủ tạo ra HTML đầy đủ cho mỗi trang trong ứng dụng của bạn và gửi nó đến trình duyệt của người dùng. Điều này có nghĩa là nội dung trang được hiển thị ngay lập tức khi người dùng truy cập trang, mà không cần phải tải thêm JavaScript.

- *Ưu điểm của SSR:*

+ Hiệu suất: SSR có thể cải thiện hiệu suất trang, đặc biệt là đối với các trang web có nhiều nội dung tĩnh.

+ SEO: SSR tốt cho SEO vì nó giúp công cụ tìm kiếm dễ dàng lập chỉ mục nội dung trang web của bạn.

+ Khả năng tiếp cận: SSR tốt cho khả năng tiếp cận vì nó đảm bảo rằng nội dung trang web của bạn có thể truy cập được ngay cả khi người dùng không có JavaScript.

- *Nhược điểm của SSR:*

+ Hiệu suất máy chủ: SSR có thể tốn nhiều tài nguyên máy chủ hơn CSR, đặc biệt là đối với các ứng dụng web có nhiều lưu lượng truy cập.

+ Tính tương tác: SSR có thể làm giảm tính tương tác của người dùng vì nó yêu cầu tải lại trang đầy đủ để cập nhật nội dung.

- **CSR** hoạt động bằng cách máy chủ gửi một trang HTML cơ bản đến trình duyệt của người dùng và sau đó sử dụng JavaScript để tải và hiển thị nội dung trang động. Điều này có nghĩa là nội dung trang có thể được cập nhật mà không cần tải lại trang đầy đủ.

- *Ưu điểm của CSR:*

+ Hiệu suất máy chủ: CSR có thể giảm tải máy chủ, đặc biệt là đối với các ứng dụng web có nhiều lưu lượng truy cập.

- + Tính tương tác: CSR có thể cải thiện tính tương tác của người dùng vì nó cho phép cập nhật nội dung trang mà không cần tải lại trang đầy đủ.
- + Trải nghiệm người dùng: CSR có thể tạo ra trải nghiệm người dùng mượt mà và giống ứng dụng hơn.
- *Nhược điểm của CSR:*
 - + Hiệu suất trang: CSR có thể làm giảm hiệu suất trang, đặc biệt là đối với các trang web có nhiều JavaScript.
 - + SEO: CSR có thể gây khó khăn cho công cụ tìm kiếm để lập chỉ mục nội dung trang web của bạn.
 - + Khả năng tiếp cận: CSR có thể làm giảm khả năng tiếp cận vì nó phụ thuộc vào JavaScript để hiển thị nội dung trang.

1.1.4 Hiệu suất cao

- Web bán hàng thời trang thường phải xử lý lượng lớn dữ liệu và yêu cầu hiệu suất cao. Node.js với kiến trúc non-blocking I/O giúp tối ưu hóa hiệu suất cho ứng dụng web, giúp đảm bảo trải nghiệm người dùng tốt nhất.
- Nodejs sử dụng mô hình phi lập trình phi đồng bộ , khả năng sử dụng event-loop, có thể xử dụng được javascript, nodejs còn có thể hỗ trợ V8, có tính khả năng mở rộng cao

1.2 Yêu cầu

- Danh sách các công việc sẽ được hỗ trợ thực hiện trên laptop, điện thoại...
- Chương trình cho phép:
 1. Admin - chức năng quản lý: tài khoản người dùng, sản phẩm, danh mục sản phẩm, đơn hàng, các chức năng thêm, xóa, sửa và các chức năng mà người dùng sử dụng được.
 2. Người dùng - đăng nhập, quản lý giỏ hàng, tìm kiếm sản phẩm với các chức năng thêm, sửa và xóa sản phẩm, đặt hàng.
 3. Khách hàng vắng lai: Xem sản phẩm, tìm kiếm sản phẩm, đăng ký.

1.3 Các biểu mẫu thống kê

- Biểu mẫu tài khoản người dùng:

Tài khoản người dùng	
Mã số tài khoản:	
Tên tài khoản:	
Mật khẩu:	
Phân quyền:	
<p>Quy định 1: Mã số tài khoản được xếp theo thứ tự tăng dần</p> <p>Quy định 2: Phải có cả tên tài khoản và mật khẩu để tạo nên 1 tài khoản người dùng</p> <p>Quy định 3: Phân quyền là cấp độ được phân quyền của tài khoản người dùng</p>	

Biểu mẫu Tài khoản người dùng

- Biểu mẫu danh mục sản phẩm

Danh sách các danh mục				
ID sản phẩm	ID thể loại	Tên sản phẩm	Tên thể loại sản phẩm	Giá sản phẩm

Biểu mẫu danh mục sản phẩm

- Biểu mẫu sản phẩm

Sản phẩm			
ID sản phẩm:		Tên sản phẩm:	
ID thể loại sản phẩm:		Tên thể loại sản phẩm:	
Giá sản phẩm:			
<p>Quy định 4: Các thông tin phải được điền đủ</p> <p>Quy định 5: ID sản phẩm, và tên sản phẩm không lặp với những sản phẩm đã có</p>			

Biểu mẫu sản phẩm

- Biểu mẫu danh sách tài khoản người dùng

Danh sách tài khoản người dùng			
ID tài khoản	Tên tài khoản	Mật khẩu	Phân quyền

Biểu mẫu tài khoản người dùng

- Biểu mẫu thể loại sản phẩm

Danh sách thể loại sản phẩm	
ID thể loại	Tên thể loại
<p>Quy định 6: Các thông tin phải được điền đủ</p>	

Biểu mẫu thể loại sản phẩm

1.4. Mô hình hóa yêu cầu

1.4.1. Xác định Actor

- Với đối tượng của Website bán hàng sẽ có 3 loại là khách xem, khách hàng, quản trị viên

+ Quản trị viên: Sẽ là người điều hành website, tương tác với khách hàng

+ Khách hàng: là những người đã đăng ký tài khoản , họ có thể có nhu cầu mua sản phẩm hoặc tìm hiểu về các mặt hàng

+ Khách xem: là những người chưa có tài khoản, họ có thể chỉ đến xem mà chưa thật sự có nhu cầu mua sản phẩm

- Vì vậy các Actor được đề ra sẽ là: Quản trị viên(Administrator), Khách hàng(Customer) và khách xem(Guest).

1.4.2. Xác định User Case

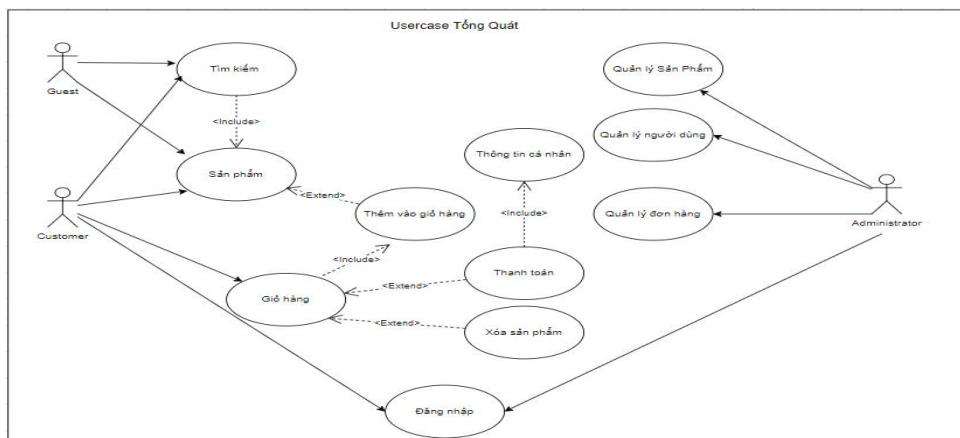
- Với Actor Quản trị viên: Sẽ có các UserCase thể hiện tính năng có thể sử dụng :Quản lý khách hàng, quản lý nhân viên, quản lý sản phẩm, quản lý danh mục, quản lý đơn hàng.

-Với Actor Guest: Sẽ có các tính năng như xem sản phẩm ,tìm kiếm, xem cái khuyến mãi, tin tức, đăng ký tài khoản.

-Với Actor Khách hàng: Sẽ có các tính năng như xem sản phẩm,tìm kiếm,Đăng nhập, xem khuyến mãi,tin tức,Đăng ký tài khoản,giỏ hàng,thanh toán.

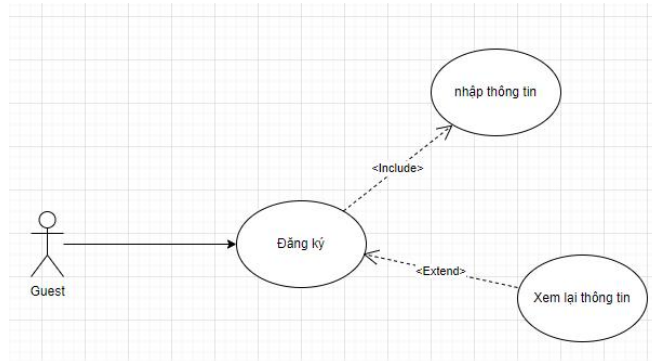
1.4.3. Sơ đồ User Case

1.4.3.1. Sơ đồ User Case tổng quát



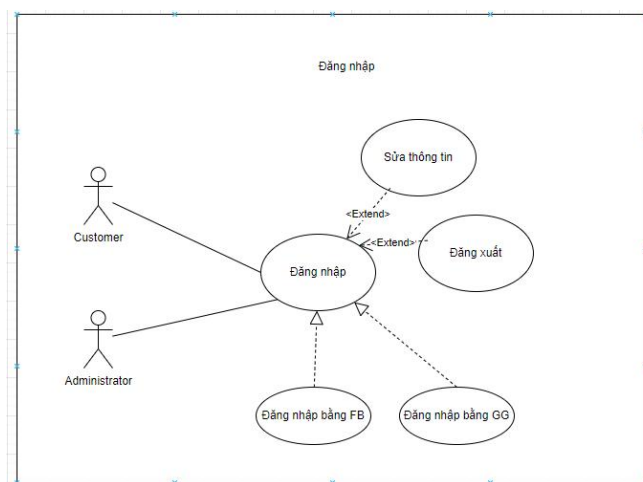
- Sơ đồ khái quát về website có các Actor như Guest(Khách xem), Customer(Khách hàng), Administrator(Quản trị viên) và các User case thể hiện chức năng mà actor có thể dùng.

1.4.3.2. Sơ đồ User Case đăng ký



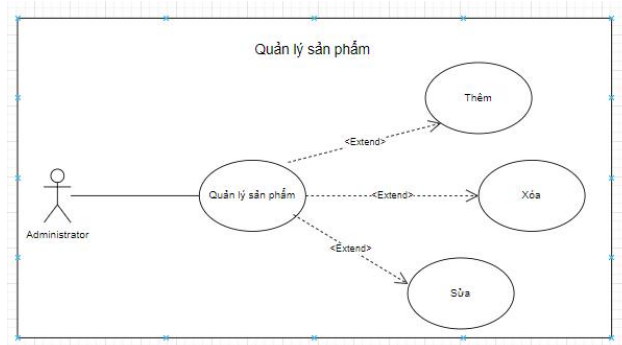
- Sơ đồ User Case thể hiện chức năng đăng ký khi khách xem(Guest) muốn đăng ký để trở thành khách hàng.

4.3.3. Sơ đồ User Case đăng nhập



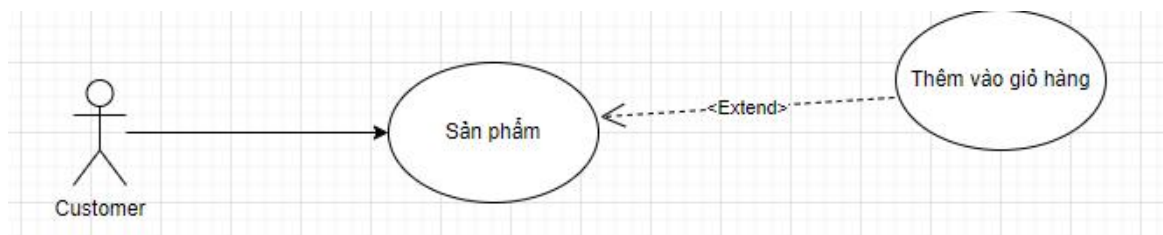
- Sơ đồ thể hiện chức năng đăng nhập của Actor Customer ,Actor Administrator và có các chức năng như đăng xuất, tạo tk, lấy lại mk.

1.4.3.4. Sơ đồ User Case quản lý sản phẩm



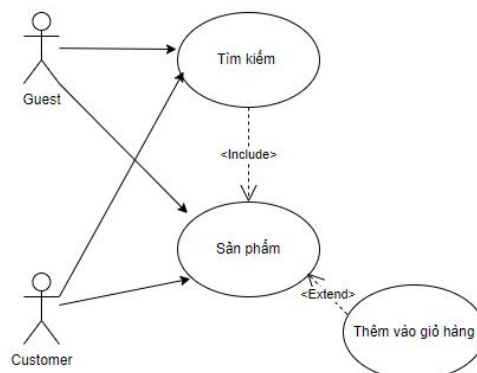
- Sơ đồ thể hiện chức năng quản lý sản phẩm của quản trị viên (Administrator) với các lựa chọn như thêm, sửa, xóa

1.4.3.5. Sơ đồ User Case thêm vào giỏ hàng



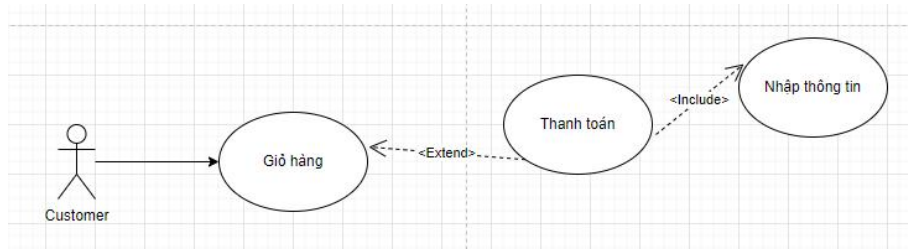
- Sơ đồ thể hiện chức năng thêm sản phẩm vào giỏ hàng khi người dùng quyết định mua món hàng đó.

1.4.3.6. Sơ đồ User Case tìm kiếm sản phẩm



- Sơ đồ thể hiện chức năng tìm kiếm sản phẩm thông qua dữ liệu sản phẩm hiện có.

1.4.3.7. Sơ đồ User Case thanh toán



- Sơ đồ thể hiện chức năng thanh toán khi người dùng đã chọn xong sản phẩm và quyết định mua hàng.

1.4.4. Mô tả các User Case

1.4.4.1. Sơ đồ User Case đăng ký

User Case	Đăng ký
Actor	Guest
Mô tả	Để Guest có thể đăng ký để có tài khoản để làm thành viên của website và sử dụng nhiều dịch vụ từ website đây.
Điều kiện tiên quyết	Khách xem ghé thăm website
Dòng sự kiện chính	1.Khách xem chọn mục đăng ký thành viên 2.Form đăng ký thành viên hiển thị 3.Khách xem nhập thông tin vào form đăng ký 4.Nhấp nút đăng ký 5.Hệ thống thông báo kết quả quá trình đăng nhập cá nhân 6.Hệ thống cập nhật thông tin của khách xem(Guest) vào danh sách khách hàng(Customer).Hệ thống hiển thị và yêu cầu nhập thông tin
Dòng sự kiện phụ	- Actor nhập sai thông tin - Hệ thống báo lỗi và yêu cầu nhập

		lại - Actor lặp lại bước 3
Điều kiện cuối	kiện	Khách xem(Guest) trở thành khách hàng(Customer) của hệ thống

1.4.4.2. Sơ đồ User Case đăng nhập

User Case	Đăng nhập
Actor	Customer, Administrator
Mô tả	Để Customer, Administrator có thể đăng nhập vào website và sử dụng dịch vụ từ website đây.
Điều kiện tiên quyết	<ul style="list-style-type: none"> - Tài khoản người dùng đã được tạo sẵn - Tài khoản người dùng đã được phân quyền - Thiết bị người dùng có kết nối internet khi thực hiện đăng nhập
Dòng sự kiện chính	<ol style="list-style-type: none"> 1.Actor khởi động giao diện chương trình 2.Hệ thống hiển thị và yêu cầu nhập thông tin 3.Actor nhập thông tin cần thiết, chọn nút đăng nhập 4.Hệ thống kiểm tra thông tin
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Actor nhập sai thông tin - Hệ thống báo lỗi và yêu cầu nhập lại - Actor lặp lại bước 3
Điều kiện cuối	<ul style="list-style-type: none"> - Người dùng đăng nhập vào website thành công - Hệ thống ghi nhận hoạt động đăng nhập thành công

1.4.4.3. Sơ đồ User Case quản lý đơn hàng

User Case	Quản lý đơn hàng
Actor	Administrator
Mô tả	Để Administrator có thể kiểm soát được đơn hàng của website.
Điều kiện tiên quyết	<ul style="list-style-type: none">- Tài khoản sử dụng phải là quản trị.- Actor quản lý phải đăng nhập được vào hệ thống.- Chọn chức năng quản lý đơn hàng.- Hệ thống sẵn sàng.
Dòng sự kiện chính	<ol style="list-style-type: none">1.Quản trị viên chọn quản lý đơn hàng2.Hệ thống hiển thị trang quản lý đơn hàng3.Actor Administrator cập nhật trạng thái đơn hàng4.Hệ thống ghi nhận thông tin, xuất thông báo và hiển thị thông tin
Điều kiện cuối	<ul style="list-style-type: none">- Thông tin của đơn hàng được lưu theo xác nhận của actor Administrator.

1.4.4.4. Sơ đồ User Case thêm sản phẩm vào giỏ hàng

User Case	Sản phẩm
Actor	Customer
Mô tả	<ul style="list-style-type: none">- Để khách hàng có thể xem sản phẩm và thêm vào giỏ hàng- User case cho phép khách hàng đưa sản phẩm được chọn vào giỏ hàng.
Điều kiện tiên quyết	<ul style="list-style-type: none">- Sản phẩm được người dùng chọn
Dòng sự kiện chính	<ol style="list-style-type: none">1.Chọn chức năng thêm vào giỏ hàng để đưa sản phẩm lưu trữ trong giỏ hàng2.Hệ thống hiển thị thông tin sản phẩm đưa vào giỏ hàng
Điều kiện cuối	<ul style="list-style-type: none">- Sản phẩm được thêm vào giỏ hàng

1.4.4.5. Sơ đồ User Case tìm kiếm sản phẩm

User Case	Tìm kiếm
Actor	Guest, Customer
Mô tả	Tìm kiếm sản phẩm theo yêu cầu của khách hàng
Điều kiện tiên quyết	- Khách hàng đang ở website cửa hàng
Dòng sự kiện chính	1.cho phép tìm kiếm các sản phẩm 2.Nhập vào form tìm kiếm 3.Actor nhập thông tin cần thiết vào form tìm kiếm 4.Nhấn nút tìm kiếm
Dòng sự kiện phụ 1	- Khách hàng dừng tìm kiếm - Hệ thống hủy việc tìm kiếm sản phẩm
Dòng sự kiện phụ 2	- Nhập sai từ khóa - Hệ thống báo không có sản phẩm tương ứng - Khách hàng lặp lại bước 3
Điều kiện cuối	- Hệ thống thông báo kết quả tìm kiếm.

1.4.4.6. Sơ đồ User Case giỏ hàng

User Case	Sản phẩm
Actor	Customer
Mô tả	- Lưu trữ những sản phẩm mà khách hàng muốn mua, có thể xem hoặc xóa các sản phẩm đã chọn khỏi giỏ hàng
Điều kiện tiên quyết	- Có sản phẩm trong giỏ hàng
Dòng sự kiện chính	1.Chọn chức năng xem giỏ hàng 2.Hệ thống hiển thị thông tin sản phẩm trong giỏ hàng 3.Có chức năng thêm vào và xóa khỏi giỏ hàng

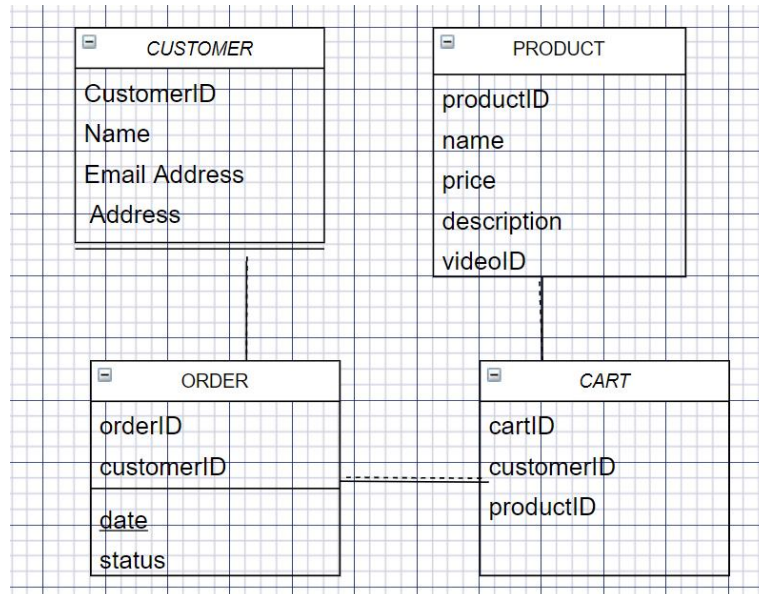
Điều kiện cuối	- Không
----------------	---------

1.4.4.7. Sơ đồ User Case thanh toán

User Case	Thanh toán
Actor	Customer
Mô tả	Thanh toán các sản phẩm đã cho vào giỏ hàng
Điều kiện tiên quyết	- Có sản phẩm trong giỏ hàng
Dòng sự kiện chính	1.chọn chức năng xem giỏ hàng 2.hệ thống hiển thị thông tin sản phẩm trong giỏ hàng 3.Chọn chức năng thanh toán 4.Nhập thông tin vào form 5.Ấn nút xác nhận
Dòng sự kiện phụ 1	- Khách hàng dừng nhập thông tin - Hệ thống hủy việc nhập thông tin khách hàng
Dòng sự kiện phụ 2	- Nhập sai thông tin - Hệ thống báo lỗi - Khách hàng lặp lại bước 4.
Điều kiện cuối	- Những thông tin xuất ra từ giỏ hàng được gửi về mục Quản lý đơn hàng của Administrator để xử lý.

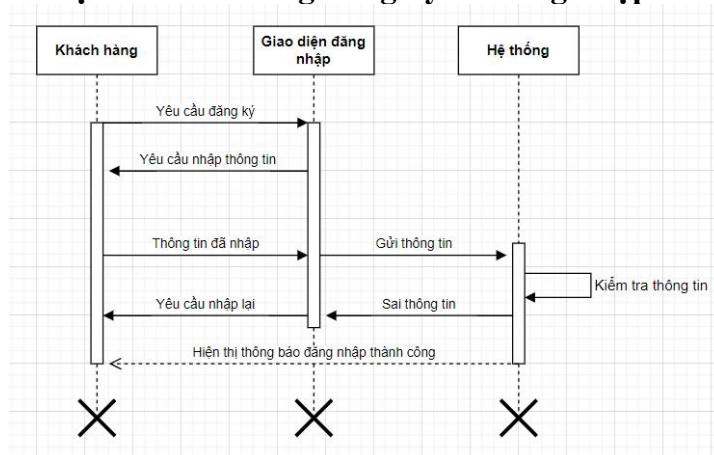
CHƯƠNG 2: PHÂN TÍCH

2.1. Sơ đồ lớp (Class Diagram)

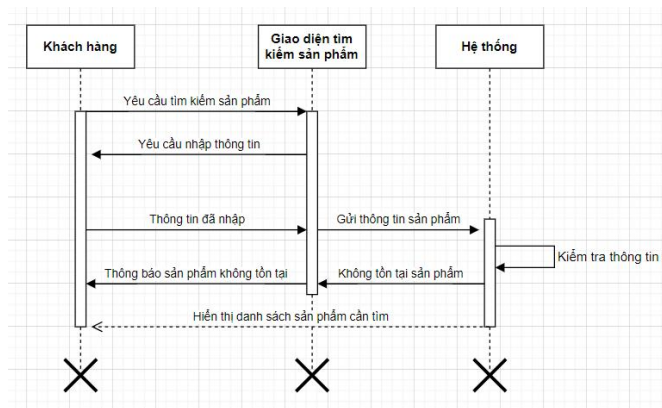


2.2. Sơ đồ tuần tự

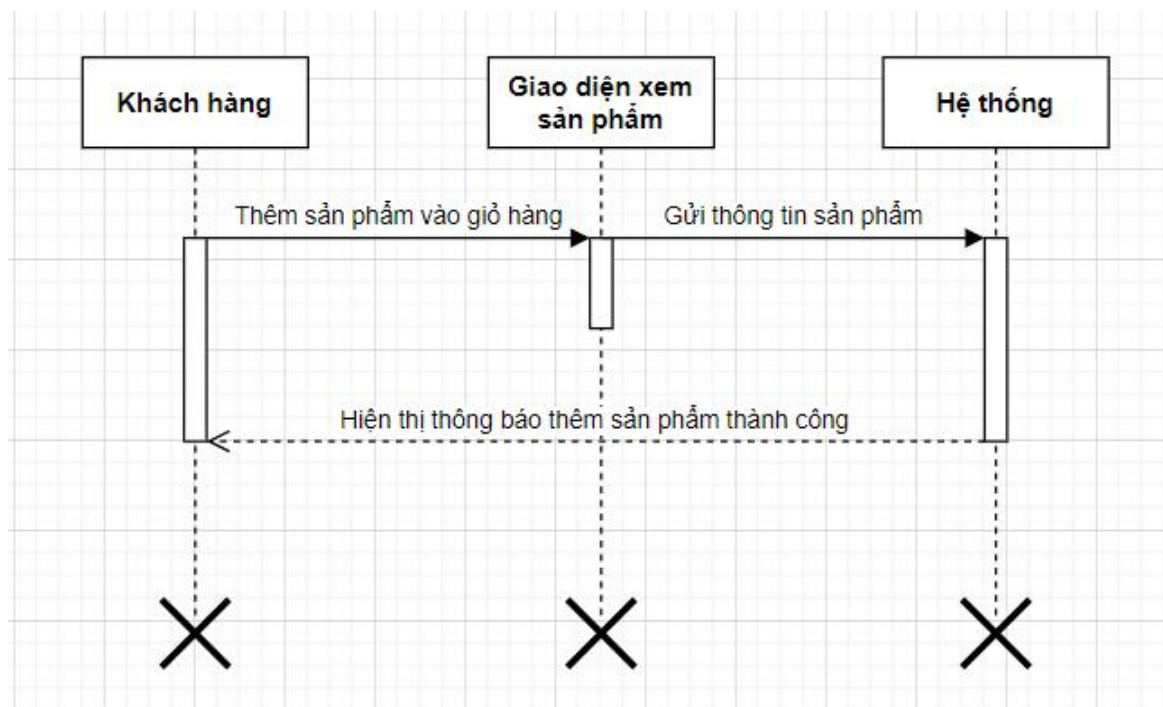
2.2.1 Sơ đồ tuần tự của chức năng đăng ký và đăng nhập



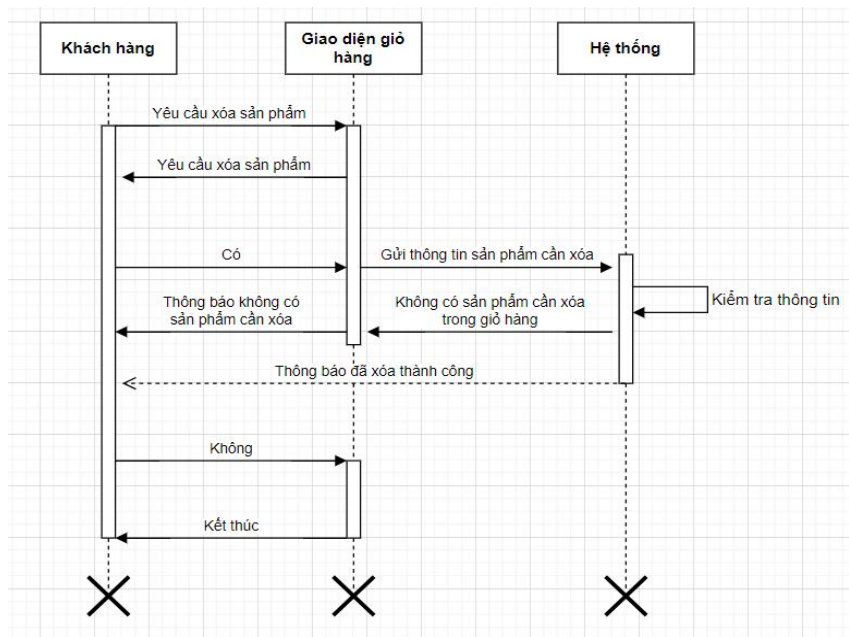
2.2.2 Sơ đồ tuần tự tìm kiếm sản phẩm



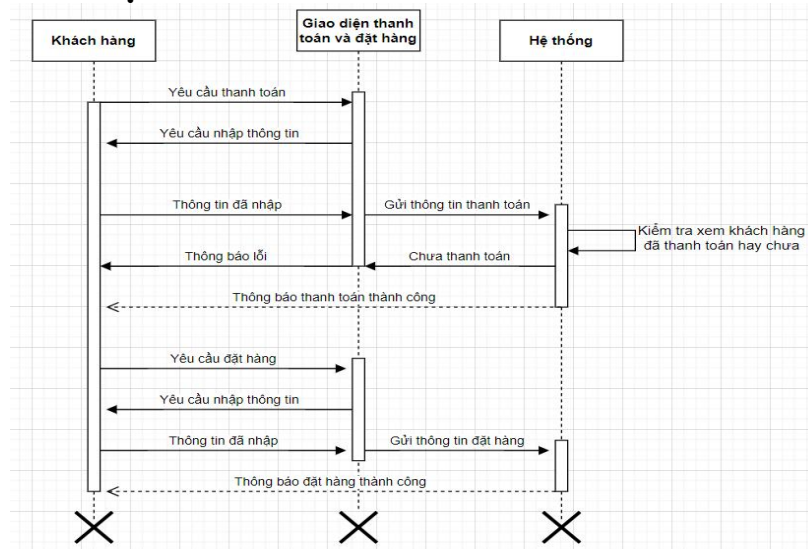
2.2.3 Sơ đồ tuần tự thêm sản phẩm vào giỏ hàng



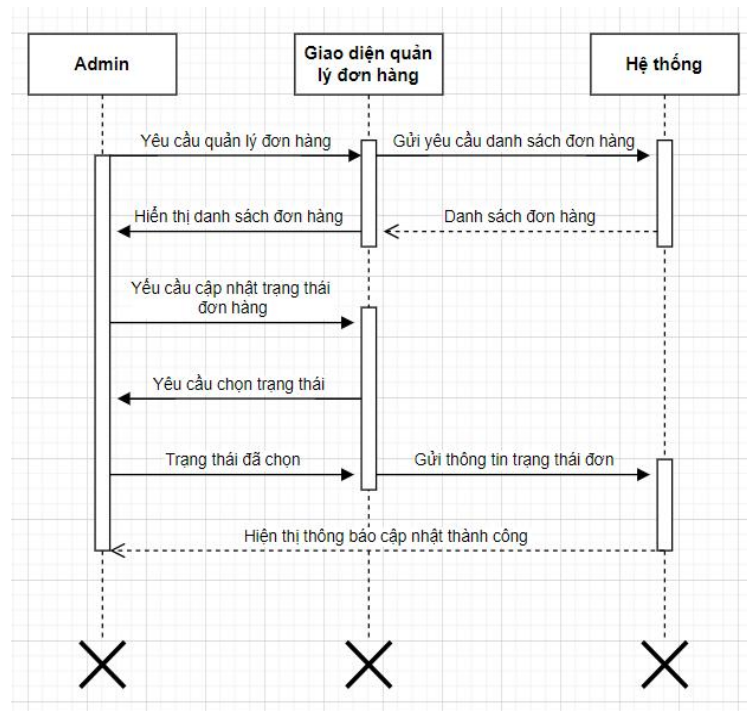
2.2.4 Sơ đồ tuần tự xóa sản phẩm trong giỏ hàng



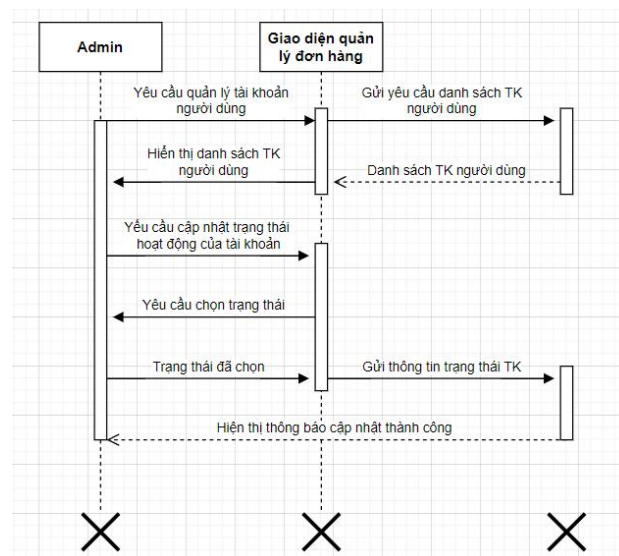
2.2.5 Sơ đồ tuần tự thanh toán



2.2.6 Sơ đồ tuần tự quản lý đơn hàng

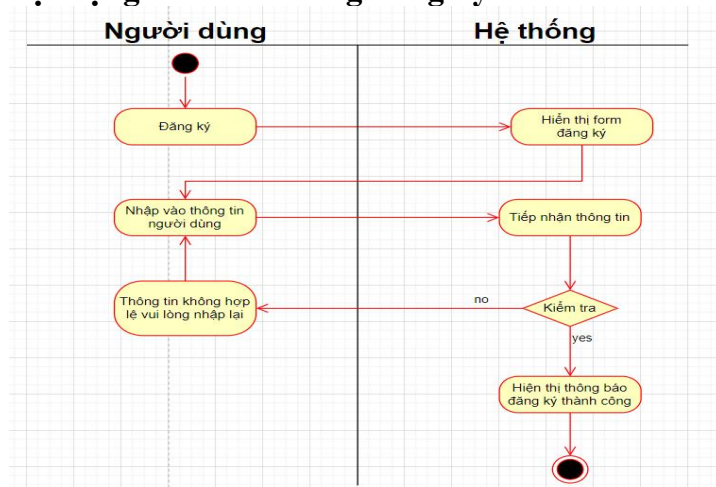


2.2.7 Sơ đồ tuần tự quản lý tài khoản người dùng

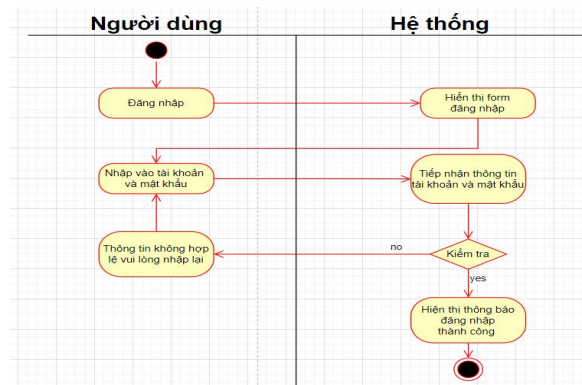


2.3. Sơ đồ hoạt động (Activity Diagram)

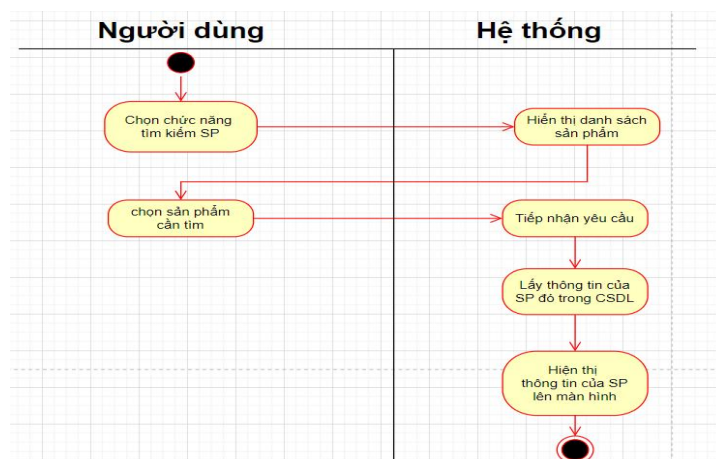
2.3.1. Sơ đồ hoạt động của chức năng đăng ký



2.3.2. Sơ đồ hoạt động của chức năng đăng nhập



2.3.3. Sơ đồ hoạt động của chức năng tìm kiếm sản phẩm



CHƯƠNG 3: THIẾT KẾ

3.1. Kiến trúc phần mềm

3.2. Thiết kế CSDL

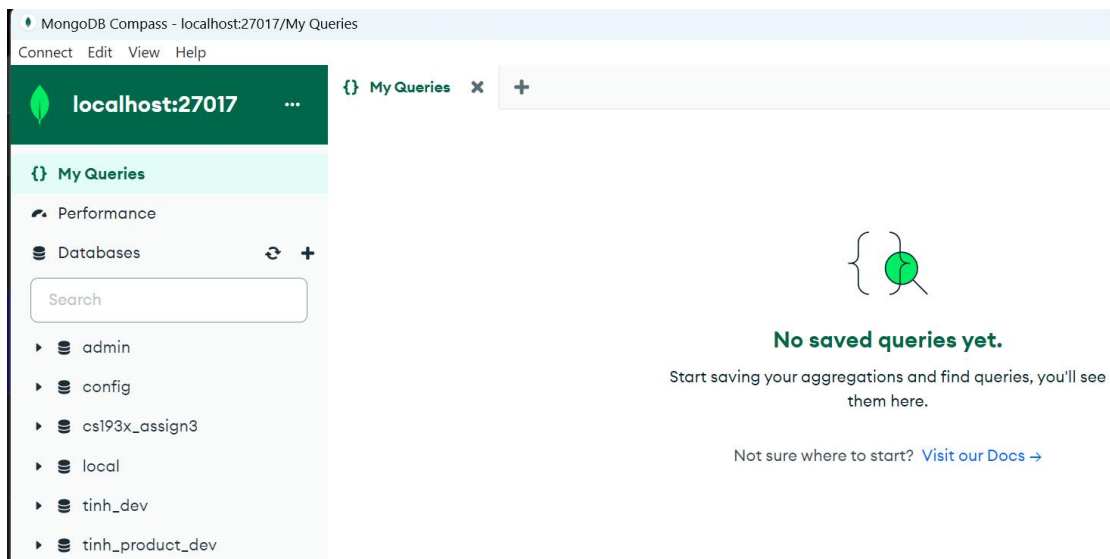
- Cơ sở dữ liệu được chia ra làm 2 loại chính:

+ Cơ sở dữ liệu quan hệ

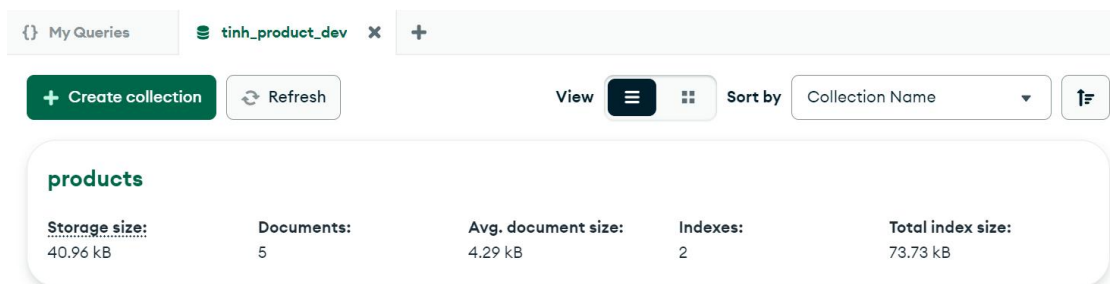
+ Cơ sở dữ liệu quan hệ không quan hệ

- Đối với nodejs thì sử dụng cơ sở dữ liệu là mongoDB để làm database lưu trữ dữ liệu là hình ảnh, giá sản phẩm, videoId của sản phẩm đó.

3.2.1 Giao diện chính của mongoDB compass



3.2.2. Table collection



3.2.3. Collection cho khóa lưu là sản phẩm

Documents 5 Aggregations Schema Indexes 2 Validation

Type a query: { field: 'value' } or [Generate query](#) [Explain](#) [Reset](#) [Find](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#) 1 - 5 of 5

```
{
  "_id": ObjectId('66082aebdcacd6bd850d627a'),
  "name": "Kappa giày thể thao Nam/Nữ 3116Z2W A04 mới lằm nha",
  "price": "790.000đ",
  "image": "https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgzViXmx-3Xt...",
  "slug": "kappa giày thể thao nam",
  "videoId": "QhHHSraD4Iw",
  "updatedAt": "2024-04-12T02:59:01.216+00:00"
}
```

3.3. Thiết kế giao diện

3.3.1 Giao diện chính của sản phẩm

POCO Mart THIÊN ĐƯỜNG MUA SẮM [Tư vấn hỗ trợ 0378277554](#) [Xin chào Thuận Tĩnh](#) [0](#)

[Danh mục sản phẩm](#) [Trang chủ](#) [Giới thiệu](#) [Sản phẩm](#) [Câu hỏi thường gặp](#) [Tuyển dụng](#)

20/9 ĐỒ GIA DỤNG - NHÀ CỬA ĐỜI SỐNG


"siêu" Sale

LÀM BẠN XIÊU LÒNG!

MUA NGAY


THỜI TRANG Thời trang nữ Thời trang nam Đồng hồ Dây dép nữ

THỜI TRANG Thời trang nữ Thời trang nam Đồng hồ Dây dép nữ



Kappa giày thể thao Nam/Nữ 3116Z2W A04 mới lằm nha


790.000đ



Áo phông Nam Nữ Thò EINSI mới

250.000đ

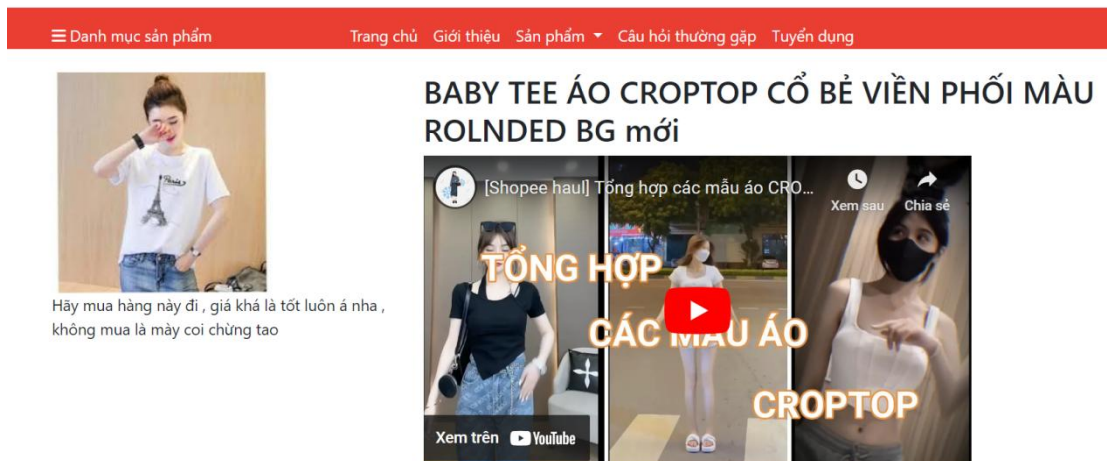
[Mua hàng](#)



BABY TEE ÁO CROPTOP CỔ BÈ VIỀN PHỐI MÀU ROLNDED BG mới

75.000đ

3.3.2 Giao diện trang chi tiết sản phẩm



3.3.2 Giao diện trang thêm mới sản phẩm

Danh mục sản phẩm

Trang chủ

Tạo mới sản phẩm

name

price

videoid

image

thêm sản phẩm

3.3.3 Giao diện trang quản lý sản phẩm

POCO Mart

THIỆN ĐƯỜNG MUA SẴM

Tìm kiếm

Tư vấn hỗ trợ

0378277554

Xin chào

Thuận Tình

0

Danh mục sản phẩm

Trang chủ

Giới thiệu

Sản phẩm

Câu hỏi thường gặp

Tuyển dụng

Sản phẩm của tôi

#	name	thời gian tạo		
1	Kappa giày thể thao Nam/Nữ 3116Z2W A04 mới lắm nha		sửa	xóa
2	Áo phông Nam Nữ Thỏ EINSI mới		sửa	xóa
3	BABY TEE ÁO CROPTOP CỔ BÈ VIỀN PHỐI MÀU ROLNDED BG mới		sửa	xóa
4	áo ba lỗ 3xxx mới	Thu Apr 04 2024 19:54:47 GMT+0700 (Indochina Time)	sửa	xóa
5	quần ship giá rẻ new	Thu Apr 04 2024 20:09:05 GMT+0700 (Indochina Time)	sửa	xóa

CHƯƠNG 4: CÀI ĐẶT KIỂM NGHIỆM

4.1. Cài đặt

- Cài đặt các thư viện hỗ trợ cho việc code website bán hàng thời trang

4.1.1 Install expressjs

- cài đặt **expressjs**
- Một số hướng dẫn cơ bản

Khởi động một dự án Nodejs

Bước 1: gõ lệnh ở terminal là **node -v** {kiểm tra xem đã cài hay chưa}

Bước 2: gõ lệnh **npm -v** {kiểm tra xem đã cài thẳng quản lí thư viện nodejs hay chưa}

Bước 3: gõ lệnh **npm init** {để khởi tạo một dự án}

Giải thích:

Description: muốn viết gì vào cũng được

Entry point: {mặc định là file index.js | muốn đổi cũng được}

- Sau khi xong mấy bước trên project sẽ tự tạo ra file package.json -> tiếp đến là ta tạo ra file index.js

4.1.2 Install Nodemon

- Công dụng của Nodemon: là lắng nghe các thay đổi của những file trong sourcecode (tức là tự nó reload lại cái dự án, k cần phải ctr +c để thoát r chạy lại , tức là nó chạy dòng lệnh nodemon index.js luôn rồi)

- hướng dẫn các cách cài đặt:

Gõ lệnh **npm install nodemon --save-dev**

Sau khi cài xong nó sẽ hiện trong file package.json ở phần dependencies

- > tiếp đến là vào file package.json -> kiểm phần scripts -> tạo ra phần "start": "nodemon --inspect index.js" -> tiếp đến là chèn thẳng --inspect (công dụng là debug lỗi) vào giữa nodemon và index.js

4.1.3 Install Morgan

- Công dụng của thằng morgan này giúp ta nhận biết được là có lỗi gì hay không, và cái request có gửi lên tới server hay chưa

- hướng dẫn cách cài đặt:

npm morgan { để cài đặt (công dụng là nó hiện cho ta thấy những cái log) }
Gõ lệnh : **npm I morgan --save-dev** để cài đặt

Vào file index.js -> import nó vào `const morgan = require('morgan')`
-> tiếp đến là cài đặt cái log theo kiểu tiêu chuẩn `app.use(morgan('combined'))`

4.1.4 Install Template engine (handlebars)

- Template engine: giúp ta tạo giao diện động, tăng khả năng tái sử dụng của html, linh hoạt dễ quản lý.

4.2. Các thử nghiệm

Kiểm thử đơn vị (Unit Testing)

- Kiểm tra các hàm và lớp riêng lẻ để đảm bảo chúng hoạt động chính xác theo yêu cầu.
- Sử dụng các thư viện kiểm thử như Mocha, Chai, và Sinon để viết các bài kiểm thử đơn vị.
- Bao gồm các trường hợp kiểm thử cho tất cả các chức năng quan trọng của ứng dụng.

Kiểm thử tích hợp (Integration Testing)

- Kiểm tra cách các thành phần khác nhau của ứng dụng hoạt động cùng nhau. Sử dụng các công cụ như Postman hoặc Insomnia để mô phỏng các yêu cầu HTTP và kiểm tra phản hồi của ứng dụng.
- Đảm bảo rằng tất cả các API endpoint hoạt động chính xác và trả về dữ liệu mong đợi.

Kiểm thử đầu cuối (End-to-End Testing)

- Kiểm tra toàn bộ chức năng của ứng dụng từ đầu đến cuối. Sử dụng các công cụ như Selenium hoặc Cypress để tự động hóa các tương tác với giao diện người dùng của ứng dụng.

- Kiểm tra các kịch bản sử dụng quan trọng của ứng dụng, chẳng hạn như đăng nhập, duyệt sản phẩm, thêm sản phẩm vào giỏ hàng và thanh toán.

Kiểm thử hiệu suất (Performance Testing)

- Đo lường hiệu suất của ứng dụng dưới tải trọng cao.
- Sử dụng các công cụ như JMeter hoặc LoadRunner để mô phỏng nhiều người dùng truy cập ứng dụng cùng lúc.
- Đảm bảo rằng ứng dụng có thể xử lý tải trọng cao mà không bị sập hoặc giảm hiệu suất.

CHƯƠNG 5: TỔNG KẾT

5.1. Kết quả đạt được

- Hiểu được quy trình hoạt động và luồng xử lý dữ liệu của nodejs
- Hiểu được cách thức hoạt động của máy chủ với máy khách
- Nắm chắc quy trình hoạt động của một browser và một server
- Áp dụng vào kiến thức cơ bản đã được học để làm được một trang web bán hàng đơn giản với đầy đủ các chức năng
- Hiểu về quy trình hoạt động và cách thức khởi chạy của các công cụ hỗ trợ để phát triển một trang web

5.2. Đánh giá ưu, khuyết điểm

5.2.1. Ưu điểm

- Người dùng có thể truy cập trang web với thiết kế dễ hiểu , dễ nhìn.
- chia từng phần bố cục rõ ràng
- Người dùng có thể truy cập vào sản phẩm và thêm vào giỏ hàng
- Đối với admin có thể quản lí sản phẩm và thêm xóa sửa sản phẩm
- hi khách hàng đăng nhập và có tài khoản thì sản phẩm đó sẽ được thêm vào giỏ hàng và chỉ cần bấm nút thanh toán

5.2.2. Khuyết điểm

- Đối với phần giỏ hàng thì chưa kết nối được tăng session và cooki
- Website còn chưa có chức năng thanh toán bằng các liên kết tài khoản mạng
- Chưa có chức năng quét mã QR cho người dùng cài app
- chưa có chức năng tìm kiếm sản phẩm cụ thể

5.2.3. Hướng phát triển của đề tài

- Do còn nhiều hạn chế ngoài ý muốn, nếu khắc phục sớm nhất có thể, thì đội ngũ nhóm sẽ cố gắng thực hiện đề cho dự án ngày càng tốt đẹp hơn

Mở rộng thị trường:

- Chọn phân khúc khách cụ thể (thời trang nam, nữ, trẻ em, cao cấp...).
- Phát triển ứng dụng di động.
- Mở rộng sang thị trường quốc tế.

Tăng trải nghiệm người dùng:

- Cá nhân hóa sản phẩm, dịch vụ.
- Tích hợp tính năng tương tác (chat, đánh giá...).
- Cung cấp dịch vụ khách hàng tuyệt vời.

Áp dụng công nghệ mới:

- Trí tuệ nhân tạo (AI) đề xuất sản phẩm, phát hiện gian lận.
- Học máy (ML) tối ưu hóa hoạt động website.

Kết hợp kênh bán hàng:

- Website, mạng xã hội, thị trường trực tuyến.
- Tham gia chương trình tiếp thị liên kết.
- Xây dựng cộng đồng:
- Diễn đàn, nhóm mạng xã hội, chương trình khách hàng thân thiết.

TÀI LIỆU THAM KHẢO

- [1].<https://www.npmjs.com/package/nodemon>
- [2].<https://expressjs.com/>
- [3].<https://github.com/Automattic/mongoose>
- [4].<https://www.npmjs.com/package/morgan>
- [5].<https://www.npmjs.com/package/mongoose-slug-generator>
- [6].<https://mongoosejs.com/>
- [7].<https://handlebarsjs.com/>
- [8].<https://snyk.io/advisor/npm-package/express-handlebars/>