

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**BÁO CÁO
CÔNG NGHỆ PHẦN MỀM
Đề tài: Website bán đồ ăn**

Trần Đức Vinh – 21020098

Nguyễn Minh Hiếu - 21020121

Nguyễn Trọng Linh - 21020546

Bùi Minh Quân - 21020382

Hà Nội - 2023

Lời cam kết

Chúng em, bao gồm trưởng nhóm (Trần Đức Vinh - 21020098) cùng các thành viên (Nguyễn Minh Hiếu - 21020121; Nguyễn Trọng Lĩnh - 21020546; Bùi Minh Quân - 21020382) xin cam kết các nội dung sau:

1. Nghiêm túc, trung thực trong quá trình làm bài tập nhóm.
2. Không sao chép nội dung bài tập từ nguồn có sẵn.

Chúng em cam đoan các điều trên là đúng sự thật trong quá trình làm bài tập lớn môn Công nghệ phần mềm (INT2208_4). Chúng em xin chịu mọi trách nhiệm nếu vi phạm hoặc thực hiện không đúng các điều kể trên theo quy định của nhà trường, lớp học cũng như của thầy cô giảng dạy học phần môn Công nghệ phần mềm (INT2208_4).

Mục lục

CHƯƠNG 1: ĐẶT VẤN ĐỀ	8
1 LÝ DO LÀM SẢN PHẨM	8
2 BỐ CỤC BÁO CÁO	9
CHƯƠNG 2: MÔ TẢ KIẾN THỨC NỀN TẢNG	10
1 TỔNG QUAN VỀ THƯƠNG MẠI ĐIỆN TỬ	10
1.1 Khái niệm về thương mại điện tử	10
1.2 Các mô hình thương mại điện tử	11
1.3 Lợi ích của thương mại điện tử	11
2 TRÌNH BÀY TỔNG QUAN VỀ MEVN STACK	12
2.1 Giới thiệu	13
2.2 VueJS	13
2.3 NodeJS	14
2.4 ExpressJS	15
2.5 MySQL, MongoDB, Redis	15
3 TỔNG KẾT CHƯƠNG	16
CHƯƠNG 3: PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU	18
1 PHÂN TÍCH YÊU CẦU	18
1.1 Yêu cầu chức năng	18
1.2 Yêu cầu phi chức năng	18
2 PHÂN TÍCH VÀ ĐẶC TẢ CA SỬ DỤNG	19
2.1 Ca sử dụng “Đăng ký tài khoản”	19
2.2 Ca sử dụng “Đăng nhập tài khoản”	20
2.3 Ca sử dụng “Thêm vào giỏ hàng”	20
2.4 Ca sử dụng “Mua hàng”	21
2.5 Ca sử dụng “Chỉnh sửa thông tin cá nhân”	21
2.6 Ca sử dụng “Xem trạng thái đơn hàng”	22
2.7 Ca sử dụng “Tìm kiếm sản phẩm”	22
2.8 Ca sử dụng “Xem thông tin sản phẩm”	23
2.9 Ca sử dụng “Thêm sản phẩm”	23
2.10 Ca sử dụng “Xóa sản phẩm”	23
2.11 Ca sử dụng “Xóa người dùng”	24
2.12 Ca sử dụng “Quản lý đơn hàng”	24
3 MÔ TẢ CÁC BIỂU ĐỒ HOẠT ĐỘNG	25
3.1 Biểu đồ hoạt động “Đăng nhập hệ thống”	25

3.2	<i>Biểu đồ hoạt động “Tìm kiếm sản phẩm”</i>	25
3.3	<i>Biểu đồ hoạt động “Đặt hàng”</i>	26
3.4	<i>Biểu đồ hoạt động “Quản lý sản phẩm”</i>	27
CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG		28
1	THIẾT KẾ KIẾN TRÚC CHUNG CỦA HỆ THỐNG	28
2	BIỂU ĐỒ TUẦN TỰ CÁC CA SỬ DỤNG CHÍNH	29
2.1	<i>Biểu đồ tuần tự “Đăng nhập”</i>	29
2.2	<i>Biểu đồ tuần tự “Tìm kiếm”</i>	29
2.3	<i>Biểu đồ tuần tự “Mua hàng”</i>	30
3	THIẾT KẾ REST API	30
4	THIẾT KẾ CƠ SỞ DỮ LIỆU	32
4.1	<i>Cơ sở dữ liệu mức căn bản</i>	32
4.2	<i>Danh sách các thành phần cơ sở dữ liệu</i>	33
4.3	<i>Quan hệ giữa các thành phần</i>	34
5	MỘT SỐ GIAO DIỆN CHÍNH	38
5.1	<i>Giao diện người dùng</i>	38
5.2	<i>Giao diện quản trị</i>	41
CHƯƠNG 5: CÀI ĐẶT VÀ KIỂM THỬ		43
1	CÀI ĐẶT HỆ THỐNG	43
2	KIỂM THỬ HỆ THỐNG	43
2.1	<i>User</i>	43
2.2	<i>Admin</i>	49
CHƯƠNG 6: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN		53
1	KẾT LUẬN	53
2	ĐỊNH HƯỚNG PHÁT TRIỂN	53
TÀI LIỆU THAM KHẢO		54
CHÚ THÍCH		55

Danh sách hình vẽ

Hình 1:	Giao diện trang chủ	9
Hình 2:	Thương mại điện tử	10
Hình 3:	MEVN stack thông thường	13
Hình 4:	MEVN stack thay thế, sử dụng trong dự án	16
Hình 5:	Ca sử dụng chung của hệ thống	19
Hình 6:	Biểu đồ hoạt động “Đăng nhập”	25
Hình 7:	Biểu đồ hoạt động “Tìm kiếm”	26
Hình 8:	Biểu đồ hoạt động “Mua hàng”	27
Hình 9:	Biểu đồ hoạt động “Quản lý sản phẩm”	27
Hình 10:	Mô hình client - server	28
Hình 11:	Mô hình MVC	28
Hình 12:	Biểu đồ tuần tự “Đăng nhập”	29
Hình 13:	Biểu đồ tuần tự “Tìm kiếm”	29
Hình 14:	Biểu đồ tuần tự “Mua hàng”	30
Hình 15:	Cách hoạt động REST API	30
Hình 16:	Xây dựng REST API xử lý các tác vụ liên quan đến user	31
Hình 17:	Thiết kế REST API các tác vụ về order	32
Hình 18:	Thiết kế cơ sở dữ liệu	33
Hình 19:	Quan hệ giữa users và user_payment	34
Hình 20:	Quan hệ giữa users và use_address	35
Hình 21:	Quan hệ giữa uses và order_details	35
Hình 22:	Quan hệ giữa users và cart_item	36
Hình 23:	Quan hệ giữa cart_item và products	36
Hình 24:	Quan hệ giữa discount và products	37
Hình 25:	Quan hệ giữa products và product_category	37
Hình 26:	Quan hệ giữa order_items và products	38
Hình 27:	Giao diện đăng ký	39
Hình 28:	Giao diện trang chủ	39
Hình 29:	Giao diện danh sách sản phẩm	40
Hình 30:	Giao diện giỏ hàng	40
Hình 31:	Giao diện đặt hàng	41
Hình 32:	Giao diện quản lý sản phẩm	41
Hình 33:	Giao diện quản lý đơn hàng	42
Hình 34:	Giao diện quản lý người dùng	42

Danh sách bảng biểu

Bảng 1 Các thành phần cơ sở dữ liệu

33

Phân chia công việc

Sau khi đăng ký bài tập lớn, chúng em đã quyết định thành lập nhóm 5 gồm 4 thành viên:

- Trần Đức Vinh
- Nguyễn Minh Hiếu
- Nguyễn Trọng Linh
- Bùi Minh Quân

Sau khi thống nhất về nhân sự và đề tài, chúng em lên kế hoạch và phân chia công việc như sau:

Tuần	Vinh	Hiếu	Linh	Quân
Lập nhóm và xác định đề tài thu thập yêu cầu				
Tuần 1	- Trao đổi, thống nhất ý tưởng - Làm quen, trao đổi về khả năng của các thành viên để xác định phân công công việc			
Tuần 2	- Trao đổi, phân tích các yêu cầu của website - Thống nhất các công nghệ sử dụng			
Mô hình hóa phần mềm				
Tuần 3	Thiết kế cơ sở dữ	Thiết kế giao diện	Thiết kế kiến trúc	Thiết kế cơ sở dữ
Tuần 4	dụng			liệu
Tuần 5	Học Sequelize, MySQL	Học VueJS	Học NodeJS, Express	Học MySQL, Redis
Tuần 6				
Cài đặt và kiểm thử				
Tuần 7	Viết các models	Thiết kế, code các	Thiết kế API, code các	Xây dựng các cấu
Tuần 10	thao tác với	giao diện và các	controllers xử lý request	trúc bảng, các ràng
	database sử	tương tác của người	từ client	buộc trong cơ sở dữ
	dụng Sequelize	dùng		liệu
Tuần 11	Viết báo cáo,	Thiết kế giao diện	Viết code sử dụng thêm	Thêm dữ liệu vào
Tuần 14	kiểm thử	của admin tương	cookie và Redis để tăng	database và kiểm
		tác của admin	hiệu suất	thử
Tuần 15	Viết báo cáo	Hoàn thiện giao	Kiểm thử	Kiểm thử
		diện		

Chương 1: Đặt vấn đề

1 Lý do làm sản phẩm

Trong thời đại công nghệ số hiện nay, việc sở hữu một trang web đã trở thành một yếu tố cần thiết để các doanh nghiệp, đặc biệt là nhà hàng, chuỗi cửa hàng có thể tiếp cận được với khách hàng. Thực tế, nhiều khách hàng đã trở nên thông thạo trong việc sử dụng internet để tìm kiếm thông tin, đặt hàng và thanh toán trực tuyến. Một trang web giúp cho nhà hàng, chuỗi cửa hàng có thể giới thiệu các sản phẩm, dịch vụ của mình đến với khách hàng một cách trực quan và chi tiết. Khách hàng có thể dễ dàng tìm kiếm thông tin về địa chỉ, giá cả, menu, thời gian hoạt động và các chương trình khuyến mãi. Bên cạnh đó, trang web còn giúp cho khách hàng có thể đặt hàng, đặt bàn trước và thanh toán trực tuyến, giúp tiết kiệm thời gian và nâng cao trải nghiệm của khách hàng. Do đó, việc sở hữu một trang web giúp cho nhà hàng, chuỗi cửa hàng nâng cao khả năng tiếp cận với khách hàng, tăng doanh số bán hàng và xây dựng được hình ảnh uy tín trong mắt khách hàng.

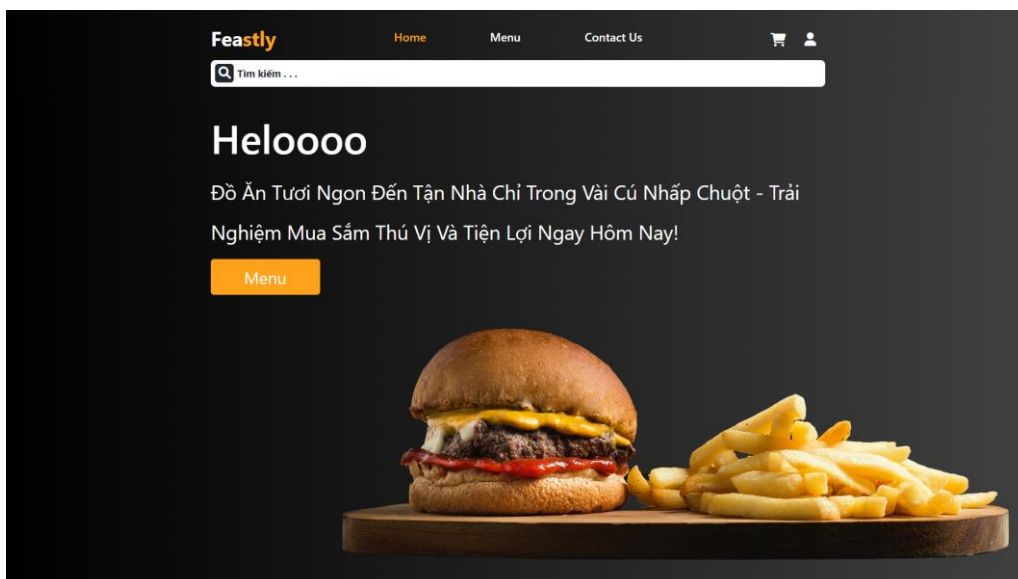
Một số nhà hàng, chuỗi cửa hàng ở Việt Nam có website bán đồ ăn bao gồm Lotteria, KFC, Pizza Hut, The Pizza Company, và Phở 24h. Tuy nhiên, việc sở hữu một trang web cũng có một số hạn chế. Trong khi nhiều nhà hàng, chuỗi cửa hàng đã nhận ra tầm quan trọng của việc sở hữu một trang web, thì vẫn có một số doanh nghiệp chưa hiểu rõ về vai trò của trang web và không đầu tư đầy đủ để phát triển trang web của mình. Điều này có thể dẫn đến một số hạn chế như thiết kế trang web không đẹp mắt, tính năng đặt hàng trực tuyến chưa phát triển hoàn chỉnh và thông tin sản phẩm chưa được cập nhật đầy đủ và chính xác.

Hạn chế đầu tiên là thiết kế trang web không đẹp mắt và chưa tương thích với các thiết bị di động. Một trang web không có thiết kế tốt sẽ làm giảm trải nghiệm của người dùng và làm giảm khả năng thu hút khách hàng đến với doanh nghiệp của bạn. Để giải quyết vấn đề này, dự án này sẽ sử dụng các công nghệ thiết kế hiện đại để tạo ra một trang web đẹp mắt, dễ sử dụng và tương thích với các thiết bị di động.

Hạn chế về tính năng đặt hàng trực tuyến cũng là một trong những vấn đề phổ biến khi phát triển trang web cho nhà hàng, chuỗi cửa hàng. Nếu tính năng đặt hàng trực tuyến chưa phát triển hoàn chỉnh, khách hàng sẽ gặp khó khăn trong việc đặt hàng trực tuyến, từ đó giảm tính tiện lợi của trang web. Với mục tiêu tăng cường tính năng đặt hàng trực tuyến, hệ thống cung cấp các tính năng đặt hàng và thanh toán trực tuyến an toàn và dễ sử dụng, hỗ trợ khách hàng có trải nghiệm mua sắm trực tuyến thuận tiện và an toàn.

Hạn chế về thông tin sản phẩm cũng là một vấn đề cần được giải quyết khi phát triển trang web cho nhà hàng, chuỗi cửa hàng. Nếu thông tin sản phẩm không được cập nhật đầy đủ và chính xác thì khách hàng sẽ không tin tưởng và có thể từ chối mua hàng. Để giải quyết vấn đề này, hệ thống cũng cung cấp các công cụ quản lý thông tin sản phẩm cho nhân viên quản lý, giúp cho việc quản lý và cập nhật thông tin sản phẩm trở nên dễ dàng và tiện lợi hơn.

Tổng kết lại, chúng ta thấy rõ ràng giá trị của việc xây dựng một trang web bán đồ ăn trực tuyến. Nhóm chúng em đã phát triển xây dựng website bán đồ ăn và đặt tên thương hiệu là “Feastly”. Dưới đây là kết quả thu được



Hình 1: Giao diện trang chủ

2 Bố cục báo cáo

Phần còn lại của báo cáo được trình bày như sau:

- **Chương 2:** Mô tả kiến thức nền tảng sẽ giới thiệu tổng quan về thương mại điện tử và các công nghệ, công cụ được sử dụng trong hệ thống. Ngoài ra, chương này cũng sẽ cung cấp thông tin về cách công nghệ và công cụ nêu trên được sử dụng để xây dựng các ứng dụng web hiện đại.
- **Chương 3:** Phân tích và đặc tả yêu cầu sẽ tập trung vào các yêu cầu của hệ thống, bao gồm các yêu cầu chức năng và phi chức năng. Chương này sẽ xác định các use case, các kịch bản sử dụng và các đặc tả yêu cầu khác.
- **Chương 4:** Phân tích thiết kế và xây dựng hệ thống sẽ trình bày về kiến trúc tổng thể của hệ thống, bao gồm các thành phần của hệ thống, mối quan hệ giữa các thành phần và các giao diện giữa chúng.
- **Chương 5:** Cài đặt và kiểm thử sẽ giới thiệu về quá trình triển khai hệ thống, bao gồm các bước cài đặt và cấu hình hệ thống. Chương này cũng sẽ trình bày về các kỹ thuật kiểm thử được sử dụng để đảm bảo tính hoạt động và độ tin cậy của hệ thống.
- **Chương 6:** Kết luận và định hướng phát triển sẽ đánh giá hiệu quả của hệ thống và đưa ra những đề xuất phát triển tiếp theo.

Chương 2: Mô tả kiến thức nền tảng

1 Tổng quan về thương mại điện tử



Hình 2: Thương mại điện tử

1.1 Khái niệm về thương mại điện tử

Thương mại điện tử hay còn gọi là Ecommerce/electronic commerce là hoạt động mua bán hàng hóa và dịch vụ trên Internet. Ước tính có khoảng 2,14 tỷ người trên toàn thế giới mua hàng hóa và dịch vụ trực tuyến trong năm 2022, và số lượng người mua hàng vẫn đang ngày một tăng lên.

Việc giao dịch, bán hàng trên sàn thương mại điện tử là một trong nhiều phương thức giúp đỡ cả người mua lẫn người bán các mặt hàng. Thay vì phải di chuyển đến các cửa hàng thì người mua có thể ở nhà và đặt nhiều sản phẩm mà mình yêu thích. Đối với nhiều doanh nghiệp thì thương mại điện tử là kênh thu lợi nhuận chính, giúp họ quảng bá các mặt hàng trên quy mô lớn và tiếp cận khách hàng trên toàn thế giới.

Để tương tác giữa người mua và người bán thì trang web thương mại điện tử chính là gian hàng trung gian trên Internet, giúp người mua và người bán có thể giao dịch với nhau. Đối với người bán, trang web là nơi họ có thể đẩy lên các mặt hàng với các thông tin chi tiết, số lượng, hình ảnh cũng như giá cả của sản phẩm. Trang web sẽ đóng vai trò là kệ sản phẩm, nhân viên bán hàng kiêm cả quầy thu ngân của kênh bán hàng trực tuyến của bạn. Đối với khách hàng, họ có thể thoải mái tìm kiếm đa dạng nhiều mặt hàng, đọc các thông tin liên quan và so sánh giá cả giữa các mặt hàng với nhau. Người mua chỉ cần vào các trang web thương mại điện tử trên các thiết bị điện thoại, máy tính ... đã có thể chọn và mua các mặt hàng, tiết kiệm được thời gian, các chi phí đi lại.

Các doanh nghiệp có thể tạo ra gian hàng mang thương hiệu của mình trên Amazon,

hoặc xây dựng trang web thương mại của riêng họ với một tên miền riêng, hoặc tiếp cận đa kênh bằng cách thực hiện cả hai cách.

1.2 Các mô hình thương mại điện tử

Có rất nhiều cách để mua bán trực tuyến, nên thương mại điện tử cũng có nhiều dạng khác nhau. Một số mô hình kinh doanh phổ biến trong thế giới thương mại điện tử là:

- **Business-to-consumer (B2C)** - Doanh nghiệp bán cho người tiêu dùng cá nhân (người dùng cuối). Thuật ngữ này phổ biến vào thời kỳ bùng nổ dot-com vào cuối những năm 90, thời kỳ mà bán lẻ và bán hàng trực tuyến các mặt hàng còn mới lạ.
- **Business-to-business (B2B)** - Việc trao đổi điện tử các sản phẩm, dịch vụ hoặc thông tin giữa hai hoặc nhiều doanh nghiệp. Thay vì bán hàng trực tiếp cho người tiêu dùng cuối, các doanh nghiệp B2B tập trung vào việc cung cấp sản phẩm, dịch vụ hoặc giải pháp cho các công ty, tổ chức hoặc doanh nghiệp khác.
- **Consumer-to-consumer (C2C)** - Hình thức kinh doanh trong đó các giao dịch thương mại xảy ra trực tiếp giữa các cá nhân, người tiêu dùng cuối cùng. Trong mô hình này, người tiêu dùng sẽ mua và bán hàng hóa, sản phẩm hoặc dịch vụ cho nhau thông qua các nền tảng trực tuyến hoặc offline.
- **Consumer-to-business (C2B)** - Người tiêu dùng tạo ra các sản phẩm, giá trị và bán cho doanh nghiệp. Ví dụ mô hình C2B phổ biến là người dùng bán các bức tranh, tệp phim, các thiết kế, ... trên các nền tảng trực tuyến.
- **Business-to-government (B2G)** - Doanh nghiệp cung cấp sản phẩm, dịch vụ hoặc giải pháp cho các cơ quan, tổ chức hoặc chính phủ. Trong mô hình này, chính phủ được coi là khách hàng hoặc đối tác kinh doanh.
- **Consumer-to-government (C2G)** - Các cá nhân, người tiêu dùng tương tác và giao dịch trực tiếp với chính phủ. Ví dụ về mô hình này là khi người dân đăng ký sử dụng dịch vụ công hoặc nộp thuế trực tiếp cho các cơ quan chính phủ.
- **Government-to-business (G2B)** - Chính phủ hoặc cơ quan chính phủ giao dịch với các doanh nghiệp. Chính phủ cung cấp các dịch vụ, thông tin hoặc cung cấp hợp đồng công cộng cho các doanh nghiệp
- **Government-to-consumer (G2C)** - Đây là mô hình chính phủ hoặc cơ quan chính phủ bán các sản phẩm, dịch vụ cho người tiêu dùng. Chính phủ cung cấp các dịch vụ công cộng, thông tin, hoặc hỗ trợ cho người dân.

Trong các mô hình kể trên, B2C là mô hình phổ biến và đa dạng nhất, đóng góp cho sự phát triển của thương mại điện tử trong thời đại số hiện nay.

1.3 Lợi ích của thương mại điện tử

Thương mại điện tử là một lĩnh vực quan trọng đang phát triển rất nhanh chóng trên

toàn cầu. Với sự gia tăng của Internet và sự phát triển của các công nghệ, mạng xã hội lớn thì thương mại điện tử trở thành một phần không thể thiếu trong đời sống của chúng ta. Các công ty lớn như Amazon, Alibaba, eBay, Lazada hay Shopee, ... đã chứng minh được tiềm năng và tầm quan trọng của thương mại điện tử trong kinh doanh toàn cầu. Một số lợi ích hàng đầu của việc bán hàng trực tuyến trên sàn thương mại điện tử có thể nhắc đến:

- **Tính sẵn có:** Ngoài các tình huống gián đoạn hoặc bảo trì định kỳ, các trang web thương mại điện tử luôn hoạt động 24/7, cho phép khách hàng truy cập và mua sắm bất cứ lúc nào.
- **Tốc độ truy cập:** Việc mua sắm trực tuyến trên các trang web thương mại điện tử cho phép khách hàng truy cập một cách nhanh chóng và thuận tiện. Thay vì phải đến tận nơi các cửa hàng vật lý thì khách hàng có thể duyệt qua các sản phẩm và thực hiện giao dịch chỉ trong vài phút. Việc này giúp tiết kiệm thời gian và năng lượng cho khách hàng.
- **Đa dạng sản phẩm:** Các trang web thương mại điện tử cung cấp một loạt sản phẩm rộng lớn và phong phú từ nhiều nhà cung cấp khác nhau. Khách hàng có thể dễ dàng tìm kiếm những sản phẩm mà họ muốn.
- **Dễ dàng tiếp cận:** Với thương mại điện tử, việc tiếp cận các sản phẩm và dịch vụ trở nên dễ dàng hơn bao giờ hết. Khách hàng có thể truy cập vào các trang web mua sắm từ bất kỳ địa điểm nào, bất kể thời gian hay khoảng cách địa lý. Điều này giúp mở rộng thị trường tiềm năng và mang lại cơ hội cho các doanh nghiệp tiếp cận khách hàng mới.
- **Tiếp cận môi trường quốc tế:** Các doanh nghiệp có thể tiếp cận thị trường quốc tế một cách dễ dàng. Với sự kết nối của Internet, doanh nghiệp có thể giao dịch với khách hàng trên khắp thế giới.
- **Chi phí thấp:** So với việc vận hành các cửa hàng truyền thống, thương mại điện tử giúp giảm chi phí thuê mặt bằng, chi phí nhân viên và lưu trữ hàng hóa. Hơn nữa, các hoạt động thương mại điện tử có thể tự động hóa, giảm thiểu công sức của con người và chi phí để vận hành.
- **Cá nhân hóa và gợi ý sản phẩm:** Các trang web thương mại điện tử có thể thu thập thông tin về sở thích và hành vi mua sắm của khách hàng, từ đó đưa ra gợi ý các sản phẩm phù hợp. Do đó, các khách hàng khi mua sắm trực tuyến trên Internet sẽ có trải nghiệm mua sắm tốt hơn.

2 Trình bày tổng quan về MEVN stack



Hình 3: MEVN stack thông thường

2.1 Giới thiệu

MEVN stack là một bộ công nghệ web phổ biến được sử dụng trong phát triển ứng dụng web đa nền tảng. Cụm từ MEVN là các thuật ngữ rút gọn của MongoDB(MySQL), ExpressJS, VueJS, NodeJS. Mỗi công nghệ đảm nhận một nhiệm vụ cụ thể, tuân theo kiến trúc ba tầng truyền thống, giúp tách biệt các trách nhiệm và logic trong việc xây dựng website, tạo ra sự linh hoạt và dễ bảo trì.

Đầu tiên, tầng giao diện người dùng (Presentation Layer) là tầng mà người dùng tương tác trực tiếp thông qua trình duyệt web. Tầng này do VueJS đảm nhận, kết hợp với các ngôn ngữ lập trình web như HTML, CSS và JavaScript... Tiếp theo là tầng ứng dụng (Application Layer) xử lý chính bởi Express.js và Node.js, tầng này chứa các logic và quy tắc xử lý dữ liệu và nghiệp vụ của ứng dụng. Cuối cùng, sử dụng MySQL ở tầng dữ liệu (Data Layer), chịu trách nhiệm quản lý và lưu trữ dữ liệu của ứng dụng, đảm bảo tính toàn vẹn và an toàn của dữ liệu.

Việc sử dụng MEVN stack với bộ các công nghệ phổ biến và tương thích với nhau giúp tăng tính ổn định và hiệu suất của ứng dụng. Đồng thời, MEVN stack cũng có thể giảm đáng kể thời gian phát triển ứng dụng và chi phí phát triển.

2.2 VueJS

Vue phiên âm đọc giống như từ view, được tạo ra bởi Evan You, một cựu nhân viên và lập trình viên của Google. Phiên bản đầu tiên của Vue được phát hành vào tháng 2 năm 2014.

VueJS là một framework mã nguồn mở của JavaScript, chủ yếu xây dựng giao diện người dùng. Nó là một trong những framework nổi tiếng được sử dụng để đơn giản hóa việc phát triển web. Không giống như các framework khác, Vue được xây dựng từ những dòng code

đơn giản nhất nhằm tối ưu hóa tốc độ. Hơn nữa, thư viện của Vue chỉ tập trung vào lớp hiển thị, rất đơn giản đối với người mới tiếp cận và dễ dàng tích hợp với nhiều hệ thống khác nhau.

Giống như các framework javascript khác như React, Ember, Angular, ... được sử dụng để xây dựng tầng giao diện người dùng thì VueJS có tính năng nổi trội khi có tốc độ tạo trang nhanh và chiếm khá ít bộ nhớ. Tập zip được tải xuống của framework này chỉ chiếm 18 KB, giúp nó cài đặt nhanh chóng mà còn tác động tích cực đến trải nghiệm của người dùng. Hơn nữa, VueJS còn sử dụng **DOM** ảo, giúp tìm các các phần tử cập nhật mà không cần truy xuất đến toàn bộ document, giúp cải thiện tốc độ và hiệu suất ứng dụng.

Tuy VueJS còn là framework khá mới nhưng dần ngày càng phát triển và khẳng định tầm ảnh hưởng của mình trong cộng đồng phát triển phần mềm.

2.3 NodeJS

NodeJS được phát triển bởi Ryan Dahl và lần đầu giới thiệu vào năm 2009. Vào thời điểm đó, các máy chủ gặp khó khăn trong việc xử lý các kết nối đồng thời có khối lượng lớn và mã chặn toàn bộ quá trình hoặc ngụ ý cần có nhiều stack. Mục tiêu ban đầu, Nodejs được tạo ra để cung cấp cho các nhà phát triển khả năng sử dụng JavaScript để tạo script phía máy chủ và thống nhất việc phát triển ứng dụng.

NodeJS là một môi trường chạy mã JavaScript phía máy chủ được xây dựng dựa trên engine JavaScript V8 của Google Chrome. Nó sử dụng mô hình không đồng bộ và sự kiện, giúp xử lý các yêu cầu đồng thời một cách hiệu quả.

Nhiều trang web sử dụng NodeJS như bộ khung chính trong tầng ứng dụng, bởi một số các ưu điểm nổi bật như:

1. Mã chung: Node.js sử dụng cùng một ngôn ngữ (JavaScript) cả phía máy chủ và phía client, giúp đơn giản hóa việc chia sẻ mã và tái sử dụng mã giữa các phần của ứng dụng.
2. Độ linh hoạt: Node.js cho phép xây dựng các ứng dụng web đa nền tảng, từ ứng dụng web truyền thống đến các ứng dụng real-time, streaming và microservices.
3. Hệ sinh thái mạnh mẽ: Node.js có một hệ sinh thái phong phú của các thư viện và công cụ phát triển. Cộng đồng Node.js rất lớn và nhiệt tình, cung cấp nhiều module và gói thư viện mở rộng để giúp phát triển ứng dụng một cách nhanh chóng và tiện lợi.
4. Tiết kiệm tài nguyên: Node.js sử dụng một luồng duy nhất để xử lý các yêu cầu, giúp giảm thiểu overhead của việc tạo và duy trì hệ thống. Hơn nữa, Node.js sử dụng mô hình non-blocking I/O, cho phép xử lý đồng thời nhiều yêu cầu mà không chờ đợi hoàn thành. Điều này giúp tối ưu hóa sử dụng tài nguyên và tăng cường hiệu suất.

Lập trình với NodeJS khá đơn giản và dễ tiếp cận, hỗ trợ đa dạng nhiều framework với những tính năng nổi trội. Điều này khiến NodeJS dần ngày càng trở nên phổ biến và thu hút đông đảo lượng lớn người sử dụng.

2.4 *ExpressJS*

ExpressJS lần đầu tiên phát hành vào năm 2010, được tạo ra bởi TJ Holowaychuk, một nhà phát triển phần mềm nổi tiếng trong cộng đồng Node.js và JavaScript. Ông lấy cảm hứng từ framework web Sinatra, một framework phát triển ứng dụng web nhỏ gọn và đơn giản được viết bằng Ruby để xây dựng Express.js như một lớp mô phỏng lại giao diện điều khiển (controller) của Sinatra bằng JavaScript. Từ đó, cung cấp một cách tiếp cận tương tự và dễ dàng sử dụng cho ngôn ngữ JavaScript và Node.js.

ExpressJS phát triển ứng dụng web phía máy chủ (Server-side), cung cấp các công cụ và tính năng cần thiết để xử lý các yêu cầu HTTP, xây dựng các route (đường dẫn) và xử lý middleware (các chức năng trung gian) để thực hiện các chức năng như xác thực, ghi nhật ký, xử lý lỗi và nhiều hơn nữa. Với Express, lập trình viên có thể cấu trúc ứng dụng theo bất kỳ cách nào mong muốn, kết hợp với đa dạng nhiều loại cơ sở dữ liệu như MySQL, MongoDB, PostgreSQL, ... dùng với các công cụ template engine như ejs, handlebars, pug, ...

Trong MEVN stack, Express đóng vai trò quan trọng trong tầng ứng dụng, xử lý các yêu cầu và trả về các kết quả cho người dùng. Nó tương thích dễ dàng với các modules khác của NodeJS như express-session, sequelize, redis, ... Hơn nữa, cấu trúc của Express khá đơn giản và dễ sử dụng cũng là một điểm cộng khiến cho framework được rất nhiều lập trình viên ưa chuộng và sử dụng.

2.5 *MySQL, MongoDB, Redis*

MySQL được phát triển vào năm 1994, là một hệ thống quản trị cơ sở dữ liệu quan hệ mã nguồn mở hoạt động theo mô hình client-server. Về nguyên tắc hoạt động, MySQL nhận các truy vấn từ người dùng và trả về kết quả như mong muốn. Nó cung cấp các câu lệnh đa dạng để kết nối và thao tác trên cơ sở dữ liệu như chỉnh sửa các quan hệ giữa các bảng, thêm, xóa, sửa các đối tượng. Hơn nữa, nó cho phép quản lý các cơ sở dữ liệu phức tạp có liên hệ chặt chẽ với nhau và tuân theo nguyên tắc **ACID model** (Atomicity, Consistency, Isolation, Durability) để đảm bảo tính toàn vẹn và cung cấp các cơ chế bảo mật mạnh mẽ.

MongoDB được phát triển bởi Dwight Merriman và Eliot Horowitz vào năm 2007 tại công ty DoubleClick nay thuộc Google. Mục tiêu của MongoDB là cung cấp một hệ quản trị cơ sở dữ liệu phi tập trung (NoSQL), linh hoạt và có thể thay đổi theo thời gian một cách dễ dàng. Nó được thiết kế theo kiểu hướng đối tượng, các đối tượng dữ liệu được lưu trữ dưới dạng các tài liệu riêng biệt bên trong một collection, thay vì lưu vào cột và hàng như MySQL. MongoDB sinh ra để tăng tốc độ truy xuất dữ liệu và phản hồi nhanh tuy nhiên điều này dẫn đến nhiều nhược điểm mà nó không thể tránh khỏi. Các tác nghiệp cần tính toàn vẹn dữ liệu thì MySQL luôn là một giải pháp tối ưu hơn hay MongoDB không có tính ràng buộc nên các quan hệ giữa các collection không được tường minh, rõ ràng.

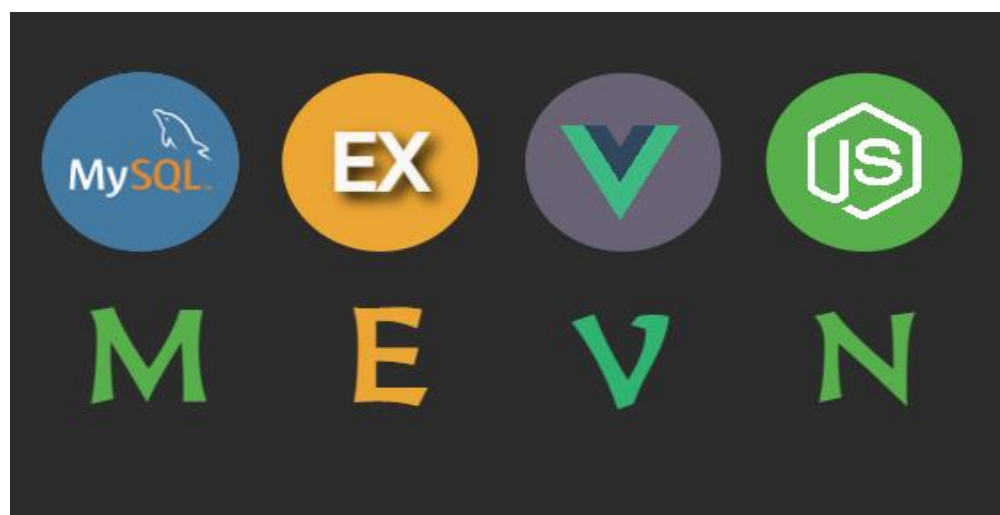
Giữa hệ quản trị cơ sở dữ liệu MySQL và MongoDB thì khó có thể so sánh hệ quản trị nào tối ưu hơn, tùy vào từng môi trường và yêu cầu cụ thể. Tuy nhiên, dựa theo một số tiêu chí dự án về website bán đồ ăn do chúng em đưa ra:

- Tính rõ ràng trong cấu trúc thiết kế cơ sở dữ liệu.
- Khả năng truy vấn phức tạp.
- Sử dụng ACID đảm bảo tính nhất quán và độ tin cậy cao của dữ liệu.
- Hệ quản trị quen thuộc, gần gũi và dễ dàng tiếp cận.

Qua những tiêu chí kể trên, chúng em quyết định **thay thế công nghệ MongoDB trong MEVN stack bằng hệ quản trị MySQL** để xử lý và lưu trữ cơ sở dữ liệu. Ngoài ra, để tăng tính bảo mật và đảm bảo an toàn dữ liệu, tránh việc phải viết lại các câu lệnh truy vấn SQL cơ bản thì NodeJS cung cấp cho người dùng module **Sequelize**. Sequelize là một **ORM** (Object-Relational Mapping) dành cho MySQL, Postgres, MariaDB, SQLite, ...giúp đơn giản hóa quá trình làm việc với cơ sở dữ liệu và cung cấp các chức năng như tạo bảng, tạo mối quan hệ, truy vấn dữ liệu và thực hiện các thao tác cập nhật.

Bên cạnh đó, để xây dựng website bán đồ ăn đáp ứng với lượng người dùng lớn, thời gian truy xuất và xác thực người dùng nhanh thì sử dụng thêm **hệ quản trị cơ sở dữ liệu Redis** sẽ tối ưu những tiêu chí này. Khi một website cần phải đăng nhập, đăng xuất, mua bán và thao tác với giỏ hàng, sẽ sử dụng **session và cookie** để lưu trữ thông tin của khách hàng. Việc một website có lượng người dùng lớn, truy cập cùng lúc đồng thời, dùng MySQL để lưu trữ session của người dùng sẽ làm giảm hiệu suất và gặp khó khăn trong việc mở rộng khi tải lớn và đòi hỏi quản lý cụm cơ sở dữ liệu phức tạp hơn. Chính vì vậy, Redis được thiết kế lưu dữ liệu trực tiếp trong bộ nhớ, phân tán và dễ dàng mở rộng trở thành một giải pháp lý tưởng cho việc lưu trữ session và các tác vụ khác của ứng dụng web.

3 Tổng kết chương



Hình 4: MEVN stack thay thế, sử dụng trong dự án

Qua chương này, ta có thể hiểu thêm về tầm quan trọng của thương mại điện tử, các mô hình hoạt động và những lợi ích to lớn mà nó đem lại. Qua đó, chúng em cũng giới thiệu về

MEVN stack bao gồm MySQL, ExpressJS, VueJS, NodeJS, bộ 4 công nghệ được chúng em xây dựng trong sản phẩm của dự án website bán đồ ăn. Ngoài ra, với một số tiêu chí sản phẩm của chúng em đề ra, độc giả có thể hiểu được vì sao hệ quản trị cơ sở dữ liệu trong MEVN stack chúng em dùng là MySQL thay cho MongoDB, khác với các websites thông thường. Bên cạnh đó, để nâng cao hiệu suất và tăng trải nghiệm của người dùng, chúng em giới thiệu về việc sử dụng thêm hệ quản trị cơ sở dữ liệu Redis, như một hệ quản trị trung gian để lưu trữ session của người dùng và thực hiện các tác vụ khác.

Chương 3: Phân tích và đặc tả yêu cầu

1 Phân tích yêu cầu

1.1 Yêu cầu chức năng

Đăng ký, đăng nhập: Cho phép khách hàng đăng ký và đăng nhập vào tài khoản của mình để đặt hàng và quản lý thông tin cá nhân.

Xem danh sách sản phẩm: Khi người dùng truy cập vào trang chủ website thì hệ thống sẽ hiển thị các sản phẩm nổi bật và các danh mục sản phẩm. Khi người dùng chọn vào 1 danh mục sản phẩm cụ thể thì tất cả sản phẩm của danh mục đó sẽ hiển thị lên giao diện.

Tìm kiếm sản phẩm: Cung cấp công cụ tìm kiếm để khách hàng có thể tìm kiếm sản phẩm 1 cách nhanh chóng và dễ dàng.

Thêm vào giỏ hàng: Sau khi khách hàng tìm thấy sản phẩm đáp ứng nhu cầu thì có thể chọn đưa sản phẩm vào giỏ hàng để mua sau hoặc mua ngay.

Mua hàng: Cung cấp giao diện đặt hàng đơn giản và dễ sử dụng cho phép khách hàng cập nhật số lượng sản phẩm trong giỏ hàng.

Thanh toán: Người dùng có thể thanh toán các sản phẩm đã chọn thông qua phương thức thanh toán trực tuyến hoặc thanh toán khi nhận hàng.

Xem lịch sử đơn mua: Sau khi đặt mua, khách hàng có thể xem lại đơn hàng đã đặt và trạng thái của đơn.

Quản lý đơn hàng: Cung cấp tính năng quản lý đơn hàng cho quản trị để theo dõi, thay đổi trạng thái đơn hàng và thông tin vận chuyển.

Quản lý sản phẩm: Cho phép quản trị quản lý được tất cả sản phẩm của cửa hàng. Có thể thêm, xóa sản phẩm. Có thể cập nhật thông tin sản phẩm về giá cả, mô tả sản phẩm, hình ảnh và giảm giá để cung cấp cho khách hàng thông tin chính xác và đầy đủ.

1.2 Yêu cầu phi chức năng

Về nội dung:

- Thông tin về sản phẩm phải được đồng nhất, liên kết xuyên suốt toàn bộ website, các dữ liệu chuẩn và chính xác về mặt thông tin.
- Hiển thị đa dạng các danh mục, các hàng được giảm giá. Giúp tăng cường trải nghiệm người dùng.

Về giao diện người dùng:

- Phải có 1 giao diện thân thiện dễ sử dụng, hỗ trợ đa thiết bị.
- Khách hàng có thể dễ dàng sử dụng được hệ thống ngay trong lần truy cập đầu tiên vào website.
- Các chức năng rõ ràng.

Về giao diện quản trị:

- Đơn giản, dễ quản lý và không thể thiếu các mục như: quản lý sản phẩm, quản

lý người dùng, quản lý đơn hàng.

Về bảo mật:

- Mật khẩu tài khoản của người dùng phải được mã hóa trong cơ sở dữ liệu.

2 Phân tích và đặc tả ca sử dụng



Hình 5: Ca sử dụng chung của hệ thống

2.1 Ca sử dụng “Đăng ký tài khoản”

Mô tả: Người dùng đăng ký tạo tài khoản bằng email cá nhân để sử dụng các chức năng của hệ thống.

Tác nhân: Người dùng chưa có tài khoản

Luồng sự kiện chính:

1. Người dùng đăng nhập các trường thông tin trên màn hình đăng ký tài khoản bao

gồm:

- a. Tên.
- b. Email.
- c. Mật khẩu.
- d. Xác nhận mật khẩu.
- e. Số điện thoại.

2. Người dùng gửi biểu mẫu đăng ký khi điền đầy đủ thông tin.

3. Hệ thống xử lý yêu cầu hoàn tất và điều hướng sang màn hình chính.

Luồng sự kiện rẽ nhánh:

- Nếu thông tin đăng ký không chính xác hoặc không đầy đủ, hệ thống sẽ yêu cầu người dùng nhập lại thông tin.
- Nếu nhập email đã được sử dụng để đăng ký, hệ thống sẽ hiển thị thông báo lỗi

Các yêu cầu đặc biệt: Hệ thống phải xác minh thông tin của người dùng và ngăn chặn việc tạo nhiều tài khoản trùng lặp.

Tiền điều kiện: Người dùng chưa có tài khoản trên hệ thống và phải truy cập vào trang đăng ký tài khoản trên hệ thống.

Điểm mở rộng: Không có.

2.2 *Ca sử dụng “Đăng nhập tài khoản”*

Mô tả: Người dùng đăng nhập vào hệ thống bằng tài khoản người dùng đã đăng ký. Quản trị đăng nhập bằng tài khoản hệ thống cung cấp.

Tác nhân: Người dùng, quản trị.

Luồng sự kiện chính:

1. Người dùng nhập tên đăng nhập và mật khẩu.
2. Hệ thống kiểm tra thông tin đăng nhập của người dùng và chấp nhận hoặc từ chối việc đăng nhập.
3. Nếu thông tin đăng nhập đúng, hệ thống đưa người dùng đến trang chính hoặc trang mà người dùng yêu cầu.

Luồng sự kiện rẽ nhánh:

1. Nếu thông tin đăng nhập không chính xác hoặc không đầy đủ, hệ thống sẽ yêu cầu người dùng nhập lại thông tin.

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Người dùng đã truy cập vào trang đăng nhập và có tài khoản đăng nhập.

Điểm mở rộng: Người dùng sẽ không phải đăng nhập lại trong vòng 1 ngày.

2.3 *Ca sử dụng “Thêm vào giỏ hàng”*

Mô tả: Người dùng tìm kiếm sản phẩm yêu thích và thêm vào giỏ hàng để mua sau hoặc tiếp tục mua sắm trên hệ thống.

Tác nhân: Người dùng.

Luồng sự kiện chính:

1. Trên giao diện danh mục các sản phẩm, người dùng tìm kiếm sản phẩm yêu thích và ấn vào “Thêm vào giỏ hàng”.
2. Hệ thống sẽ lưu trữ thông tin sản phẩm mà người dùng thêm vào giỏ.
3. Người dùng có thể tiếp tục mua sắm hoặc xem giỏ hàng của mình để hoàn tất đơn hàng.

Luồng sự kiện rẽ nhánh: Không có.

Các yêu cầu đặc biệt: Không thể thêm các sản phẩm đã hết hàng.

Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống.

Điểm mở rộng: Tại giao diện giỏ hàng sẽ có các chức năng: xóa giỏ hàng, cập nhật số lượng cho sản phẩm đã đặt, xóa 1 hoặc nhiều sản phẩm, hiển thị tổng số tiền phải thanh toán.

2.4 *Ca sử dụng “Mua hàng”*

Mô tả: Ca sử dụng này cho phép khách hàng mua các sản phẩm trên website.

Tác nhân: Người dùng.

Luồng sự kiện chính:

1. Người dùng truy cập vào giỏ hàng và xác nhận các sản phẩm cần mua.
2. Người dùng ấn nút mua ngay để mua hàng.
3. Hệ thống yêu cầu người dùng cung cấp thông tin địa chỉ giao hàng.
4. Người dùng nhập thông tin địa chỉ giao hàng và phương thức thanh toán.
5. Người dùng ấn nút đặt hàng và hàng sẽ được giao đến địa chỉ.

Luồng sự kiện rẽ nhánh: Nếu thông tin địa chỉ nhận không chính xác hoặc không đầy đủ, hệ thống sẽ yêu cầu người dùng nhập lại thông tin.

Các yêu cầu đặc biệt: Khách hàng có thể lựa chọn thanh toán trực tiếp hoặc thanh toán trực tuyến..

Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống. Các sản phẩm đã được thêm vào giỏ hàng.

Điểm mở rộng: Địa điểm giao hàng đúng và đầy đủ các vị trí trên bản đồ Việt Nam.

2.5 *Ca sử dụng “Chỉnh sửa thông tin cá nhân”*

Mô tả: Ca sử dụng này mô tả quá trình chỉnh sửa thông tin cá nhân của người dùng.

Tác nhân: Người dùng.

Luồng sự kiện chính:

1. Trên giao diện thông tin cá nhân, người dùng chọn các tùy chọn và thay đổi thông tin cá nhân của mình
2. Người dùng lưu thay đổi

3. Hệ thống cập nhật thông tin của người dùng.

Luồng sự kiện rẽ nhánh: Nếu thông tin người dùng nhập không chính xác, hệ thống hiển thị thông báo lỗi và yêu cầu nhập thông tin hợp lệ

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống.

Điểm mở rộng: Không có.

2.6 *Ca sử dụng “Xem trạng thái đơn hàng”*

Mô tả: Ca sử dụng này mô tả quá trình người dùng xem lại các đơn mua và xem chi tiết đơn hàng.

Tác nhân: Người dùng.

Luồng sự kiện chính:

1. Người dùng truy cập vào trang đơn mua của mình trên trang web.
2. Hệ thống hiển thị thông tin và trạng thái các đơn đã mua.
3. Người dùng chọn đơn hàng cần xem chi tiết
4. Hệ thống hiển thị chi tiết đơn hàng bao gồm thông tin sản phẩm, số lượng, giá cả, tổng số tiền, trạng thái vận chuyển.

Luồng sự kiện rẽ nhánh: Nếu không có đơn hàng nào, hệ thống sẽ hiển thị thông báo không có đơn hàng nào để xem.

Các yêu cầu đặc biệt: Người dùng có thể hủy đơn hàng nếu trạng thái đang xử lý.

Các trạng thái thì không thể hủy đơn hàng.

Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống.

Điểm mở rộng: Không có.

2.7 *Ca sử dụng “Tìm kiếm sản phẩm”*

Mô tả: Người dùng nhập từ khóa tìm kiếm và hệ thống hiển thị kết quả tìm kiếm.

Tác nhân: Người dùng.

Luồng sự kiện chính:

1. Người dùng nhập từ khóa tìm kiếm vào ô tìm kiếm trên trang web.
2. Hệ thống tìm kiếm và hiển thị các sản phẩm liên quan tới từ khóa tìm kiếm của người dùng.
3. Người dùng chọn sản phẩm trong kết quả tìm kiếm để xem chi tiết hoặc thêm vào giỏ hàng.

Luồng sự kiện rẽ nhánh: Nếu không có sản phẩm nào liên quan đến từ khóa, hệ thống sẽ không hiển thị sản phẩm nào.

Các yêu cầu đặc biệt: Thêm tính năng lọc sản phẩm theo danh mục, theo mức giá giúp người dùng tìm kiếm nhanh chóng sản phẩm phù hợp.

Tiền điều kiện: Không có.

Điểm mở rộng: Có thể tìm sản phẩm với các từ khóa không dấu hoặc không đầy đủ.

2.8 *Ca sử dụng “Xem thông tin sản phẩm”*

Mô tả: Người dùng chọn sản phẩm và xem thông tin chi tiết của sản phẩm.

Tác nhân: Người dùng.

Luồng sự kiện chính:

1. Người dùng truy cập vào trang danh mục sản phẩm và chọn sản phẩm cần xem chi tiết.
2. Hệ thống hiển thị thông tin chi tiết của sản phẩm bao gồm hình ảnh, giá ban đầu, giá được giả, tên sản phẩm, mô tả sản phẩm.
3. Người dùng đọc thông tin chi tiết của sản phẩm và quyết định mua hoặc không mua sản phẩm đó.

Luồng sự kiện rẽ nhánh: Không có.

Các yêu cầu đặc biệt: Không có.

Tiền điều kiện: Không có.

Điểm mở rộng: Tại giao diện thông tin chi tiết sản phẩm, có thêm chức năng điều chỉnh số lượng trước khi thêm vào giỏ hàng.

2.9 *Ca sử dụng “Thêm sản phẩm”*

Mô tả: Ca sử dụng này mô tả quá trình quản trị thêm sản phẩm mới vào hệ thống.

Tác nhân: Quản trị.

Luồng sự kiện chính:

1. Quản trị truy cập vào trang quản lý sản phẩm và chọn thêm sản phẩm mới.
2. Quản trị nhập thông tin chi tiết về sản phẩm bao gồm hình ảnh, giá, mô tả sản phẩm, giá cả, danh mục và giảm giá ưu đãi.
3. Quản trị ấn nút “Thêm”
4. Hệ thống kiểm tra thông tin và thêm vào hệ thống.

Luồng sự kiện rẽ nhánh: Nếu thông tin sản phẩm không hợp lệ hoặc không đầy đủ, hệ thống sẽ yêu cầu quản trị nhập đầy đủ.

Các yêu cầu đặc biệt: Chỉ có quản trị mới có thể thêm sản phẩm.

Tiền điều kiện: Quản trị đã đăng nhập vào hệ thống.

Điểm mở rộng: Không có.

2.10 *Ca sử dụng “Xóa sản phẩm”*

Mô tả: Ca sử dụng này mô tả quá trình quản trị xóa sản phẩm khỏi hệ thống.

Tác nhân: Quản trị.

Luồng sự kiện chính:

1. Quản trị truy cập vào trang quản lý sản phẩm.
2. Hệ thống hiển thị danh sách các sản phẩm đã có.
3. Quản trị chọn sản phẩm cần xóa và hệ thống sẽ hiển thị hộp thoại xác nhận xóa sản phẩm.
4. Quản trị xác nhận xóa sản phẩm và hệ thống xóa sản phẩm khỏi hệ thống.

Luồng sự kiện rẽ nhánh: Nếu quản trị hủy hành động xóa, hệ thống sẽ tắt hộp thoại, việc xóa sẽ bị hủy. Ca sử dụng được bắt đầu lại từ đầu

Các yêu cầu đặc biệt: Chỉ có quản trị mới có thể xóa sản phẩm.

Tiền điều kiện: Quản trị đã đăng nhập vào hệ thống.

Điểm mở rộng: Không có.

2.11 Ca sử dụng “Xóa người dùng”

Mô tả: Ca sử dụng này mô tả quá trình quản trị xóa người dùng khỏi hệ thống.

Tác nhân: Quản trị.

Luồng sự kiện chính:

1. Quản trị truy cập vào trang quản lý người dùng trên hệ thống.
2. Hệ thống hiển thị danh sách các người dùng đã đăng ký tài khoản.
3. Quản trị chọn người dùng cần xóa và hệ thống sẽ hiển thị hộp thoại xác nhận xóa người dùng.
4. Quản trị xác nhận xóa và hệ thống xóa người dùng khỏi hệ thống.

Luồng sự kiện rẽ nhánh: Nếu quản trị hủy xác nhận xóa thì hệ thống sẽ không làm gì.

Các yêu cầu đặc biệt: Chỉ có quản trị mới có thể xóa người dùng.

Tiền điều kiện: Quản trị đã đăng nhập vào hệ thống.

Điểm mở rộng: Không có.

2.12 Ca sử dụng “Quản lý đơn hàng”

Mô tả: Ca sử dụng này mô tả quá trình quản trị quản lý và cập nhật trạng thái đơn hàng của người dùng.

Tác nhân: Quản trị.

Luồng sự kiện chính:

1. Quản trị truy cập vào trang quản lý đơn hàng trên hệ thống.
2. Hệ thống hiển thị danh sách các đơn hàng đã được đặt và trạng thái từng đơn hàng.
3. Quản trị có thể xem chi tiết từng đơn hàng, bao gồm thông tin khách hàng, sản phẩm được đặt hàng, số lượng, giá cả, trạng thái đơn hàng và thời gian đặt hàng.
4. Quản trị có thể cập nhật trạng thái của các đơn hàng và thông báo cho khách

hàng về trạng thái của đơn hàng. Các trạng thái bao gồm:

- Đang xử lý.
- Đang giao hàng.
- Đã giao hàng.

Luồng sự kiện rẽ nhánh: Không.

Các yêu cầu đặc biệt: Chỉ có quản trị mới có thể quản lý đơn hàng của người dùng.

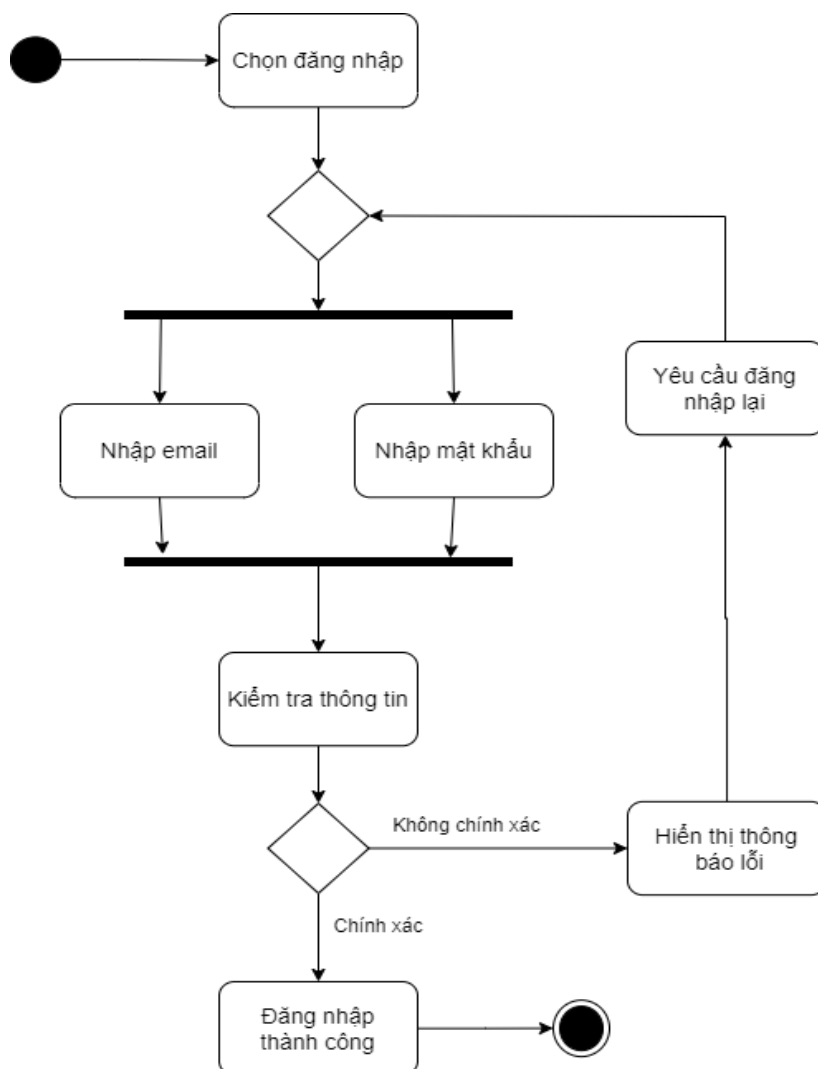
Tiền điều kiện: Quản trị đã đăng nhập vào hệ thống.

Điểm mở rộng: Không có.

3 Mô tả các biểu đồ hoạt động

3.1 Biểu đồ hoạt động “Đăng nhập hệ thống”

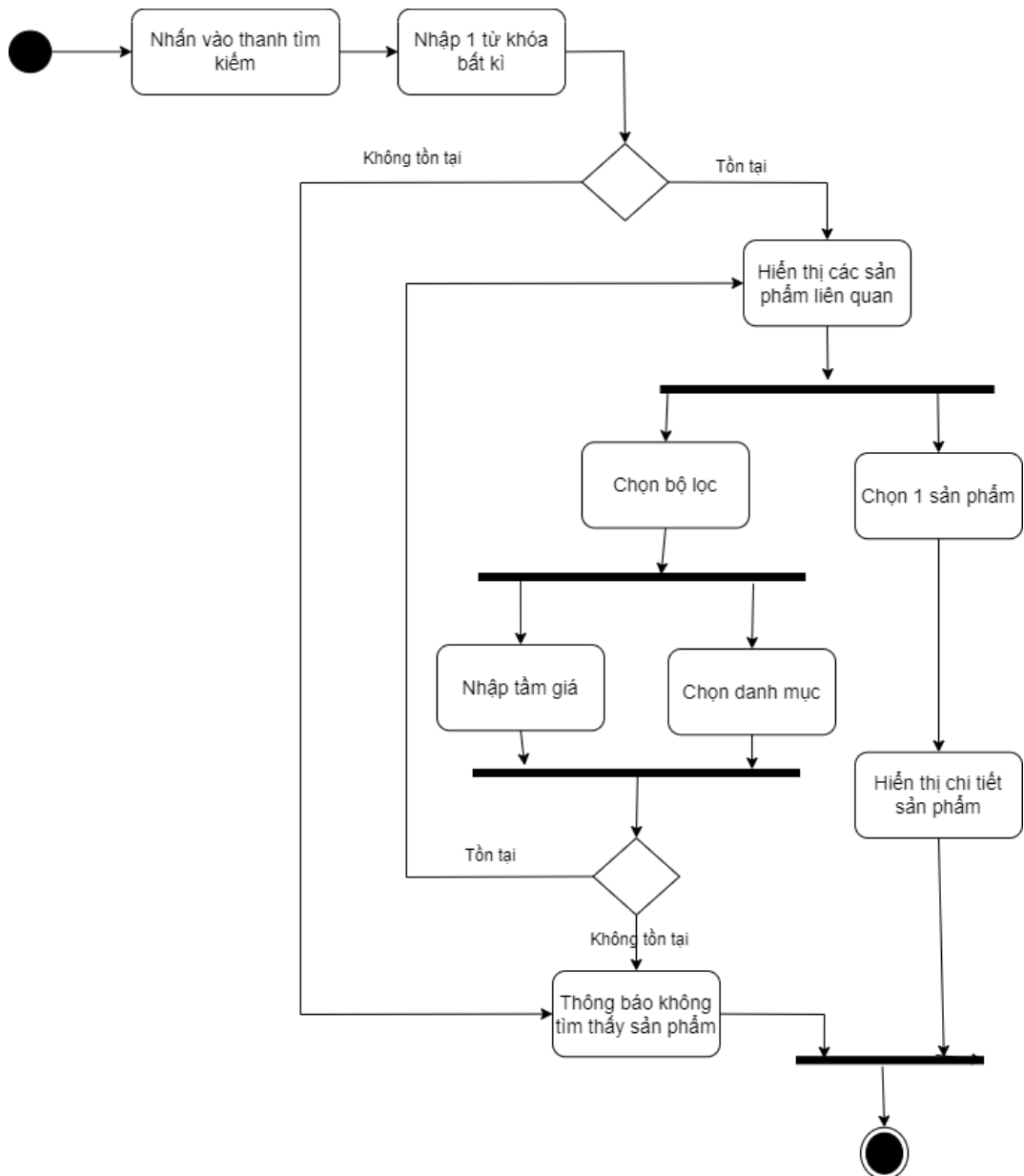
- Đối tượng: Người dùng đã có tài khoản và quản trị.



Hình 6: Biểu đồ hoạt động “Đăng nhập”

3.2 Biểu đồ hoạt động “Tìm kiếm sản phẩm”

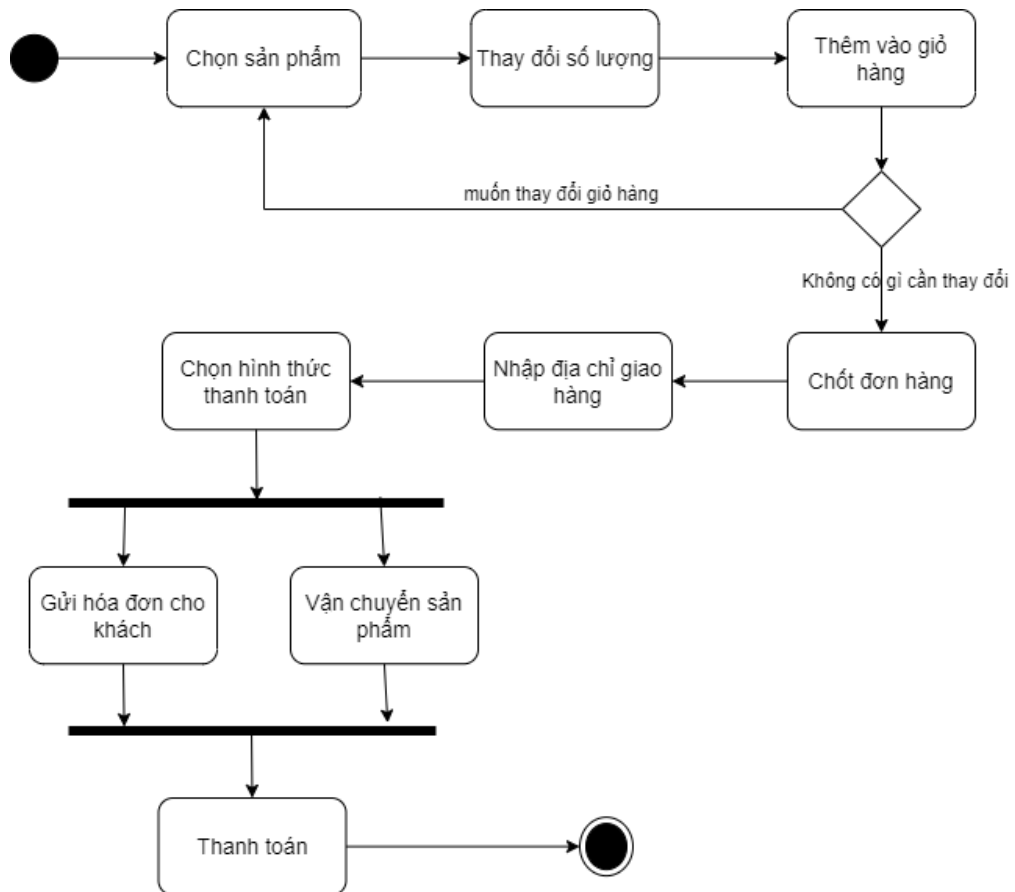
- Đối tượng: Người dùng đã đăng nhập hoặc chưa đăng nhập.



Hình 7: Biểu đồ hoạt động “Tìm kiếm”

3.3 Biểu đồ hoạt động “Đặt hàng”

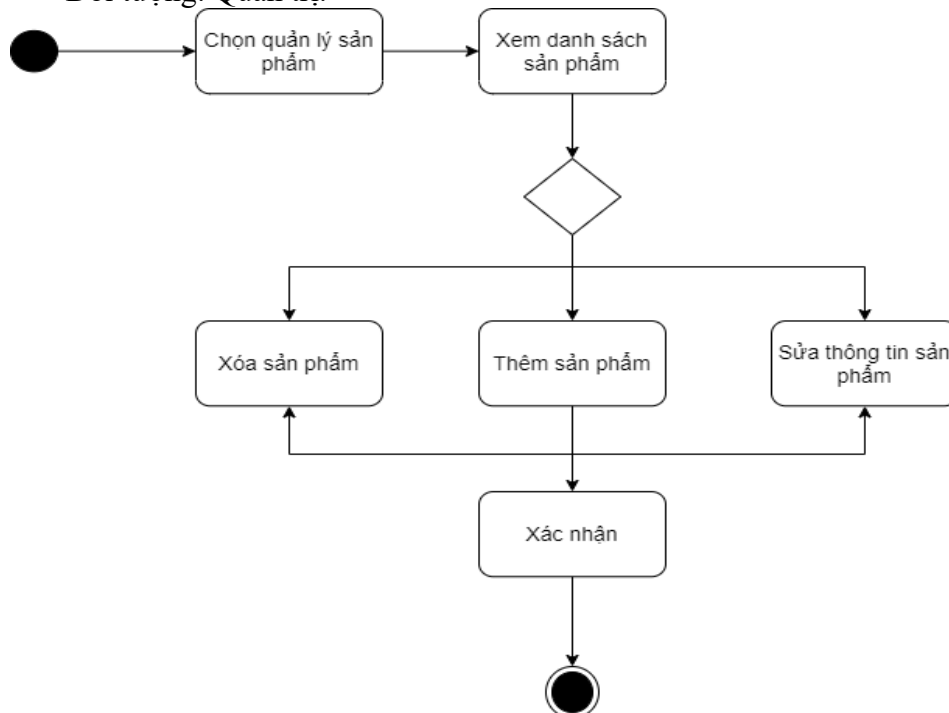
- Đối tượng: Người dùng đã đăng nhập vào hệ thống.



Hình 8: Biểu đồ hoạt động “Mua hàng”

3.4 Biểu đồ hoạt động “Quản lý sản phẩm”

- Đối tượng: Quản trị.

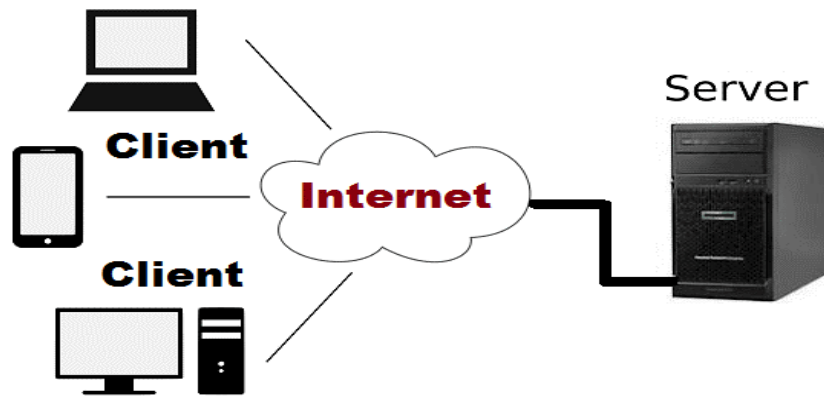


Hình 9: Biểu đồ hoạt động “Quản lý sản phẩm”

Chương 4: Phân tích thiết kế và xây dựng hệ thống

1 Thiết kế kiến trúc chung của hệ thống

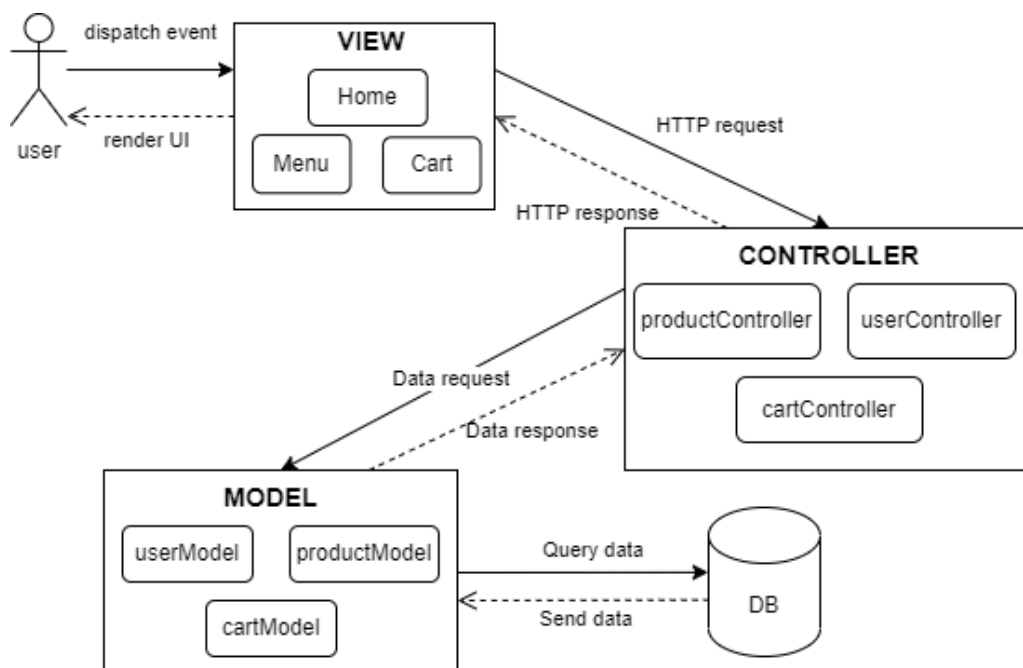
Dự án này được thiết kế theo mô hình client - server khá phổ biến hiện nay cho các ứng dụng web.



Hình 10: Mô hình client - server

Hơn nữa, chúng em xây dựng dự án dựa trên mô hình kiến trúc MVC (Model - View - Controller) là một kiến trúc phần mềm được sử dụng để xây dựng các ứng dụng phần mềm. Nó chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác:

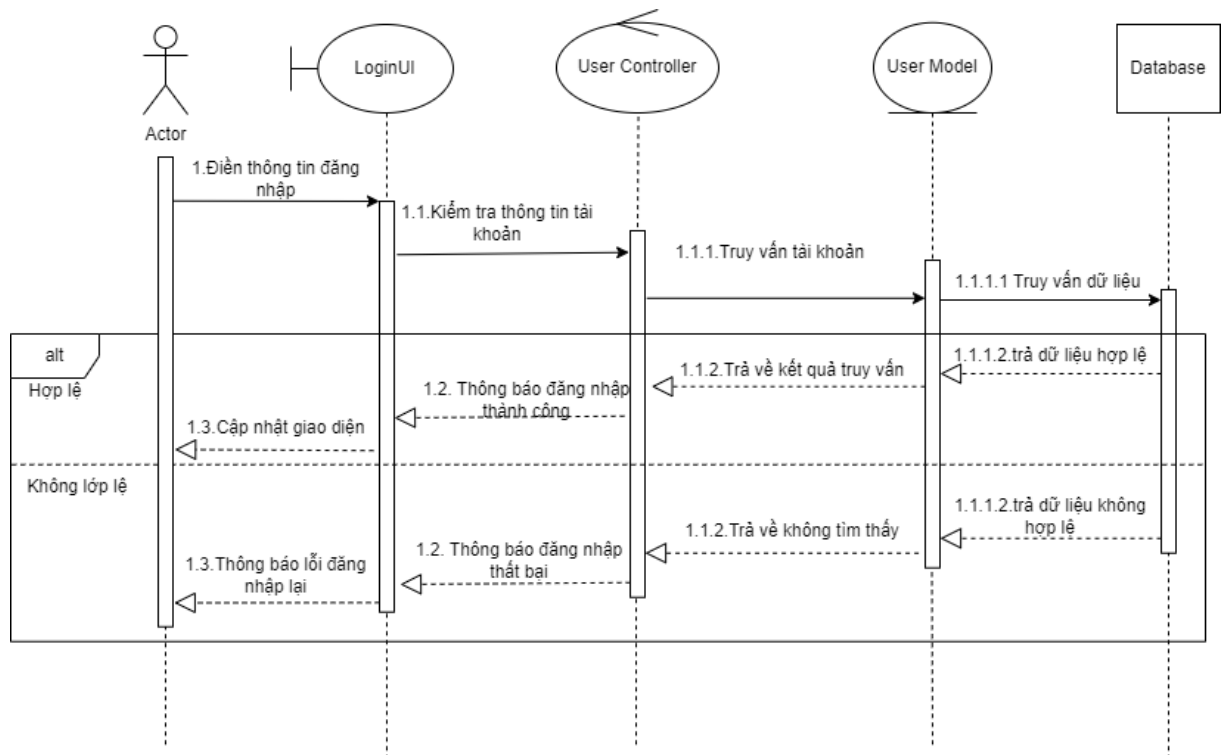
- Model (dữ liệu): Nơi lưu trữ thông tin và xử lý logic liên quan đến dữ liệu.
- View (giao diện): Nơi hiển thị dữ liệu cho người dùng.
- Controller (bộ điều khiển): Điều khiển sự tương tác của hai thành phần Model và View.



Hình 11: Mô hình MVC

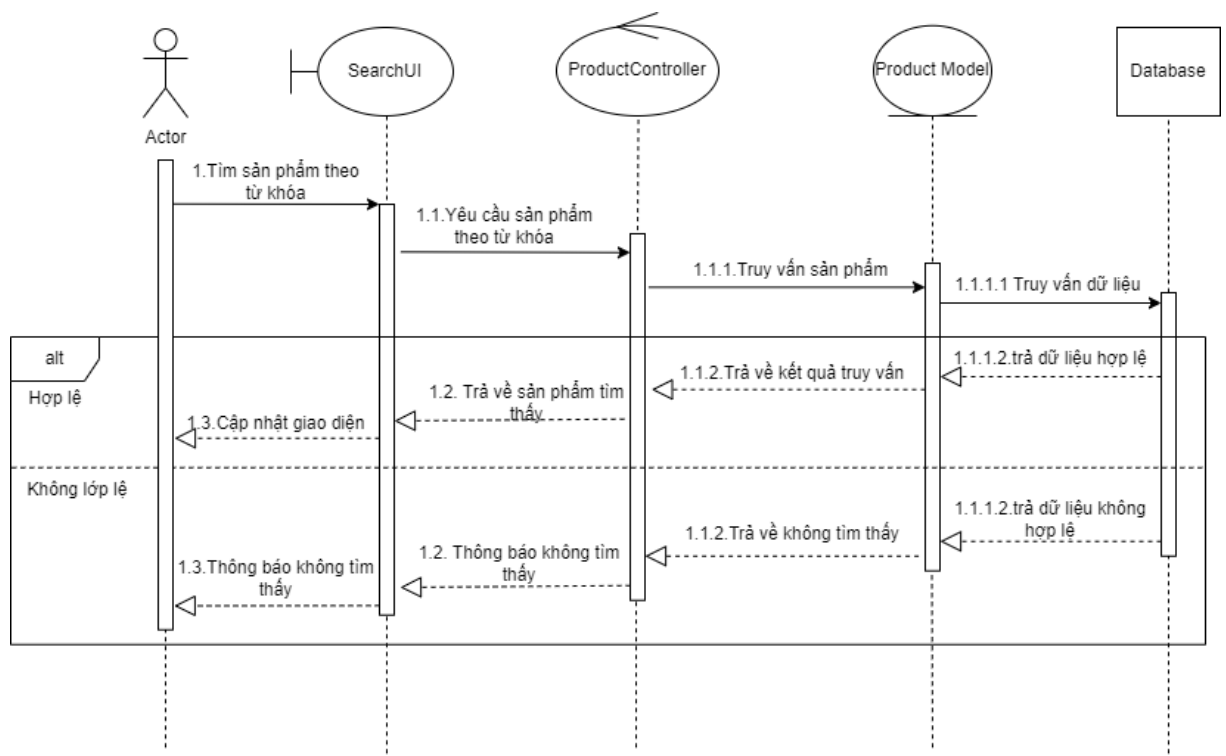
2 Biểu đồ tuần tự các ca sử dụng chính

2.1 Biểu đồ tuần tự “Đăng nhập”



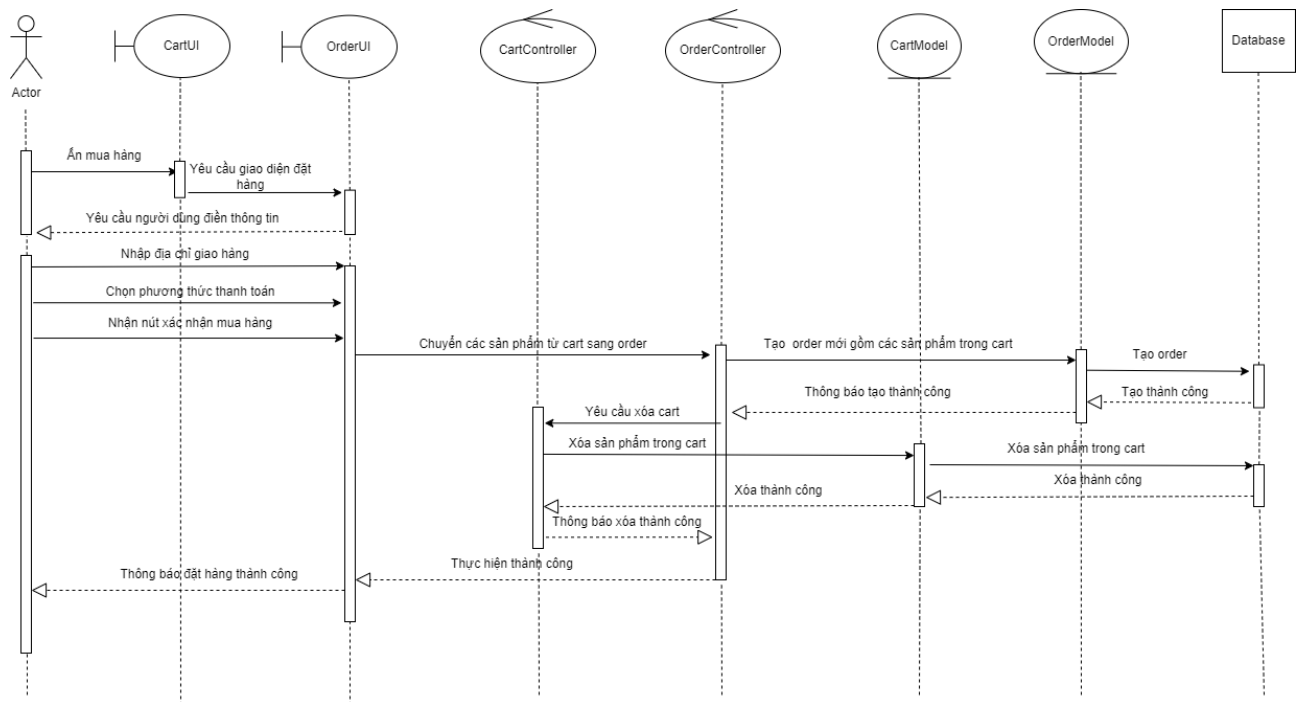
Hình 12: Biểu đồ tuần tự “Đăng nhập”

2.2 Biểu đồ tuần tự “Tìm kiếm”



Hình 13: Biểu đồ tuần tự “Tìm kiếm”

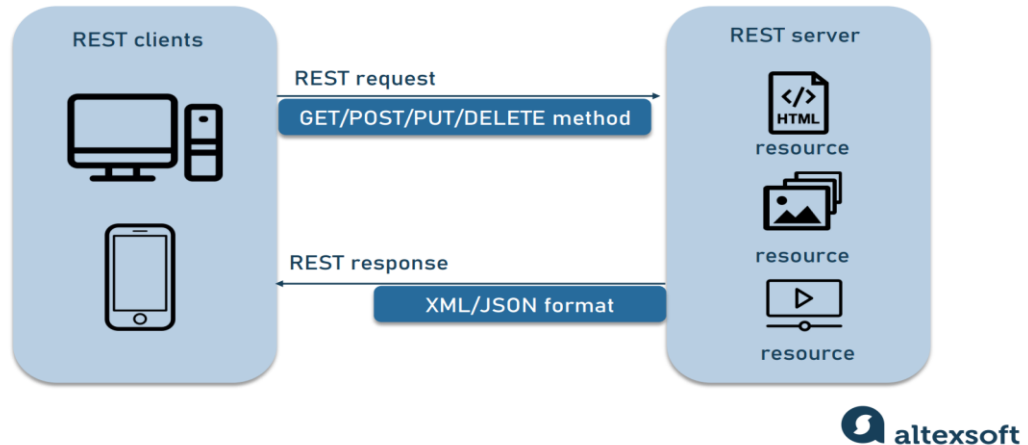
2.3 Biểu đồ tuần tự “Mua hàng”



Hình 14: Biểu đồ tuần tự “Mua hàng”

3 Thiết kế REST API

REST API IN ACTION



Hình 15: Cách hoạt động REST API

Trong sản phẩm của chúng em, để thiết kế API dễ hiểu và phân chia rõ ràng chúng em tuân thủ theo các tiêu chí sau:

- Xác định mục đích của API, từ đó xác định phạm vi và chức năng của API.
- Xác định endpoints cho từng các tác vụ, các tác vụ có liên quan đến nhau sẽ có cùng một thành phần tên trong yêu cầu để dễ dàng đọc hiểu và tăng tính nhất quán.
- Xác định các phương thức phù hợp với nhiệm vụ của từng tác vụ: GET (truy vấn và lấy dữ liệu từ nguồn tài nguyên), POST (gửi dữ liệu mới lên máy chủ để tạo mới một tài

nguyên), PUT (cập nhật hoặc tạo mới một tài nguyên với dữ liệu đã xác định), DELETE (xóa một tài nguyên cụ thể)

- Quản lý trạng thái và lỗi: Xác định cách API xử lý và trả về trạng thái và lỗi. Định nghĩa các mã trạng thái HTTP (200, 400, 404, 500) để biểu thị thành công, lỗi người dùng, lỗi không tìm thấy, và lỗi hệ thống.

Dự án của chúng em về xây dựng website bán đồ ăn, đầu tiên ta cần xây dựng các chức năng để thực hiện các tác vụ như đăng nhập, đăng ký, đăng xuất, lấy thông tin, ... chúng em xây dựng như sau:



```
JavaScript
router.get('/info', isAuth, userController.getInfo);
router.get('/login', isAlreadyLogin, userController.getLoginPage);
router.post('/login', isAlreadyLogin, userController.login);
router.post('/logout', isAuth, userController.logout);
router.post('/register', isAlreadyLogin, userController.register);
router.put('/edit-info', isAuth, userController.updateUserInfo);
router.put('/change-password', isAuth, userController.updatePassword);
router.get('/admin/all-user', isAuth, isAdmin, userController.getAllUser);
router.delete('/admin/delete-user', isAuth, isAdmin, userController.deleteUser);
```

Hình 16: Xây dựng REST API xử lý các tác vụ liên quan đến user

- Chúng em đặt tên các đường dẫn URL cùng với các tác vụ mà nó đảm nhận, với các tác vụ của admin thì sẽ thêm cụm từ **/admin/** vào đường dẫn để phân biệt với các chức năng của user.
- Lựa chọn phương thức GET với các tác vụ yêu cầu lấy dữ liệu như GET /info, GET /login, ... Phương thức POST để gửi dữ liệu lên server như username, password thực hiện đăng nhập, đăng ký, ... Các phương thức PUT để chỉnh sửa dữ liệu, thay đổi thông tin người dùng, và phương thức DELETE để xóa user.
- Sử dụng các phương thức như **isAuth**, **isAdmin** để xác thực người dùng và admin.
- Các hàm xử lý yêu cầu từ phía người dùng, kiểm soát và xử lý lỗi, trả về các mã trạng thái và kết quả yêu cầu sẽ được xử lý trong các file Controller.

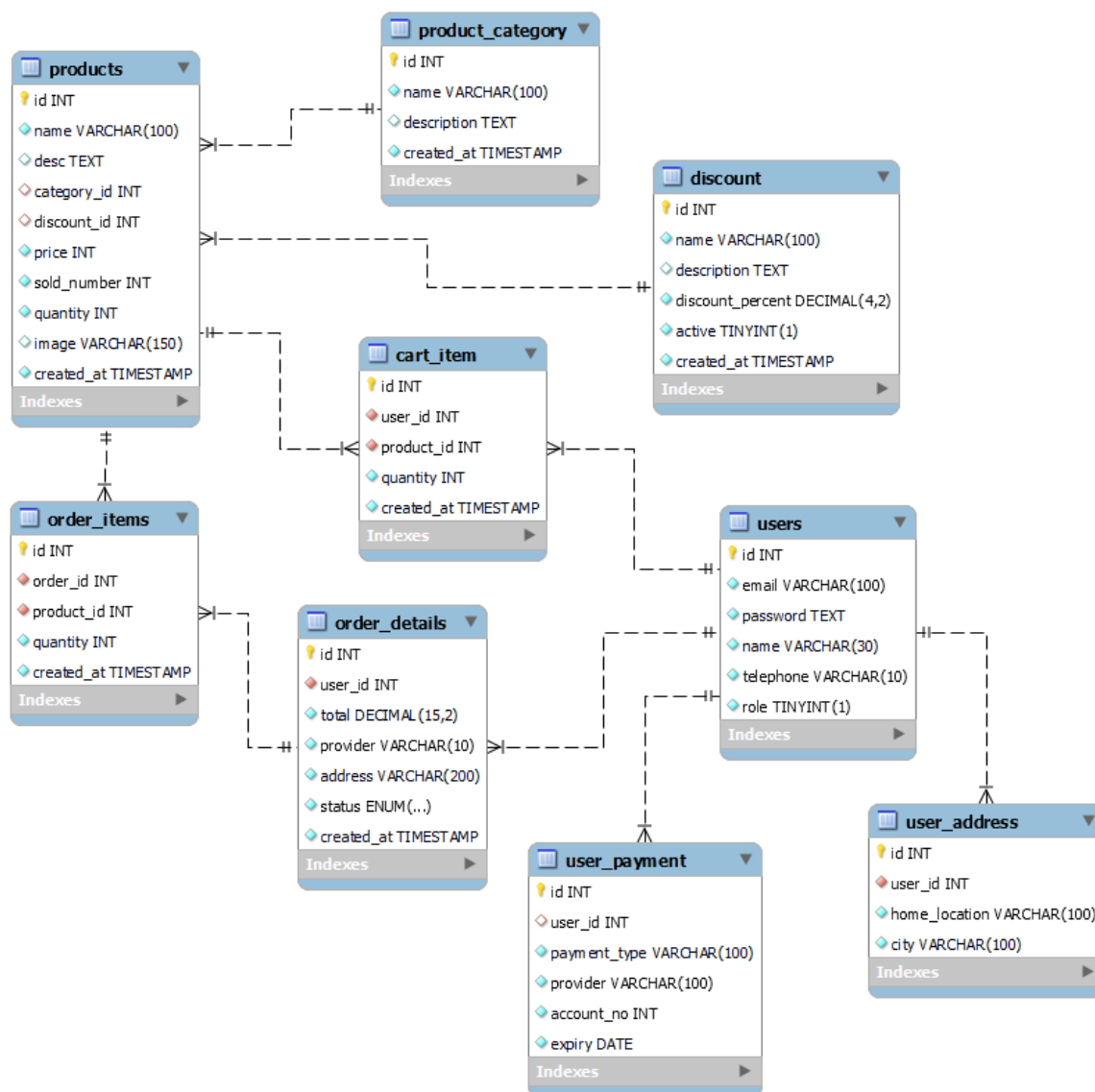
Với từng tác vụ cụ thể, chúng em đều tuân theo các tiêu chí thiết kế REST API đã đề cập ở trên:

```
JavaScript ▾ Copy Caption ...  
  
router.get(  
  '/admin/get-all-orders',  
  isAuth,  
  isAdmin,  
  orderController.getAllOrders,  
);  
router.put(  
  '/admin/update-status',  
  isAuth,  
  isAdmin,  
  orderController.postUpdateStatus,  
);  
  
router.get('/order-by-user', isAuth, orderController.getOrderByUser);  
router.post('/create-order', isAuth, orderController.postCreateOrder);  
router.put('/update-order', isAuth, orderController.postUpdateOrder);  
router.delete('/delete-order', isAuth, orderController.postDeleteOrder);
```

Hình 17: Thiết kế REST API các tác vụ về order

4 Thiết kế cơ sở dữ liệu

4.1 Cơ sở dữ liệu mức căn bản



Hình 18: Thiết kế cơ sở dữ liệu

4.2 Danh sách các thành phần cơ sở dữ liệu

Bảng 1 Các thành phần cơ sở dữ liệu

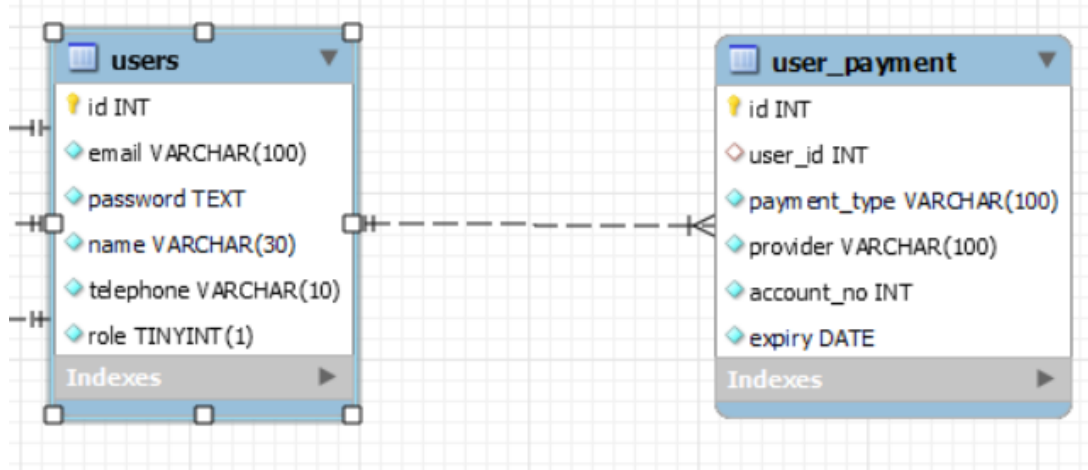
Tên bảng	Nhiệm vụ
users	Lưu thông tin của cơ bản của người dùng
user_payment	Lưu thông tin về các phương thức thanh toán của người dùng
user_address	Lưu địa chỉ của người dùng
discount	Lưu thông tin về các chương trình giảm giá

products	Lưu thông tin chi tiết về các sản phẩm
product_category	Lưu thông tin về danh mục sản phẩm
order_details	Lưu thông tin chi tiết về đơn hàng.
order_items	Lưu thông tin các sản phẩm của đơn hàng. Có tham chiếu tới bảng order_details
cart_item	Lưu thông tin về các sản phẩm có trong giỏ hàng.

4.3 Quan hệ giữa các thành phần

Quan hệ giữa users và user_payment

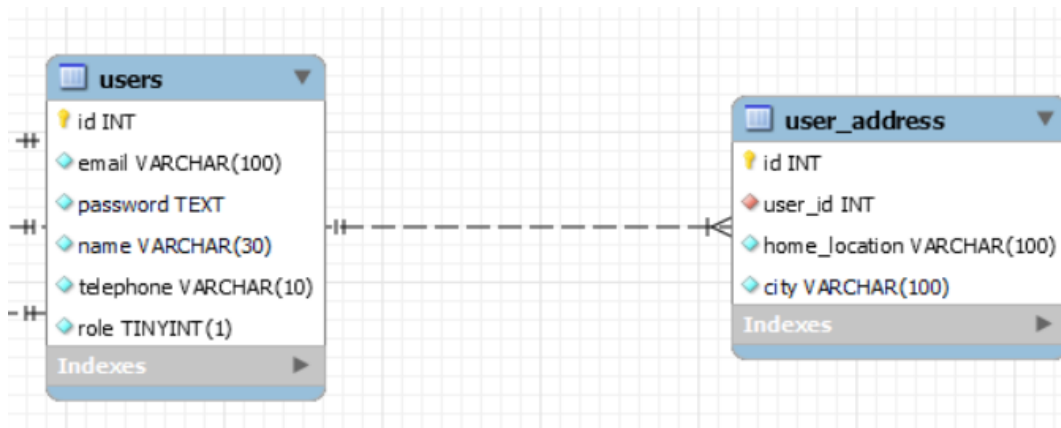
Mô tả: Mỗi user_payment sẽ thuộc về 1 user và mỗi users có thể có nhiều user_payment. Mỗi quan hệ giữa users và user_payment là quan hệ 1-n (một-nhiều).



Hình 19: Quan hệ giữa users và user_payment

Quan hệ giữa users và user_address

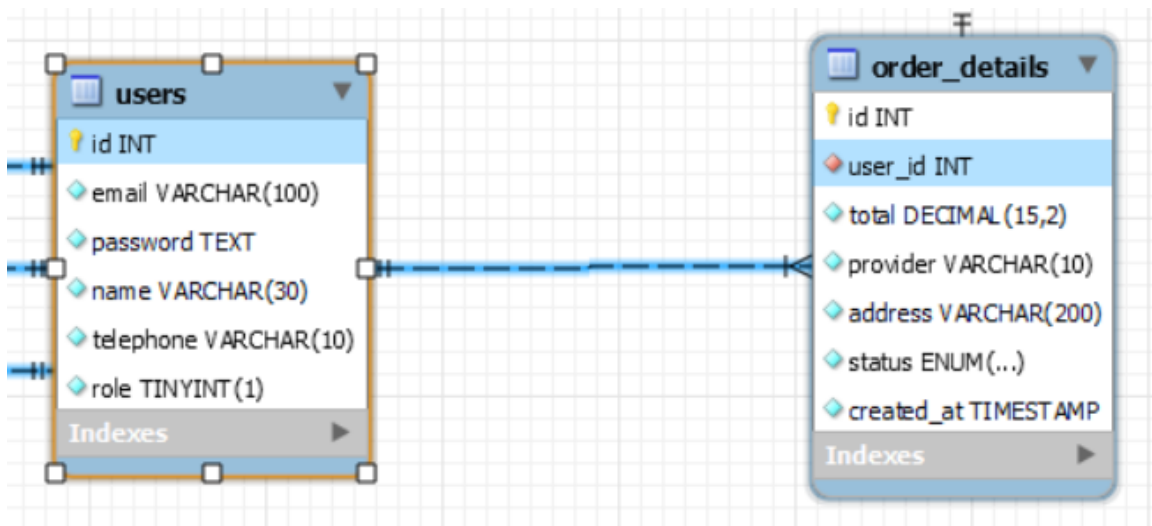
Mô tả: Mỗi users có thể có nhiều user_address, mỗi user_address thuộc về duy nhất 1 user. Mỗi quan hệ giữa users và users_address là quan hệ 1-n (một – nhiều).



Hình 20: Quan hệ giữa users và use_address

Quan hệ giữa users và order_details

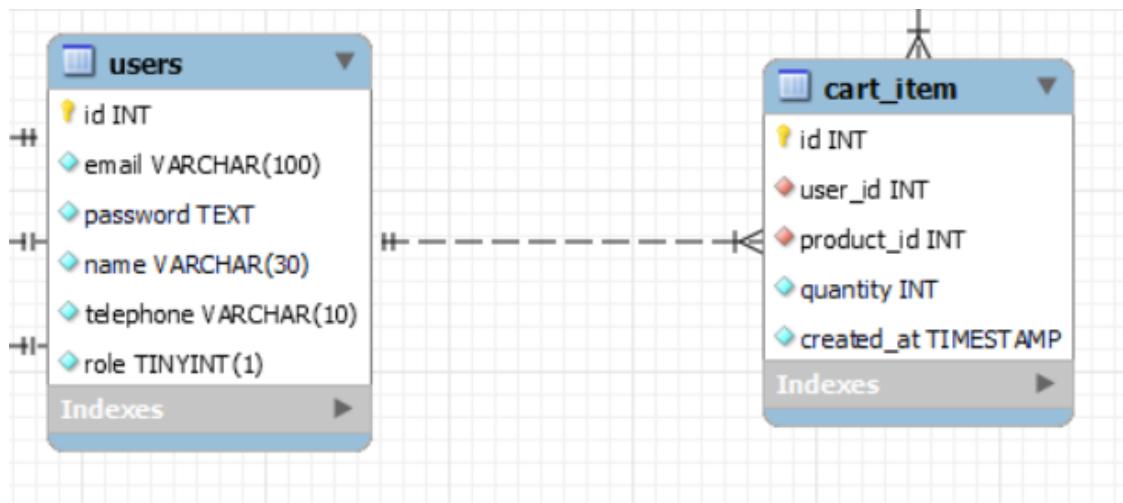
Mô tả: Mỗi users có thể có nhiều order_details, 1 order_details chỉ thuộc về 1 users. Mỗi quan hệ giữa users và order_details là quan hệ 1-n (một - nhiều)



Hình 21: Quan hệ giữa uses và order_details

Quan hệ giữa cart_item và users

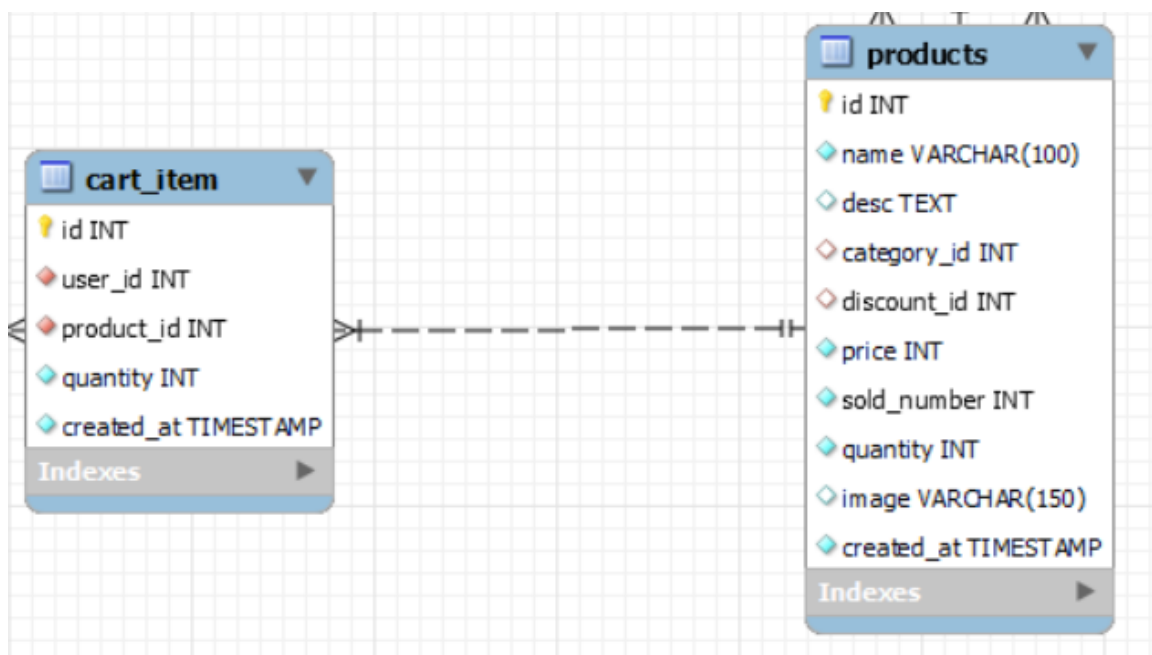
Mô tả: mỗi users sẽ có nhiều cart_item, 1 cart_item sẽ chỉ thuộc về 1 users. Mỗi quan hệ giữa users và cart_item là quan hệ 1-n (một - nhiều).



Hình 22: Quan hệ giữa users và cart_item

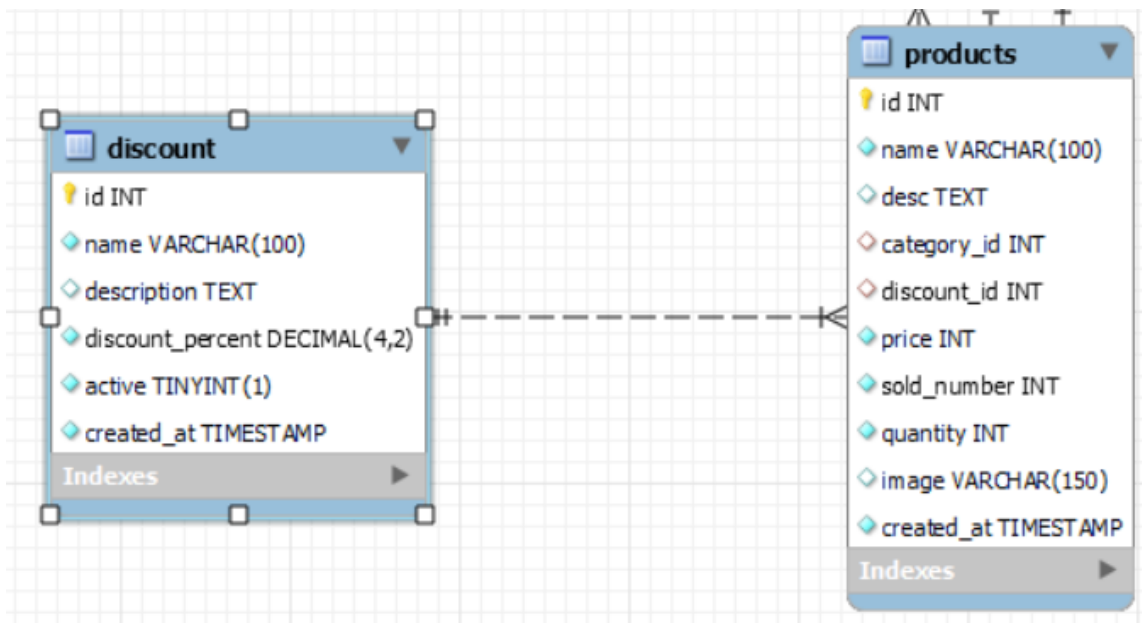
Quan hệ giữa cart_item và products

Mô tả: mỗi cart_item có 1 products, mỗi products có thể thuộc nhiều cart_item. Mỗi quan hệ giữa products và cart_item là quan hệ 1-n (một - nhiều).



Hình 23: Quan hệ giữa cart_item và products

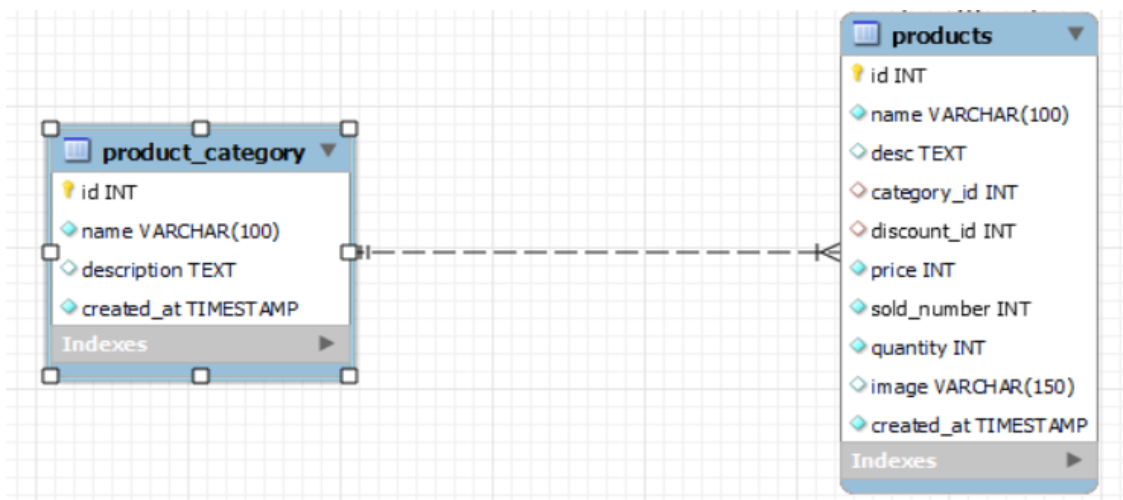
Quan hệ giữa products và discount



Hình 24: Quan hệ giữa discount và products

Quan hệ giữa products và product_category

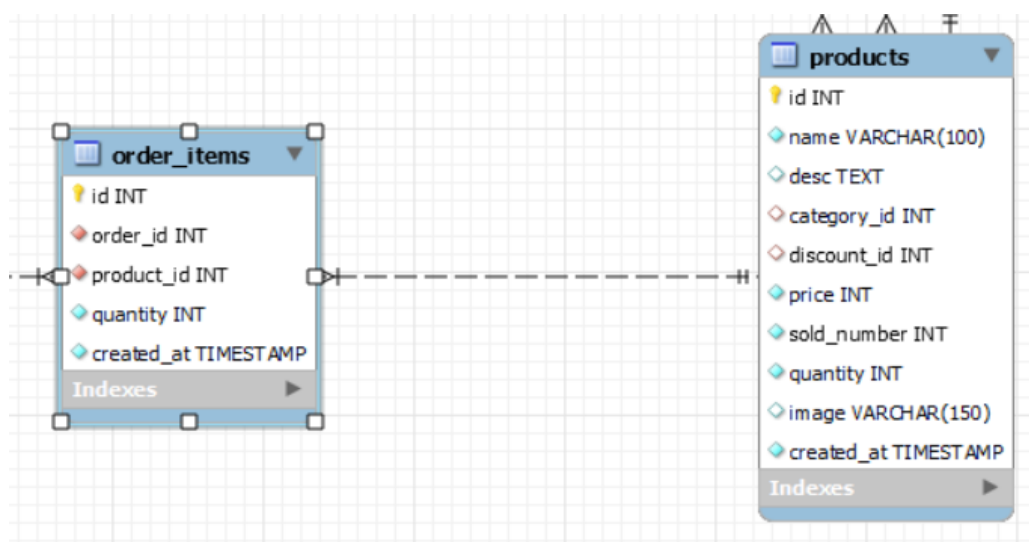
Mô tả: Mỗi products sẽ thuộc về 1 product_category, mỗi product_category có thể là nhiều products. Mỗi quan hệ giữa products và product_category là 1-n (một - nhiều).



Hình 25: Quan hệ giữa products và product_category

Quan hệ giữa products và order_items

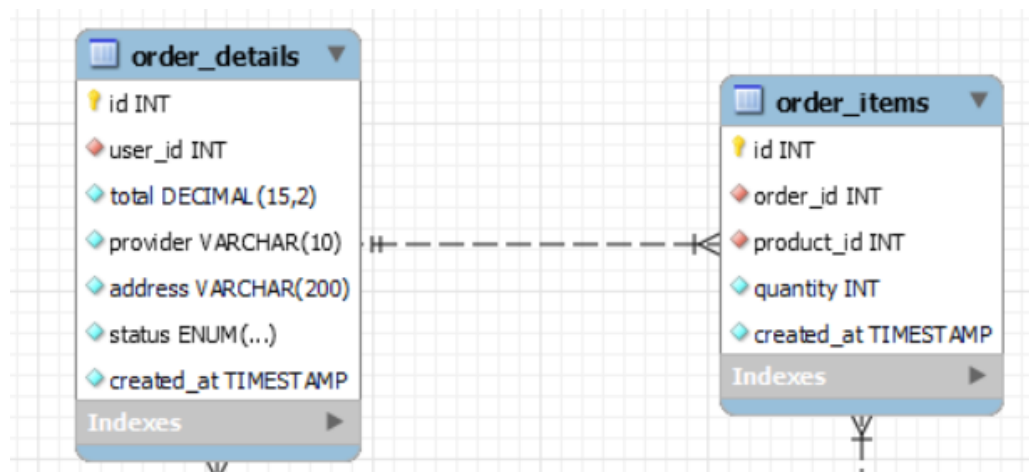
Mô tả: Mỗi products sẽ thuộc về 1 order_items, mỗi order_items sẽ có 1 products. Mỗi quan hệ giữa products và order_items là 1-1 (một - một).



Hình 26: Quan hệ giữa order_items và products

Quan hệ giữa order_items và order_details

Mô tả: Mỗi order_items sẽ thuộc về 1 order_details, mỗi order_details có thể có nhiều order_items. Mỗi quan hệ giữa order_details và order_detail là 1-n (một – nhiều).



5 Một số giao diện chính

5.1 Giao diện người dùng

Giao diện đăng ký

Feastly Home Menu Contact Us

Tim kiếm ...

TẠO TÀI KHOẢN

Nhập Tên Của Bạn

Nhập Mail Của Bạn
vinh@gmail.com

Nhập Mật Khẩu Của Bạn

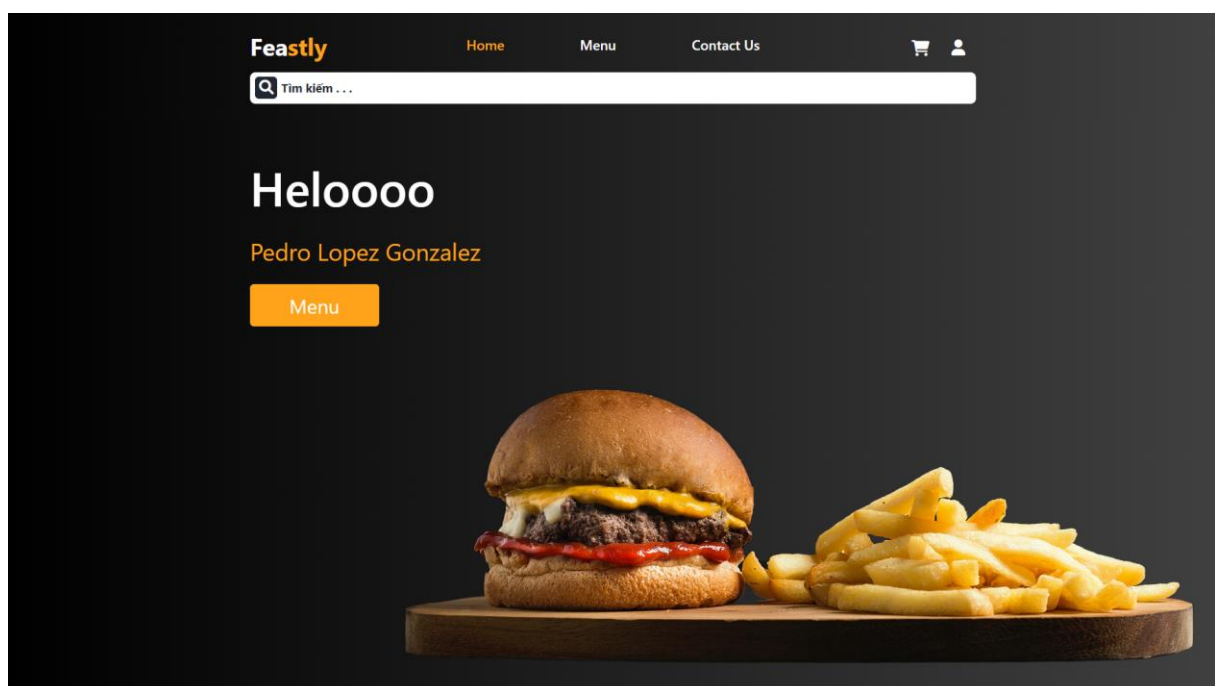
Nhập Lại Mật Khẩu Của Bạn

Nhập Số Điện Thoại Của Bạn

Đăng Ký
Đã Có Tài Khoản? Đăng Nhập

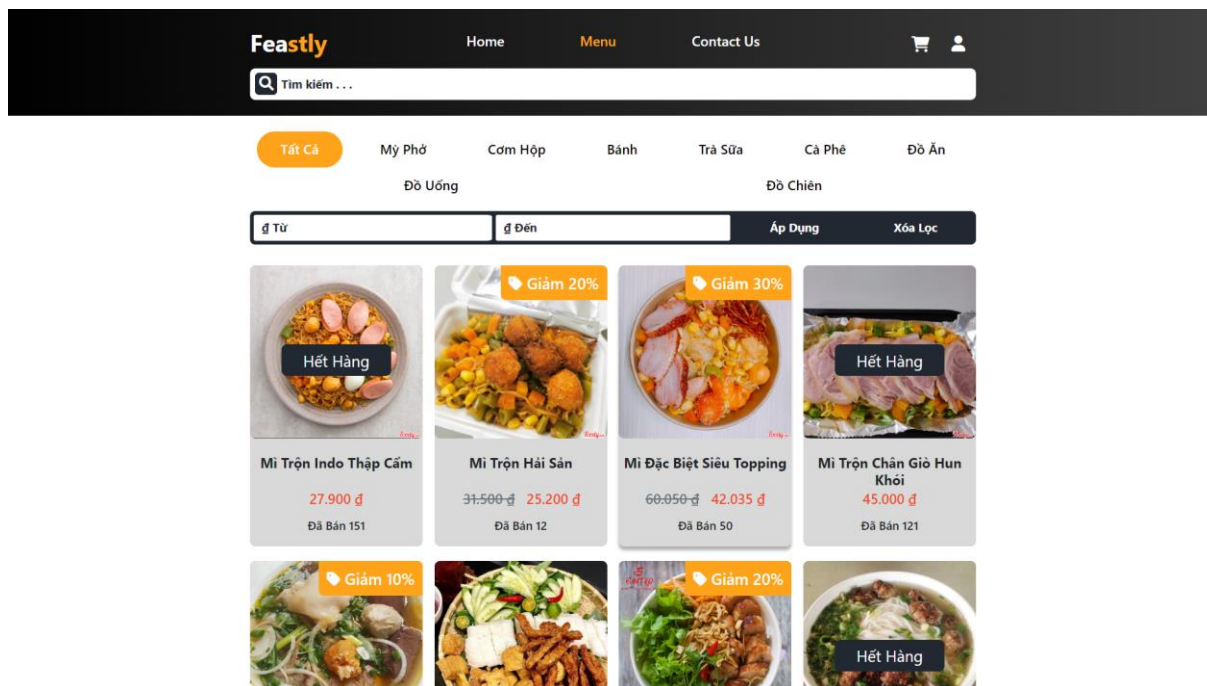
Hình 27: Giao diện đăng ký

Giao diện trang chủ



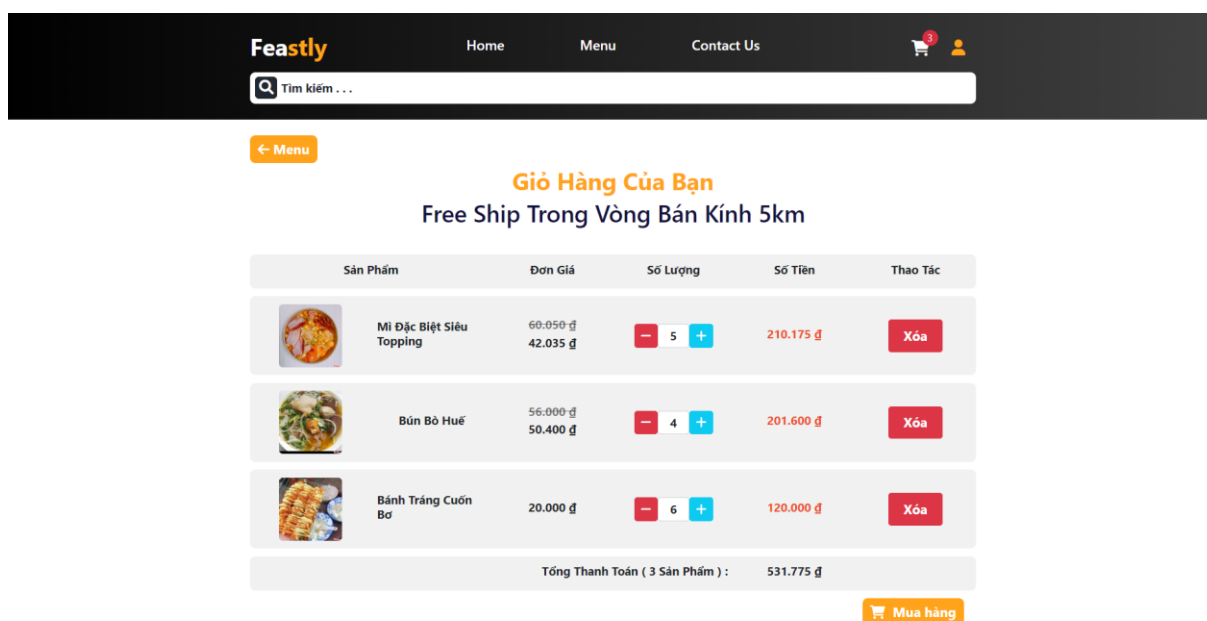
Hình 28: Giao diện trang chủ

Giao diện danh sách sản phẩm



Hình 29: Giao diện danh sách sản phẩm

Giao diện giỏ hàng



Hình 30: Giao diện giỏ hàng




Giao diện đặt hàng

Địa Chỉ Nhận Hàng

Vinh
(+84) 123 456 789

114 Xuân Thủy , Phường Dịch Vọng , Quận Cầu Giấy ,
Thành Phố Hà Nội

Thay Đổi

Sản Phẩm	Đơn Giá	Số Lượng	Thành Tiền
 Mì Đặc Biệt Siêu Topping	42.035 đ	5	210.175 đ
 Bún Bò Huế	50.400 đ	4	201.600 đ
 Bánh Tráng Cuốn Bơ	20.000 đ	6	120.000 đ
Tổng Số Tiền (3 Sản Phẩm):			531.775 đ

Phương Thức Thanh Toán:

Thanh Toán Bằng QR

Thanh Toán Khi Nhận Hàng

Đặt hàng

Hình 31: Giao diện đặt hàng

5.2 Giao diện quản trị



Giao diện quản lý sản phẩm


Feastly

Home





Menu

Contact Us

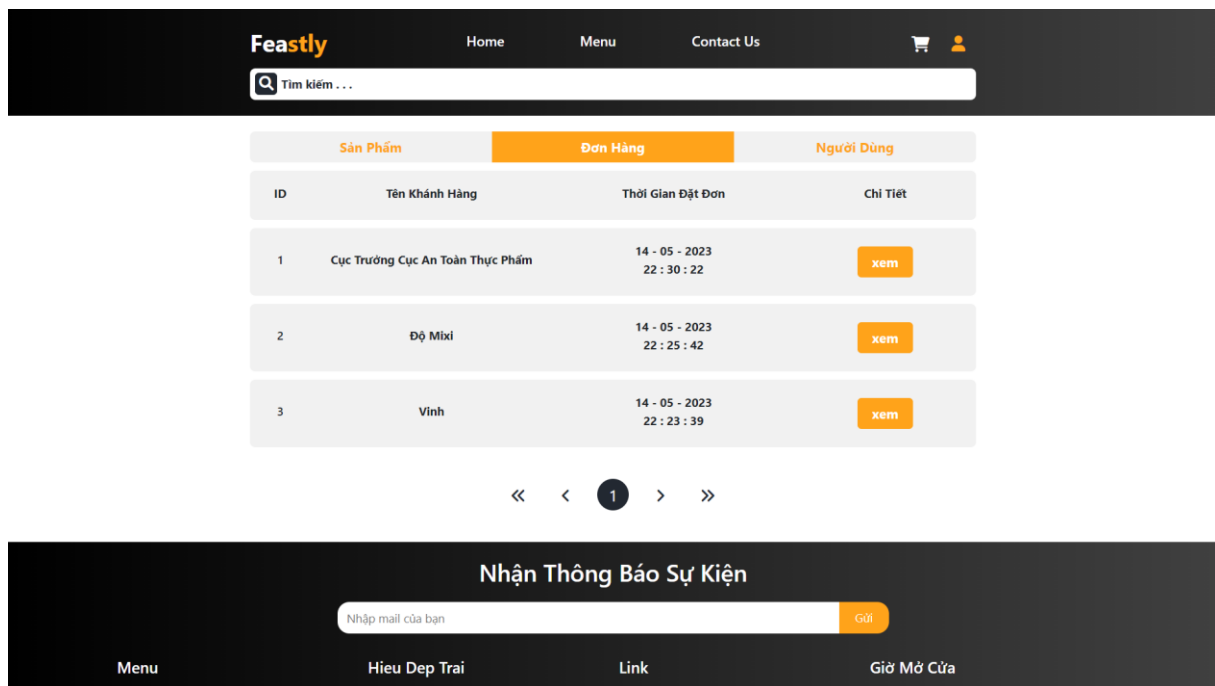


Tìm kiếm ...

Sản Phẩm	Đơn Hàng	Người Dùng	
<div>Thêm sản phẩm mới</div>			
Thông Tin Sản Phẩm	Đơn Giá	Đã Bán	Chi Tiết
 <div>Mì Trộn Indo Thập Cẩm</div>	27.900 đ	151	<div>sửa</div> <div>xóa</div>
 <div>Mì Trộn Hải Sản</div>	<div>31.500 đ</div> <div>25.200 đ</div>	12	<div>sửa</div> <div>xóa</div>
 <div>Mì Đặc Biệt Siêu Topping</div>	<div>60.050 đ</div> <div>42.035 đ</div>	50	<div>sửa</div> <div>xóa</div>
 <div>Mì Trộn Chân Giò Hun Khô</div>	45.000 đ	121	<div>sửa</div> <div>xóa</div>

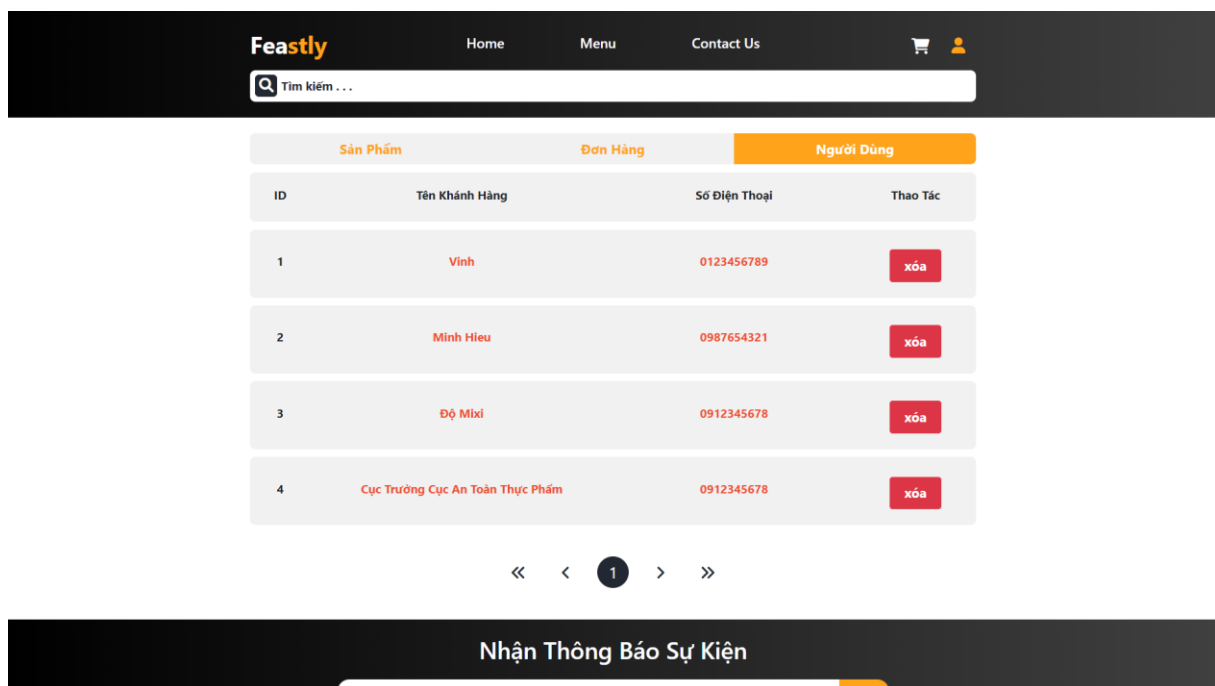
Hình 32: Giao diện quản lý sản phẩm

Giao diện quản lý đơn hàng



Hình 33: Giao diện quản lý đơn hàng

Giao diện quản lý người dùng



Hình 34: Giao diện quản lý người dùng

Chương 5: Cài đặt và kiểm thử

1 Cài đặt hệ thống

Yêu cầu:

- Cần có sẵn IDE (các phần mềm hỗ trợ lập trình)
- Tải xuống NodeJS, npm.
- Tải xuống hệ quản trị cơ sở dữ liệu MySQL (sử dụng xampp) và Redis server.

Cài đặt:

- Tải xuống bản sao folder dự án trên <https://github.com/hieuunm/SE.git> thông qua câu lệnh: **git clone https://github.com/hieuunm/SE.git**.
- Nhập theo thứ tự dữ liệu vào MySQL qua 2 file là [feastly.sql](#) và [feastly-data.sql](#).
- Mở folder dự án trong IDE (Visual Studio Code), bật terminal và truy cập lần lượt vào folder client và server chạy câu lệnh: **npm install**.
- Sau khi tải xuống các modules và không xảy ra lỗi, chương trình đã có đầy đủ các điều kiện để khởi chạy.

Chạy chương trình:

- Đầu tiên, ta cần chạy 2 hệ quản trị cơ sở dữ liệu là mysql server và redis server.
- Bật lên 2 terminals trong IDE, 1 terminal truy cập vào folder client chạy câu lệnh: **npm run serve**. Terminal còn lại truy cập vào folder server chạy câu lệnh: **npm start**.
- Truy cập trên browser địa chỉ: **http://localhost:8080/**.

2 Kiểm thử hệ thống

Sau khi đã khắc phục các lỗi và thực hiện kiểm thử lại hệ thống, dưới đây là danh sách mô tả chi tiết kết quả các ca kiểm thử đã được thực hiện tính đến ngày 8/5/2023.

2.1 User

STT	Yêu cầu chức năng	Tên ca kiểm thử	Điều kiện	Dữ liệu	Kết quả mong đợi	Kết quả thực tế	Trạng thái
1	Đăng ký tài khoản	Hiển thị màn hình đăng ký tài khoản		Không	Hiển thị giao diện cho người dùng đăng ký tài khoản	Như mong đợi	Thành công
		Xác thực tính chính xác của thông tin		{ "name": "fullname", "email": "admin", "password": "admin", "phonenumber": "123"	Báo lỗi các thông tin sai định dạng, không	Như mong đợi	Thành công

				}	đúng độ dài yêu cầu		
		Xác thực các thông tin khi đăng ký tài khoản	Đã tồn tại tài khoản sử dụng email đó	{ “name”: “fullname”, “email”: “admin@gmail.com”, “password”: “admin”, “phonenumber”: “0123456789” }	Thông báo đã có email tồn tại, yêu cầu nhập email khác	Như mong đợi	Thành công
		Gửi yêu cầu đến trang đăng ký	Đã đăng nhập thành công	Không	Điều hướng đến trang chủ của hệ thống	Như mong đợi	Thành công
		Xác thực tài khoản khi đăng ký thành công		{ “name”: “fullname”, “email”: “admin@gmail.com”, “password”: “admin”, “phonenumber”: “0123456789” }	Điều hướng đến trang chủ của hệ thống	Như mong đợi	Thành công
2	Đăng nhập vào hệ thống	Hiện thị màn hình đăng nhập hệ thống		Không	Hiện thị giao diện đăng nhập cho người dùng	Như mong đợi	Thành công
		User gửi yêu cầu đến trang đăng nhập.	Đã hoàn thành việc đăng nhập	Không	Điều hướng đến trang chủ của hệ thống	Như mong đợi	Thành công
		Nhập không đủ dữ liệu		{ “email”: “vinh@gmail.com”, }	Hiện thị thông báo lỗi, yêu cầu	Như mong đợi	Thành công

		vào đăng nhập			điền đủ thông tin.		
		Xác thực tài khoản đăng nhập		{ “email” : “admin@gmail.com” “password” : “admin” }	Điều hướng đến trang chủ của hệ thống nếu thông tin hợp lệ	Như mong đợi	Thành công
3	Đăng xuất khỏi tài khoản đang đăng nhập	Kiểm tra đăng xuất tài khoản	Đã đăng nhập thành công	Không	Rời khỏi màn hình tài khoản hiện tại, điều hướng về trang home.	Như mong đợi	Thành công
		Kiểm tra đăng xuất tài khoản khi chưa đăng nhập		Không	Thông báo bạn cần đăng nhập trước mới có thể đăng xuất	Như mong đợi	Thành công
4	Thay đổi thông tin	Hiện thị màn hình thay đổi thông tin tài khoản		Không	Thông báo chưa đăng nhập, điều hướng đến trang đăng nhập	Không thông báo lỗi, vẫn điều hướng đến trang thông tin	Thất bại
		Thay đổi thông tin người dùng	Đã đăng nhập thành công	{ “name”: “12” “phonenumber”: “12345”	Thông báo name chỉ chứa ký tự	Như mong đợi	Thành công

		không đúng theo cấu trúc		}	và số bắt đầu bằng 0		
		Kiểm tra cập nhật dữ liệu mới thay đổi	Đã đăng nhập thành công	{ “name”: “quan” “phonenumner”: “0123459890” }	Điều hướng về trang thông tin người dùng, hiển thị các thông tin mới cập nhật	Như mong đợi	Thành công
		Thay đổi mật khẩu nhập vào mật khẩu hiện tại không đúng	Đã đăng nhập thành công	{ “currentpassword”: “password” “newpassword”: “newpassword” “recapnewpassword”: “newpassword” }	Thông báo mật khẩu hiện tại không đúng	Như mong đợi	Thành công
5	Tìm kiếm theo từ khóa	Hiển thị hình ảnh những sản phẩm theo từ khóa đã nhập		{ “search”: “my tron” }	Hiện thị ra các sản phẩm có chứa cụm từ “my tron”	Như mong đợi	Thành công
6	Tìm kiếm theo khoảng giá	Nhập vào khoảng giá không hợp lệ		{ “minprice”:"500000", “maxprice”:"300000” }	Thông báo lỗi khoảng giá, không hiển thị ra sản phẩm nào	Như mong đợi	Thành công
		Hiển thị sản phẩm theo khoảng		{ “minprice”:"10000”, “maxprice”:"300000” }	Đưa ra sản phẩm có giá trị trong	Như mong đợi	Thành công

		giá đã nhập			khoảng giá đã nhập vào thanh ghi		
7	Thêm vào giỏ hàng	Thêm sản phẩm vào giỏ hàng khi chưa đăng nhập		Không	Thông báo cần đăng nhập trước khi sử dụng chức năng	Như mong đợi	Thành công
		Thêm sản phẩm vào giỏ hàng của người dùng	Đã đăng nhập thành công	{“product_id”:17, “quantity”:2}	Thêm dữ liệu sản phẩm vào giỏ hàng của bản thân	Như mong đợi	Thành công
8	Xóa sản phẩm trong giỏ hàng	Xóa sản phẩm trong giỏ hàng	Đã đăng nhập thành công	{“product_id”:17}	Cập nhật lại số sản phẩm có trong giỏ hàng	Như mong đợi	Thành công
		Xóa sản phẩm không có trong giỏ hàng của người dùng	Đã đăng nhập thành công	{“product_id”:18}	Thông báo sản phẩm không có trong giỏ hàng, không thể xóa	Như mong đợi	Thành công
9	Chỉnh sửa số lượng sản phẩm trong giỏ hàng	Chỉnh sửa số lượng sản phẩm xuống 0	Đã đăng nhập thành công	{ “product_id”:17, “decrease_btn”: true }	Thông báo bạn có muốn xóa sản phẩm khỏi giỏ hàng	Như mong đợi	Thành công

10	Xác nhận thanh toán giỏ hàng	Xác nhận thanh toán khi trong giỏ hàng không có sản phẩm	Đã đăng nhập thành công	Không có gì trong giỏ hàng	Thông báo cần phải có sản phẩm mới thanh toán hoặc không hiển thị ra nút thanh toán	Như mong đợi	Thành công
11	Thanh toán hóa đơn, tạo order	Lấy thông tin địa chỉ của user đã lưu từ trước	Đã đăng nhập thành công	Không	Trả về địa chỉ lưu gần nhất của khách hàng, nếu không có thì không trả về gì	Như mong đợi	Thành công
		Tạo order, không có thông tin địa chỉ của khách hàng	Đã đăng nhập thành công	Dữ liệu là các sản phẩm trong cart, giá cả, thông tin khách hàng không bao gồm địa chỉ	Yêu cầu phải nhập địa chỉ mới có thể hoàn thiện thanh toán	Như mong đợi	Thành công
12	Quản lý các đơn mua	Hiển thị lịch sử các đơn hàng của khách hàng.	Đã đăng nhập thành công	Không	Hiển thị thông tin, trạng thái các đơn hàng theo thứ tự thời gian mới nhất đến lâu nhất.	Như mong đợi	Thành công
		Hủy đơn hàng khi trạng thái là “Đang giao	Đã đăng nhập thành công	{“order_id”: 1}	Không cho phép hủy đơn hàng khi không ở trạng thái	Như mong đợi	Thành công

		hàng” hoặc “Đã hoàn thành”			“Đang xử lý”		
--	--	-------------------------------------	--	--	-----------------	--	--

2.2 Admin

STT	Yêu cầu chức năng	Tên ca kiểm thử	Điều kiện	Dữ liệu	Kết quả mong đợi	Kết quả thực tế	Trạng thái
1	Truy cập vào trang chủ của admin	User truy cập vào trang chủ admin	Đã đăng nhập vào hệ thống với tư cách user	Không	Báo lỗi cho user và điều hướng về trang chủ của hệ thống	Như mong đợi	Thành công
2	Thêm sản phẩm	Thêm sản phẩm với dữ liệu theo bảng product	Đã đăng nhập vào hệ thống với tư cách admin	{ “name”: “mỳ Ý”, “desc”: “Mỳ” “price”: ”10000” “quantity”: ”13” “discount”: “15%” “category”: “Mỳ Phở” “image”: “link” }	Hiển thị thông báo đã thêm thành công và hiển thị nó trên menu của người dùng	Như mong đợi	Thành công
		Nhập giá sản phẩm không hợp lệ trong việc thêm sản phẩm	Đã đăng nhập vào hệ thống với tư cách admin	{ “name”: “mỳ Ý”, “desc”: “Mỳ ” “price”: ”0” “quantity”: ”10” “discount”: “15%” “category”: “Mỳ Phở” “image”: “link” }	Báo lỗi và thông báo giá không hợp lệ	Không hiển thị thông báo thêm thành công	Thành công

		Nhập quantity không hợp lệ trong việc thêm sản phẩm	Đã đăng nhập vào hệ thống với tư cách admin	{ “name”: “mỳ Ý”, “desc”: “Mỳ” “price”: ”10000” “quantity”: ”0” “discount”: “15%” “category”: “Mỳ Phở” “image”: “link” }	Báo lỗi và thông báo quantity không hợp lệ	Lưu sản phẩm vào danh sách sản phẩm	Thất bại
		Nhập vào tên sản phẩm có ký tự đặc biệt	Đã đăng nhập vào hệ thống với tư cách admin	{ “name”: “@admin”, “desc”: “Mỳ” “price”: ”10000” “quantity”: ”10” “discount”: “15%” “category”: “Mỳ Phở” “image”: “link” }	Báo lỗi và thông báo không được phép có ký tự đặc biệt ở tên sản phẩm	Vẫn lưu sản phẩm với tên đã nhập	Thất bại
		Không nhập vào các thông tin không quá quan trọng	Đã đăng nhập vào hệ thống với tư cách admin	{ “name”: “mỳ Ý”, “desc”: “” “price”: ”10000” “quantity”: ”13” “discount”: “15%” “category”: “” “image”: “link” }	Báo lỗi và không lưu sản phẩm vào danh sách sản phẩm	Không cho phép lưu sản phẩm	Thành công
		Hiển thị giá sản phẩm với sau khi đã áp dụng discount	Đã đăng nhập vào hệ thống với tư cách admin	{ “name”: “mỳ Ý”, “desc”: “Mỳ” “price”: ”10000” “quantity”: ”13” “discount”: “15%” “category”: “Mỳ Phở” }	Hiển thị sản phẩm trên giao diện user với giá sản phẩm tính sau khi áp dụng discount	Như mong đợi	Thành công

				“image”: “link” }			
3	Xóa sản phẩm	Xóa sản phẩm khỏi danh sách sản phẩm	Đã đăng nhập vào hệ thống với tư cách admin	Không	Không hiển thị sản phẩm bị xóa trên giao diện user	Như mong đợi	Thành công
4	Thay đổi thông tin sản phẩm	Nhập giá sản phẩm bằng 0	Đã đăng nhập vào hệ thống với tư cách admin	{ “name”: “My tron”, “price”: “0”, “discount”: “15%”, “quantity”: “30” }	Không lưu lại những thông tin đã nhập	Vẫn lưu lại giá sản phẩm với giá bằng 0	Thất bại
		Chỉnh sửa thông tin sản phẩm	Đã đăng nhập vào hệ thống với tư cách admin	{ “name”: “My tron”, “price” : “100000”, “discount”: “30%” “quantity”: “30” }	Cập nhật lại thông tin sản phẩm trên giao diện người dùng	Như mong đợi	Thành công
		Cập nhật thông tin sản phẩm trong giỏ hàng của user	Sau khi admin thay đổi thông tin sản phẩm, đăng nhập vào hệ thống với tư cách user	Không	Cập nhật lại thông tin của sản phẩm trong giỏ hàng của người dùng	Như mong đợi	Thành công
5	Danh sách các	Danh sách đơn hàng đã	Đã đăng nhập vào hệ thống	Không	Hiển thị danh sách các đơn	Như mong đợi	Thành công

	đơn hàng	được xác nhận	với tư cách admin		hàng đã được yêu cầu		
		Xem chi tiết đơn hàng của khách hàng	Đã đăng nhập vào hệ thống với tư cách admin	Không	Hiển thị thông tin chi tiết về yêu cầu của khách hàng về sản phẩm	Như mong đợi	Thành công
		Cập nhật trạng thái đơn hàng	Đã đăng nhập vào hệ thống với tư cách admin	{ “order_id” : “1”, “statusOrder”: “Đang giao hàng” }	Cập nhật trạng thái đơn hàng có id là 1 thành “Đang giao hàng”	Như mong đợi	Thành công
6	Quản lý khách hàng	Danh sách các user có trong hệ thống	Đã đăng nhập vào hệ thống với tư cách admin	Không	Hiển thị danh sách các user đã tạo tài khoản trong hệ thống	Như mong đợi	Thành công
		Xóa tài khoản user	Đã đăng nhập với tư cách admin	{ “user_id”: “1” }	User có id là 1 bị xóa không thể đăng nhập vào hệ thống	Như mong đợi	Thành công
		Xóa các đơn hàng đã đặt của user bị xóa tài khoản	Đã đăng nhập với tư cách admin và xóa 1 user	Không	Xóa các đơn hàng của user đã bị xóa tài khoản khỏi danh sách các đơn hàng	Như mong đợi	Thành công

Chương 6: Kết luận và định hướng phát triển

1 Kết luận

Kết thúc quá trình phát triển phần mềm đa số đã hoàn thành được những yêu cầu đã đặt ra trước đó của nhóm như là giúp các nhà hàng, chuỗi cửa hàng dễ dàng quản lý sản phẩm, cung cấp thông tin sản phẩm tới khách hàng.

Nhược điểm

- Dữ liệu chưa đầy đủ. Mỗi sản phẩm mới có 1 ảnh hiển thị.
- Chưa có hình thức đăng nhập bằng tài khoản mạng xã hội
- Giao diện chưa linh hoạt, không thể thay đổi template cho giao diện
- Cơ sở dữ liệu nhỏ, chưa phân quyền tối ưu.

2 Định 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 trong tương lai nhóm chúng em sẽ có những cập nhật sau đây:

- Thiết kế trang web đẹp hơn, tiện dụng hơn, có thêm chức năng và sinh động hơn.
- Xây dựng thêm 1 diễn đàn trên trang web để khách hàng có thể trao đổi bình luận về sản phẩm
- Tạo một phiên bản dành cho khách nước ngoài muốn mua hàng.
- Nâng cao khả năng bảo mật cho chương trình.

TÀI LIỆU THAM KHẢO

- [1] “Tìm hiểu về thương mại điện tử”. [Trực tuyến]. Khả dụng: <https://sell.amazon.vn/training-resource/what-is-ecommerce>. [Đã truy cập: 08/05/2023].
- [2] “Các lợi ích mà thương mại điện tử mang lại”. [Trực tuyến]. Khả dụng: <https://www.techtarget.com/searchcio/definition/e-commerce>. [Đã truy cập: 07/05/2023].
- [3] “MEVN stack là gì? Từng công nghệ đảm nhận xử lý ở tầng nào?”. [Trực tuyến]. Khả dụng: <https://medium.com/bb-tutorials-and-thoughts/how-to-develop-and-build-mevn-stack-10b6e8ff7f7c>. [Đã truy cập: 09/05/2023].
- [4] “Tìm hiểu thêm về MEVN stack”. [Trực tuyến]. Khả dụng: <https://www.geeksforgeeks.org/what-is-mevn-stack/>. [Đã truy cập: 07/05/2023].
- [5] “VueJS là gì? Ứng dụng và tính năng nổi trội.”. [Trực tuyến]. Khả dụng: <https://glints.com/vn/blog/vuejs-la-gi/#.ZGDuunZByUm>. [Đã truy cập: 11/05/2023].
- [6] “Tìm hiểu thêm về framework VueJS”. [Trực tuyến]. Khả dụng: <https://viblo.asia/p/gioi-thieu-framework-vuejs-gDVK2m0X5Lj>. [Đã truy cập: 11/05/2023].
- [7] “NodeJS là gì? Những điều cần biết về NodeJS”. [Trực tuyến]. Khả dụng: <https://niithanoi.edu.vn/nodejs-la-gi-tong-hop-day-du-ve-nodejs-ban-can-biet.html>. [Đã truy cập: 09/05/2023].
- [8] “ExpressJS là gì? Tại sao nên tìm hiểu ExpressJS?”. [Trực tuyến]. Khả dụng: <https://wiki.tino.org/expressjs-la-gi/>. [Đã truy cập: 10/05/2023].
- [9] “Tại sao ExpressJS lại được sử dụng trong NodeJS?”. [Trực tuyến]. Khả dụng: <https://caodang.fpt.edu.vn/tin-tuc-poly/ha-noi-tin-sinh-vien/tai-sao-express-js-duoc-su-dung-trong-node-js.html>. [Đã truy cập: 10/05/2023].
- [10] “Tìm hiểu nguồn gốc, ưu và nhược điểm của MySQL”. [Trực tuyến]. Khả dụng: <https://www.dinhnghia.com.vn/mysql-la-gi-uu-nhuoc-diem-va-tam-quan-trong-cua-mysql/>. [Đã truy cập: 10/05/2023].
- [11] “Tổng quan về MongoDB”. [Trực tuyến]. Khả dụng: <https://viblo.asia/p/tong-quan-ve-mongodb-EoDkQoxqGbV>. [Đã truy cập: 10/05/2023].
- [12] “Tìm hiểu về Redis”. [Trực tuyến]. Khả dụng: <https://viblo.asia/p/tim-hieu-ve-redis-LzD5dXEW5jY>. [Đã truy cập: 11/05/2023].
- [13] “Tham khảo cách làm báo cáo của nhóm 7, đề tài: Ứng dụng tìm trọ cho sinh viên, đại học Công Nghệ”. [Ngoại tuyến]. [Đã truy cập: 14/05/2023].

CHÚ THÍCH

- (1) “Tỉ lệ người mua bán trên các websites thương mại điện tử”. Tìm hiểu thêm:
<https://buildops.com/commercial-construction/number-of-digital-buyers-worldwide/#:~:text=According%20to%20a%20report%20by,56%20billion%20in%202022.>
- (2) “Thời kỳ bùng nổ dot-com”: bóng bóng thị trường cổ phiếu của các công ty công nghệ cao được đầu cơ. Tìm hiểu thêm tại:
https://vi.wikipedia.org/wiki/Bong_b%C3%B3ng_dot-com.
- (3) “Dom”: tên gọi viết tắt Document Object Model. DOM được dùng để truy xuất và thao tác trên các tài liệu có cấu trúc dạng HTML hay XML bằng các ngôn ngữ lập trình thông dụng như Javascript, PHP... thông qua các nút đối tượng. Dom ảo có thể giúp việc truy xuất đến từng nút nhanh hơn.
- (4) ACID(Atomicity, Consistency, Isolation, Durability). Tìm hiểu thêm tại:
<https://www.lifewire.com/the-acid-model-1019731>.
- (5) “Sequelize”: module của NodeJS, hỗ trợ thao tác với các cơ sở dữ liệu. Tìm hiểu thêm tại: <https://sequelize.org/>.
- (6) “ORM”: viết tắt của Object-Relational Mapping. Tìm hiểu thêm tại:
<https://www.theserverside.com/definition/object-relational-mapping>.
- (7) “Session và cookie”: lưu những thông tin xác thực người dùng. Tìm hiểu thêm tại:
<https://topdev.vn/blog/session-la-gi-cookie-la-gi/>.