

Nhận xét của người hướng dẫn

Nhận xét của người phản biện

Tóm tắt

Tên đề tài: HỆ THỐNG TẠO VÀ LIÊN KẾT SÀN THƯƠNG MẠI ĐIỆN TỬ CHO NHÀ SẢN XUẤT VÀ CỬA HÀNG

Sinh viên thực hiện: ...Trần Ngọc Huy

Số thẻ sinh viên: 102180208

Lớp sinh hoạt: 18TCLC_DT3.....

Trong thời kì cả nước thúc đẩy khởi nghiệp. Các mô hình, sản phẩm nông nghiệp, thủ công, truyền thống liên tục ra đời và cải tiến nâng cao chất lượng và quy mô sản xuất.

Quá trình trên giúp cho sản phẩm ngày một đa dạng. Tính truyền thống, đặc trưng vùng miền có cơ hội phát triển.

OCOPEE¹ phát triển giải pháp giúp cho những đơn vị chủ thể tạo kênh thông tin với chi phí thấp. Các cửa hàng phân phối những sản phẩm trên có sàn thương mại riêng để chủ động truyền thông theo cách riêng của họ. Việc xây dựng hệ thống trên, đồng thời cũng phát triển được sự liên kết thông tin giữa các đơn vị với nhau. Tránh sự lãng phí về nhân lực trong quá trình đồng bộ dữ liệu.

OCOPEE định hướng phát triển các tính tối thiểu để một sàn thương mại điện tử có thể hoạt động. Bởi vì khối lượng công việc và nghiệp vụ khá nhiều. Nên mỗi tính năng thêm vào cần được suy xét thêm các yếu tố mang tính doanh nghiệp như tiến độ và ngân sách.

Định hướng tạo nên đột phá cho dự án sẽ được phát triển khi dữ liệu khách hàng và số lượng các nhà sản xuất đủ lớn. Khi đó vấn đề gần nhất cần giải quyết là hệ thống gợi ý giá bán và thu thập thông tin phản hồi từ khách hàng để cải thiện quá trình sản xuất.

¹Viết tắt của tên đề tài

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Sinh viên thực hiện: ...*Trần Ngọc Huy*

Số thẻ sinh viên:*102180208* Lớp sinh hoạt:*18TCLC_DT3*

Khoa:*Công nghệ thông tin* Ngành:*Công nghệ Phần Mềm*

1. Tên đề tài đồ án: *HỆ THỐNG TẠO VÀ LIÊN KẾT SÀN THƯƠNG MẠI ĐIỆN TỬ CHO NHÀ SẢN XUẤT VÀ CỬA HÀNG*

2. Đề tài thuộc diện: [] Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu: Không có.

4. Nội dung các phần thuyết minh và tính toán:

Mở đầu: Phần mở đầu của luận văn, giới thiệu về nhu cầu thực tế và lý do thực hiện đề tài, đồng thời giới thiệu sơ lược về đề tài và mục tiêu phải đạt được.

Chương 1. Cơ sở lý thuyết: trình bày những lý thuyết học được, lý do lựa chọn và cách áp dụng vào hệ thống.

Chương 2. Phân tích và thiết kế: trình bày mục tiêu. Các hồ sơ phân tích và hồ sơ thiết kế trong xây dựng hệ thống.

Chương 3. Triển khai và đánh giá kết quả: mô tả cách cài đặt, vận hành hệ thống và đánh giá kết quả đạt được.

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ): Không có.

6. Họ tên người hướng dẫn: TS. Lê Thị Mỹ Hạnh

7. Ngày giao nhiệm vụ đồ án:/...../ 2022

8. Ngày hoàn thành đồ án:/...../ 2022

Trưởng bộ môn Đà Nẵng, Ngày 1 tháng 7 năm 2022

Người hướng dẫn

Lời nói đầu

Ngành công nghệ phần mềm nghiên cứu về các kỹ thuật, phương pháp, hạ tầng của phần mềm máy tính. Người phát triển phần mềm không đơn giản là chỉ cần biết đến ngôn ngữ lập trình là có thể phát triển được hết. Sau thời gian học tập và nghiên cứu tại trường Đại Học Bách Khoa Đà Nẵng. Cùng với sự thực hành miệt mài và những kinh nghiệm, tư duy, định hướng ngoài giờ lên lớp từ các thầy cô. Em nhận thấy mình tiến bộ toàn diện, không chỉ các kiến thức chuyên ngành. Các phương pháp nghiên cứu, tinh thần học tập, tư duy giải quyết vấn đề,... cũng ngày một hoàn thiện.

Các bạn tân sinh viên có thể thắc mắc rằng vai trò của trường Đại học là gì trong thời kì kiến thức mở. Nếu các bạn thấy được mục tiêu và định hướng của mỗi người là khác nhau khi họ tiếp xúc với những cộng đồng khác nhau. Thì các bạn sẽ nhận ra rằng: bạn bè, thầy cô, giáo trình, cơ sở vật chất tốt không cải thiện năng lực của bạn. Đó là môi trường phát triển. Đơn giản nếu xem năng lực của cá nhân là sản phẩm của tư duy, thì môi trường là "*nguyên liệu*" của sản phẩm đó. Nó cần quá trình chế tác bởi chính năng lực tư duy của bạn. Hiểu được như vậy sẽ không còn nhập nhằng giữa các quan điểm "*Giỏi thì ở đâu cũng giỏi*" hay "*Không thầy đố mày làm nên*".

Đề tài này là kết quả của quá trình học tập tại trường Đại Học Bách Khoa Đà Nẵng. Không chỉ dừng lại ở mức độ là một luận án tốt nghiệp. Những hạn chế tồn đọng sẽ được tiếp tục hoàn thiện khi năng lực và điều kiện phát triển. Đồ án sẽ tiếp tục triển khai và hướng đến công năng thực tế cho các chủ thể doanh nghiệp. Hộ kinh doanh.

Cuối cùng em xin cảm ơn thầy cô và bạn bè đã đồng hành cùng em trong suốt năm tháng vừa qua. Cảm ơn cô Lê Thị Mỹ Hạnh đã hướng dẫn em hoàn thành dự án, luận văn tốt nghiệp.

Em xin chân hành cảm ơn!

Cam đoan

Tôi xin cam đoan:

- Báo cáo đồ án tốt nghiệp đề tài: *HỆ THỐNG TẠO VÀ LIÊN KẾT SÀN THƯƠNG MẠI ĐIỆN TỬ CHO NHÀ SẢN XUẤT VÀ CỬA HÀNG* là công trình nghiên cứu của chính cá nhân tôi dưới sự hướng dẫn trực tiếp của giảng viên TS. Lê Thị Mỹ Hạnh.
- Tôi đã tự đọc nghiên cứu, dịch tài liệu và tổng hợp các kiến thức đã làm nên báo cáo này và đảm bảo không sao chép ở bất cứ đâu.
- Những lý thuyết trong luận văn đều được sử dụng tài liệu như tôi đã tham khảo ở phần tài liệu tham khảo đã có trong báo cáo. Nếu có vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Trần Ngọc Huy

Mục lục

Tóm tắt

Nhiệm vụ đồ án

Lời nói đầu **i**

Cam đoan **ii**

Mục lục **v**

Danh sách các bảng, hình vẽ **vii**

Danh sách viết tắt **viii**

Danh sách các cụm từ viết tắt **viii**

Mở đầu **1**

Sứ mệnh 1

Vấn đề 1

1 Cơ sở lý thuyết **3**

1.1 Môi trường 3

1.1.1 Trình duyệt [moz22] 3

1.1.2 Máy chủ 4

1.2 Nền tảng 5

1.2.1 Mạng máy tính 5

1.2.2 Công nghệ web 6

1.2.3 Phân tích và thiết kế hệ thống 8

1.2.4 Kiến trúc hướng dịch vụ 10

1.2.5 Hệ quản trị cơ sở dữ liệu 11

1.2.6 Quản lí mã nguồn 12

1.3	Công cụ	13
1.3.1	Node.js	13
1.3.2	PM2.js	14
1.3.3	React.js [Rea22]	14
1.3.4	Next.js	15
1.3.5	GraphQL	15
1.3.6	Apollo Server	17
1.3.7	MongoDB	17
1.4	Dịch vụ	18
1.4.1	Dịch vụ lưu trữ	18
1.4.2	Dịch vụ địa chỉ	18
1.4.3	Dịch vụ thư điện tử	18
1.4.4	Dịch vụ tên miền	19
1.4.5	Dịch vụ máy chủ ảo	19
2	Phân tích thiết kế	20
2.1	Mục tiêu	20
2.1.1	Mô tả chung	21
2.1.2	Yêu cầu tính năng	22
2.1.3	Yêu cầu giao diện	22
2.1.4	Yêu cầu hiệu suất	23
2.1.5	Yêu cầu thiết kế	23
2.1.6	Yêu cầu phi tính năng	23
2.1.7	Phụ lục	23
2.1.8	Vai trò	25
2.1.9	Hoạt động	25
2.1.10	Phân quyền	25
2.2	Phân tích hoạt động	29
2.2.1	Sơ đồ ca sử dụng	30
2.2.2	Sơ đồ hoạt động	34
2.2.3	Sơ đồ trạng thái	37
2.2.4	Sơ đồ tuần tự	38
2.3	Thiết kế cấu trúc	39
2.3.1	Sơ đồ lớp	39
2.3.2	Sơ đồ đóng gói	40
2.3.3	Sơ đồ khối thành phần	41
2.3.4	Sơ đồ phát hành	42

3	Triển khai và đánh giá	43
3.1	Triển khai	43
3.1.1	Nội quy, hướng dẫn, yêu cầu	43
3.1.2	Kiến thức khác	43
3.2	Mô tả chi tiết	45
3.2.1	Dữ liệu	46
3.2.2	API.	54
3.2.3	Quản lí mã nguồn	61
3.3	Kết quả đạt được	61
3.3.1	Dự án đã phát hành thực tế	62
3.3.2	Dự án đang thử nghiệm	62
3.4	Những vấn đề hạn chế	62
3.5	Hướng phát triển	62
	Kết luận	63
	Tài liệu tham khảo	64
	Phụ lục	

Danh sách bảng

3.1	Mô tả bảng ContractConsignment.	46
3.2	Mô tả bảng ContractLabor.	46
3.3	Mô tả bảng InteractionComment.	46
3.4	Mô tả bảng InteractionReaction.	46
3.5	Mô tả bảng Interaction.	46
3.6	Mô tả bảng NotificationMailer.	46
3.7	Mô tả bảng Notification.	47
3.8	Mô tả bảng Relationship.	47
3.9	Mô tả bảng TeamInvitation.	47
3.10	Mô tả bảng Team.	47
3.11	Mô tả bảng User.	47
3.12	Mô tả bảng View.	48
3.13	Mô tả bảng WorkPaid.	48
3.14	Mô tả bảng WorkType.	48
3.15	Mô tả bảng Work.	48
3.16	Mô tả bảng Banner.	48
3.17	Mô tả bảng Contact.	48
3.18	Mô tả bảng Page.	49
3.19	Mô tả bảng ProductAttributeValue.	49
3.20	Mô tả bảng ProductAttribute.	49
3.21	Mô tả bảng ProductBrand.	50
3.22	Mô tả bảng ProductCartItem.	50
3.23	Mô tả bảng ProductCart.	50
3.24	Mô tả bảng ProductCategory.	50
3.25	Mô tả bảng ProductDiscount.	50
3.26	Mô tả bảng ProductHashtag.	51
3.27	Mô tả bảng ProductOrderStatus.	51
3.28	Mô tả bảng ProductOrder.	51
3.29	Mô tả bảng ProductStock.	51
3.30	Mô tả bảng Product.	52
3.31	Mô tả bảng Translate.	52

3.32	Mô tả bảng UploadFile.	52
3.33	Mô tả bảng UploadImage.	52
3.34	Mô tả bảng FAQ.	52
3.35	Mô tả bảng Feature.	53
3.36	Mô tả bảng PostHashtag.	53
3.37	Mô tả bảng Post.	53
3.38	Mô tả bảng Service.	53
3.39	Mô tả bảng Testimonial.	53

Danh sách từ viết tắt

C C programming language. 12

CSS Cascading Style Sheets. 7

HTML HyperText Markup Language. 7

HTTP Hypertext Transfer Protocol. 7, 10, 23

MVC Model View Controller. 22

OCOPEE Hệ thống tạo và liên kết sản thương mại điện tử cho nhà sản xuất và cửa hàng. 1, 2

SQL Structured Query Language. 18

UML Unified Modeling Language. 8

VPS Virtual Private Server. 4, 5

Từ khóa tìm kiếm

latex official letterhead, table, list, font, cover letter

server database, dbms, microservice, monolithic, simple store, vps provider, domain provider, dns, ip, ci/cd, stateless, stateful, token, SHA, RSA, SSL, reverse proxy, load balance, design pattern, auto scale, network applications, thread, asynchronous, api, http

client client side, ssr, csr, html, dom, style sheet, react ui framework, react api state, react api client

Mở đầu

Sứ mệnh

Việt Nam là một nước có truyền thống nông nghiệp. Với điều kiện tài nguyên thiên nhiên thuận lợi, nguồn nhân lực dồi dào. Nông nghiệp đóng vai trò quan trọng trong sự phát triển kinh tế nước nhà.

Từ kỷ Hồng Bàng, trăm tộc Việt đã hợp sức tương trợ nhau. Nay, thời kì trăm hoa đua nở của nông nghiệp & công nghệ. OCOPEE² muốn tham gia vào xây dựng nền nông nghiệp Việt Nam. Với vai trò như một bộ phận thúc đẩy giao thương mua bán.

Vấn đề

Tìm kiếm đầu ra cho sản phẩm là vấn đề hết sức quan trọng đối với đơn vị sản xuất. Một trong những kênh bán hàng hiệu quả có thể kể đến đó là chương trình triển lãm hội chợ, phân phối tại các cửa hàng.

Tuy nhiên, đối với các doanh nghiệp khởi nghiệp. Sản phẩm được cải tiến cập nhật thông tin liên tục. Sản lượng không đủ lớn để phân phối rộng. Website là một công cụ tuyệt vời để đăng tải các thông tin cần thiết thay vì in trực tiếp lên bao bì của sản phẩm. Nó cũng giúp doanh nghiệp quảng bá sản phẩm. Giúp khách hàng tìm kiếm tra cứu thông tin hữu ích.

Về việc phát triển website. Do nguồn lực không đủ, rất nhiều hộ kinh doanh, doanh nghiệp khởi nghiệp không thể xây dựng một Website với các thông tin cơ bản. Việc cập nhật thường xuyên thông tin sản phẩm cũng mất rất nhiều thời gian công sức của chủ thể.

Về việc phân phối thông tin. Khi thông tin sản phẩm được đăng tải lên các trang thương mại điện tử khác hoặc trang web của các cửa hàng địa phương. Việc cập nhật và điều chỉnh giá là hết sức tốn thời gian.

Dropshipping là phương pháp thực hiện bán lẻ mà một cửa hàng không lưu giữ sản phẩm được bán trong kho của mình. Thay vào đó, khi một cửa hàng bán một sản phẩm cụ thể, không có sản phẩm lưu kho mà họ mua sản phẩm từ một bên thứ 3 và vận chuyển trực tiếp cho khách hàng. Kết quả là, những người bán hàng đó không bao giờ nhìn thấy sản phẩm hoặc xử lý sản phẩm.

Sự khác biệt lớn nhất giữa Dropshipping và các mô hình bán lẻ khác là các thương nhân bán hàng không cần kho hàng hoặc không có hàng lưu kho. Thay

²viết tắt của tên đề tài

vào đó, các thương nhân này mua hàng tồn kho khi cần thiết của một bên thứ 3 – thường là những người bán buôn hoặc nhà sản xuất – để hoàn thành đơn hàng của họ.

Giải quyết các vấn đề nêu trên có thể xem như việc tạo tự động các sàn thương mại điện tử với hình thức dropshipping cho nhà sản xuất hoặc xử lý đơn có lưu kho cho cửa hàng.

Giải pháp

OCOPÉE cung cấp công cụ tạo trang thông tin sản phẩm riêng cho từng nhà sản xuất. Và cũng có thể chia sẻ dữ liệu sản phẩm cho các cửa hàng. Các đơn vị phân phối và triển lãm có thể tích hợp trực tiếp dữ liệu gốc vào hệ thống của họ. Chủ động được phương pháp truyền thông mà không phụ thuộc vào sàn thương mại điện tử bên khác.

OCOPÉE xác định đối tượng trực tiếp phục vụ là khách mua hàng. Các tính năng nhắm đến hoặc ưu tiên phát triển đều lấy người dùng làm gốc.

- Khách hàng là người ủng hộ sử dụng nông sản chất lượng.
- Khách mua hàng tin tưởng chất lượng sản phẩm đăng tải trên hội chợ trực tuyến.
- Khách hàng cảm thấy tiện lợi trong quy trình mua hàng.
- Khách hàng được tư vấn thông tin đầy đủ về giá trị sử dụng của sản phẩm.
- Khách hàng dễ dàng tìm kiếm sản phẩm phù hợp với nhu cầu.

Nhà bán hàng là người trả tiền cho hệ thống. Thông qua việc phục vụ người mua nông sản. Hệ thống gián tiếp đem lại giá trị cho nhà bán hàng.

- Không gian mở rộng.
- Thời gian linh hoạt.
- Khách hàng tiềm năng.
- Phù hợp với quy mô, bảng

Định hướng tạo nên đột phá cho dự án sẽ được phát triển khi dữ liệu khách hàng và số lượng các nhà sản xuất đủ lớn. Khi đó vấn đề gần nhất cần giải quyết là hệ thống gợi ý giá bán và thu thập thông tin phản hồi từ khách hàng để cải thiện quá trình sản xuất.

Chương 1

Cơ sở lý thuyết

Cơ sở lý thuyết nêu ra các môi trường, công cụ, dịch vụ,... là nền tảng để xây dựng và phát triển hệ thống. Trong chương này trình bày các khái niệm, phân tích và lí do lựa chọn các cơ sở đó. Tổng quan của đề án phát triển dựa trên các công nghệ phát triển phần mềm. Web là một mô hình linh hoạt mà máy chủ có thể chạy trên nhiều môi trường. Ứng dụng phía máy khách có thể chạy đa hệ điều hành thông qua ứng dụng trình duyệt. Mô hình khách chủ giúp trải nghiệm người dùng liên mạch hơn, dữ liệu được đồng bộ đi khắp mọi nơi mặc kệ rào cản giữa các hệ điều hành đương thời. Tất nhiên, khi những tính năng đã ổn định và cần hiệu suất cao hơn. Phát triển ứng dụng gốc của hệ điều hành cho những tính năng đó cũng không quá khó khăn. Có thể nói môi trường web đã đạt đến sự cân bằng của dữ liệu và nền tảng. Nó không tạo ra các đột phá, các tính năng đặc thù như những ứng dụng gốc. Nhưng là một lựa chọn an toàn nếu bạn đang phát triển một hệ thống mới với sự thay đổi cập nhật liên tục của các nền tảng hoặc yêu cầu của khách hàng.

1.1 Môi trường

1.1.1 Trình duyệt [moz22]

Trình duyệt web đưa bạn đến mọi nơi trên internet, cho phép bạn xem văn bản, hình ảnh và video từ mọi nơi trên thế giới.

Web là một công cụ rộng lớn và mạnh mẽ. Trong một vài thập kỷ, Internet đã thay đổi cách chúng ta làm việc, cách chúng ta chơi và cách chúng ta tương tác với nhau. Tùy thuộc vào cách nó được sử dụng, nó là cầu nối giữa các quốc gia, thúc đẩy thương mại, nuôi dưỡng các mối quan hệ, thúc đẩy động cơ đổi mới của tương lai và chịu trách nhiệm về nhiều meme hơn những gì chúng ta biết phải làm.

Điều quan trọng là mọi người đều có quyền truy cập vào web, nhưng điều quan trọng là tất cả chúng ta hiểu các công cụ mà chúng ta sử dụng để truy

cập web. Chúng tôi sử dụng các trình duyệt web như Mozilla Firefox, Google Chrome, Microsoft Edge và Apple Safari mỗi ngày, nhưng chúng ta có hiểu chúng là gì và cách chúng hoạt động không? Trong một khoảng thời gian ngắn, chúng tôi đã hết ngạc nhiên trước khả năng gửi email cho ai đó trên khắp thế giới, đến sự thay đổi trong cách chúng tôi nghĩ về thông tin. Vấn đề không phải là bạn biết bao nhiêu nữa mà chỉ đơn giản là câu hỏi về trình duyệt hoặc ứng dụng nào có thể đưa bạn đến thông tin đó nhanh nhất.

Trình duyệt web hoạt động như thế nào?

Trình duyệt web đưa bạn đến bất cứ đâu trên internet. Nó lấy thông tin từ các phần khác của web và hiển thị trên máy tính để bàn hoặc thiết bị di động của bạn. Thông tin được truyền bằng Giao thức truyền siêu văn bản, xác định cách thức truyền văn bản, hình ảnh và video trên web. Thông tin này cần được chia sẻ và hiển thị ở định dạng nhất quán để mọi người sử dụng bất kỳ trình duyệt nào, ở bất kỳ đâu trên thế giới đều có thể xem thông tin.

Đáng buồn thay, không phải tất cả các nhà sản xuất trình duyệt đều chọn giải thích định dạng theo cùng một cách. Đối với người dùng, điều này có nghĩa là một trang web có thể trông và hoạt động khác nhau. Tạo tính nhất quán giữa các trình duyệt để mọi người dùng có thể tận hưởng Internet, bất kể trình duyệt họ chọn, được gọi là tiêu chuẩn web.

Khi trình duyệt web tìm nạp dữ liệu từ một máy chủ được kết nối internet, nó sử dụng một phần mềm được gọi là công cụ kết xuất để dịch dữ liệu đó thành văn bản và hình ảnh. Dữ liệu này được viết bằng ngôn ngữ đánh dấu siêu văn bản (HTML) và các trình duyệt web đọc mã này để tạo ra những gì chúng ta nhìn thấy, nghe thấy và trải nghiệm trên internet.

Siêu liên kết cho phép người dùng đi theo đường dẫn đến các trang hoặc trang web khác trên web. Mỗi trang web, hình ảnh và video đều có Bộ định vị tài nguyên thống nhất (URL) duy nhất của riêng nó, còn được gọi là địa chỉ web. Khi trình duyệt truy cập vào máy chủ để lấy dữ liệu, địa chỉ web sẽ cho trình duyệt biết nơi tìm kiếm từng mục được mô tả trong html, sau đó cho trình duyệt biết vị trí của nó trên trang web.

1.1.2 Máy chủ

A VPS, là một dạng máy chủ cho thuê. Trong đó, tài nguyên được tạo ra nhờ các công nghệ ảo hóa. Mỗi VPS được cài đặt trên một máy chủ vật lý. Nghĩa là một máy chủ vật lý có thể cài và chạy nhiều VPS. Mỗi VPS có hệ điều hành, ứng dụng, tài nguyên và không gian lưu trữ riêng được chia ra từ máy chủ vật lý.

Điều đó làm cho VPS hiệu suất và linh hoạt hơn các dạng lưu trữ phần mềm máy chủ khác.

Với sự linh hoạt và chi phí hợp lí. VPS cũng có gặp một số vấn đề trong đó đáng kể là khi người dùng truy cập tăng đột biến. Khả năng mở rộng hệ thống là hạn chế hơn so với dịch vụ sử dụng công nghệ đám mây.

1.2 Nền tảng

1.2.1 Mạng máy tính

Mạng máy tính thực hiện nhiệm vụ gì?

Mạng máy tính được tạo ra lần đầu tiên vào cuối những năm 1950 để sử dụng trong quân đội và quốc phòng. Mạng máy tính ban đầu được sử dụng để truyền dữ liệu qua đường dây điện thoại và bị hạn chế về mặt ứng dụng thương mại cũng như khoa học. Nhờ sự ra đời của công nghệ Internet, mạng máy tính đã trở thành một phần không thể thiếu đối với các doanh nghiệp.

Những giải pháp mạng thời nay không chỉ dừng lại ở khả năng kết nối. Chúng đóng vai trò rất quan trọng đối với quá trình chuyển đổi kỹ thuật số và thành công của doanh nghiệp hiện nay. Những khả năng cơ bản của mạng đã trở nên dễ lập trình hơn, tự động cũng như bảo mật hơn.

Kiến trúc mạng máy tính gồm những loại nào?

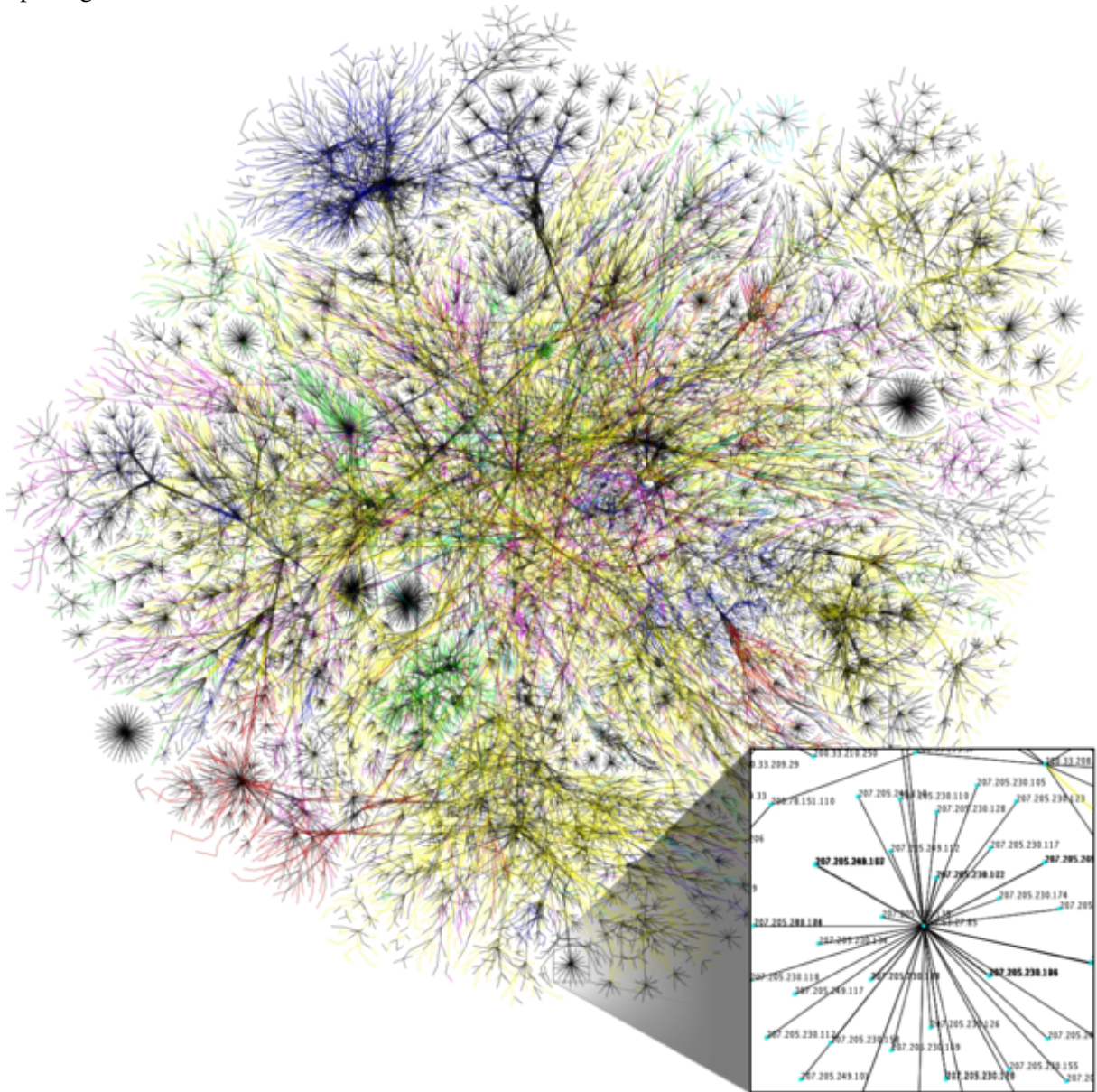
Thiết kế mạng máy tính gồm hai loại chính:

1. Kiến trúc khách - chủ Trong loại mạng máy tính này, các nút có thể là máy chủ hoặc máy khách. Nút máy chủ cung cấp các tài nguyên như bộ nhớ, công suất xử lý hoặc dữ liệu cho nút máy khách. Nút máy chủ cũng có thể quản lý hành vi của nút máy khách. Các máy khách có thể giao tiếp với nhau nhưng không chia sẻ tài nguyên. Ví dụ: một số thiết bị máy tính trong mạng doanh nghiệp lưu trữ dữ liệu và cài đặt cấu hình. Những thiết bị này là các máy chủ nằm trong mạng. Các máy khách có thể truy cập dữ liệu này bằng cách gửi một yêu cầu tới máy chủ.

2. Kiến trúc ngang hàng Trong kiến trúc Ngang hàng (P2P), các máy tính được kết nối có công suất và đặc quyền ngang nhau. Không có máy chủ tập trung để điều phối hoạt động. Mỗi thiết bị trong mạng máy tính có thể đóng vai trò là máy khách hoặc máy chủ. Mỗi thiết bị ngang hàng có thể chia sẻ một số tài nguyên của mình như bộ nhớ và công suất xử lý với toàn bộ mạng máy tính. Ví dụ: một

số công ty sử dụng kiến trúc P2P để lưu trữ các ứng dụng tiêu tốn nhiều bộ nhớ, chẳng hạn như kết xuất đồ họa 3D, trên nhiều thiết bị kỹ thuật số.

Hình 1.1: Bản đồ một phần của Internet dựa trên dữ liệu ngày 15 tháng 1 năm 2005 được tìm thấy trên opte.org



1.2.2 Công nghệ web

Công nghệ web là tổng hợp các công nghệ kỹ thuật phát triển web chạy trên môi trường máy chủ (mục 1.1.2) và trình duyệt web (mục 1.1.1)

Máy chủ web [ngi] lưu trữ và cung cấp nội dung của một trang web - chẳng hạn như văn bản, hình ảnh, video và dữ liệu ứng dụng - cho khách hàng yêu cầu.

Loại ứng dụng khách phổ biến nhất là chương trình trình duyệt web, trình duyệt yêu cầu dữ liệu từ trang web của bạn khi người dùng nhấp vào liên kết hoặc tải xuống tài liệu trên trang được hiển thị trong trình duyệt.

Máy chủ web giao tiếp với trình duyệt web bằng Giao thức truyền siêu văn bản (HTTP). Nội dung của hầu hết các trang web được mã hóa bằng Ngôn ngữ đánh dấu siêu văn bản (HTML). Nội dung có thể tĩnh (ví dụ: văn bản và hình ảnh) hoặc động (ví dụ: giá đã tính hoặc danh sách các mặt hàng mà khách hàng đã đánh dấu để mua). Để cung cấp nội dung động, hầu hết các máy chủ web hỗ trợ ngôn ngữ kịch bản phía máy chủ để mã hóa logic nghiệp vụ vào giao tiếp. Các ngôn ngữ được hỗ trợ phổ biến bao gồm Active Server Pages (ASP), Javascript, PHP, Python và Ruby.

Máy chủ web cũng có thể lưu nội dung vào bộ nhớ cache để tăng tốc độ phân phối nội dung thường được yêu cầu. Quá trình này còn được gọi là tăng tốc web.

Máy chủ web có thể lưu trữ một trang web hoặc nhiều trang web sử dụng cùng một tài nguyên phần mềm và phần cứng, được gọi là lưu trữ ảo. Máy chủ web cũng có thể giới hạn tốc độ phản hồi đối với các máy khách khác nhau để ngăn một máy khách thống trị các tài nguyên được sử dụng tốt hơn để đáp ứng các yêu cầu từ một số lượng lớn máy khách.

Mặc dù máy chủ web thường lưu trữ các trang web có thể truy cập được trên Internet, chúng cũng có thể được sử dụng để giao tiếp giữa máy khách web và máy chủ trong các mạng cục bộ chẳng hạn như mạng nội bộ của công ty. Một máy chủ web thậm chí có thể được nhúng vào một thiết bị như một máy ảnh kỹ thuật số để người dùng có thể giao tiếp với thiết bị thông qua bất kỳ trình duyệt Web thông dụng nào.

HyperText Markup Language (HTML) là một trong những thành phần cơ bản nhất để dựng thành web. Nó định nghĩa và cấu trúc nội dung của web.

"Siêu văn bản" đề cập đến các liên kết kết nối các trang web với nhau, trong một trang web hoặc giữa các trang web. Liên kết là một phần cơ bản của Web. Bằng cách tải nội dung lên Internet và liên kết nội dung đó với các trang do người khác tạo, bạn đã trở thành người tham gia trong World Wide Web.

Cascading Style Sheets (CSS) là một ngôn ngữ định kiểu sử dụng để mô tả các hiển thị của ngôn ngữ đánh dấu như HTML và các ngôn ngữ tương tự. CSS mô tả cách mà các thành phần sẽ được hiển thị trên màn hình, trên giấy, khi nói và trên các phương tiện khác. CSS là một trong những ngôn ngữ chính của web mở và cũng là tiêu chuẩn trên các trình duyệt theo thông số W3C. Trước đây, việc phát triển các phần khác nhau của đặc tả CSS đã được thống nhất, cho phép tạo

phiên bản cho các đề xuất mới nhất. Bạn có thể đã nghe nói về CSS1, CSS2.1, CSS3. Tuy nhiên, CSS4 chưa bao giờ trở thành phiên bản chính thức.

1.2.3 Phân tích và thiết kế hệ thống

Unified Modeling Language (UML) là một mô hình trực quan hóa nhằm:

- Mô hình hóa nghiệp vụ và các tiến trình tương tự.
- Phân tích, thiết kế, triển khai các hệ thống phần mềm.

Unified Modeling Language (UML) là một ngôn ngữ phổ biến cho việc phân tích nghiệp vụ, kiến trúc phần mềm. Lập trình viên sử dụng nó để mô tả, yêu cầu, thiết kế, viết tài liệu cho: nghiệp vụ có sẵn hoặc yêu cầu nghiệp vụ mới, cấu trúc và hành vi của các hệ thống phần mềm.

UML có thể được áp dụng cho các lĩnh vực ứng dụng đa dạng (ví dụ: ngân hàng, tài chính, internet, hàng không vũ trụ, chăm sóc sức khỏe, v.v.) Nó có thể được sử dụng với tất cả các phương pháp phát triển phần mềm đối tượng và thành phần chính và cho các nền tảng triển khai khác nhau.

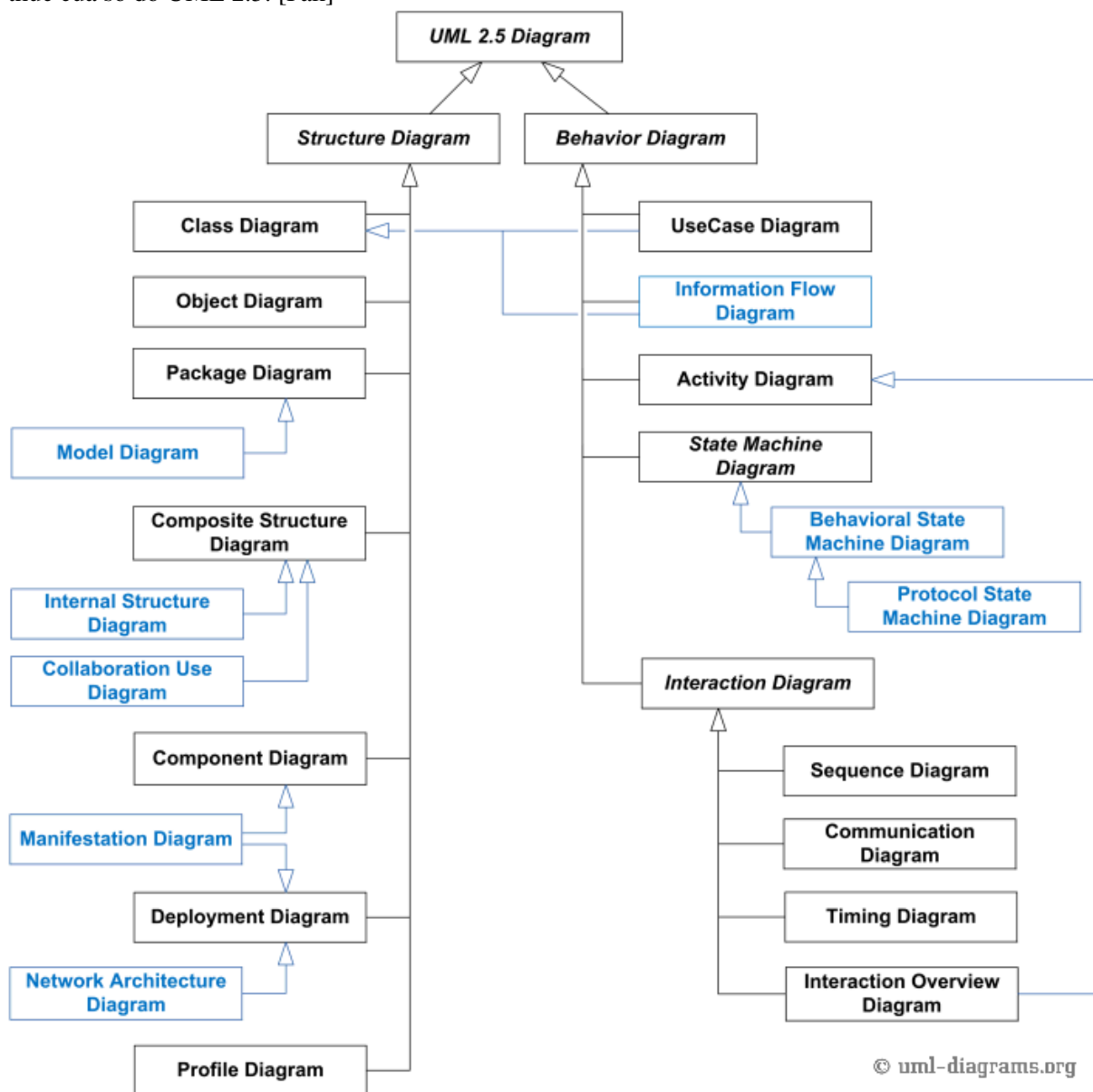
UML là một phương pháp mô hình hóa phần mềm, chứ không phải là quy trình phát triển phần mềm.

- cung cấp hướng dẫn về thứ tự các hoạt động của nhóm
- chỉ định những gì tạo tác nên được phát triển
- chỉ đạo nhiệm vụ của các nhà phát triển cá nhân và toàn bộ nhóm
- đưa ra các tiêu chí để giám sát và đo lường các sản phẩm và hoạt động của dự án.

UML chủ động độc lập với quy trình và có thể được áp dụng trong mọi bối cảnh của các quy trình khác nhau. Tuy nhiên, nó vẫn phù hợp nhất cho các quy trình phát triển theo hướng nhanh chóng đưa dự án vào sử dụng, lặp đi lặp lại và tăng dần.

Khái niệm hướng đối tượng [Han22] được xây dựng trên nền tảng của khái niệm lập trình có cấu trúc và sự trừu tượng hóa dữ liệu. Sự thay đổi căn bản ở chỗ, một chương trình hướng đối tượng được thiết kế xoay quanh dữ liệu mà chúng ta có thể làm việc trên đó, hơn là theo bản thân chức năng của chương trình. Điều này hoàn toàn tự nhiên một khi chúng ta hiểu rằng mục tiêu của chương trình là xử lý dữ liệu. Suy cho cùng, công việc mà máy tính thực hiện vẫn thường được gọi là xử lý dữ liệu. Dữ liệu và thao tác liên kết với nhau ở một mức cơ bản (còn có thể gọi là mức thấp), mỗi thứ đều đòi hỏi ở thứ kia có mục tiêu cụ

Hình 1.2: Tổng quan về sơ đồ UML 2.5. Lưu ý, các mục có màu xanh lam không thuộc phân loại chính thức của sơ đồ UML 2.5. [Fak]



thể, các chương trình hướng đối tượng làm tường minh mối quan hệ này. Lập trình hướng đối tượng (Object Oriented Programming - gọi tắt là OOP) hay chi tiết hơn là Lập trình định hướng đối tượng, chính là phương pháp lập trình lấy đối tượng làm nền tảng để xây dựng thuật giải, xây dựng chương trình. Thực chất đây không phải là một phương pháp mới mà là một cách nhìn mới trong việc lập trình. Để phân biệt, với phương pháp lập trình theo kiểu cấu trúc mà chúng ta quen thuộc trước đây, hay còn gọi là phương pháp lập trình hướng thủ tục (Procedure-Oriented Programming), người lập trình phân tích một nhiệm vụ lớn thành nhiều công việc nhỏ hơn, sau đó dần dần chi tiết, cụ thể hoá để được các vấn đề đơn giản, để tìm ra cách giải quyết vấn đề dưới dạng những thuật

giải cụ thể rõ ràng qua đó dễ dàng minh hoạ bằng ngôn ngữ giải thuật (hay còn gọi các thuật giải này là các chương trình con). Cách thức phân tích và thiết kế như vậy chúng ta gọi là nguyên lý lập trình từ trên xuống (top-down), để thể hiện quá trình suy diễn từ cái chung cho đến cái cụ thể. Các chương trình con là những chức năng độc lập, sự ghép nối chúng lại với nhau cho chúng ta một hệ thống chương trình để giải quyết vấn đề đặt ra. Chính vì vậy, cách thức phân tích một hệ thống lấy chương trình con làm nền tảng, chương trình con đóng vai trò trung tâm của việc lập trình, được hiểu như phương pháp lập trình hướng về thủ tục. Tuy nhiên, khi phân tích để thiết kế một hệ thống không nhất thiết phải luôn luôn suy nghĩ theo hướng “làm thế nào để giải quyết công việc”, chúng ta có thể định hướng tư duy theo phong cách “với một số đối tượng đã có, phải làm gì để giải quyết được công việc đặt ra” hoặc phong phú hơn, “làm cái gì với một số đối tượng đã có đó”, từ đó cũng có thể giải quyết được những công việc cụ thể. Với phương pháp phân tích trong đó đối tượng đóng vai trò trung tâm của việc lập trình như vậy, người ta gọi là nguyên lý lập trình từ dưới lên (Bottom-up). Lập trình hướng đối tượng liên kết cấu trúc dữ liệu với các thao tác, theo cách mà tất cả thường nghĩ về thế giới quanh mình. Chúng ta thường gán một số các hoạt động cụ thể với một loại hoạt động nào đó và đặt các giả thiết của mình trên các quan hệ đó.

1.2.4 Kiến trúc hướng dịch vụ

Kiến trúc hướng dịch vụ là một khái niệm để chỉ các hệ thống có kiến trúc chia nhỏ ra thành các phần nhỏ và đủ để giải quyết một vấn đề một cách độc lập. Kiến trúc này ra đời một cách tự nhiên khi các kỹ sư cố gắng làm cho các thành phần ít phụ thuộc vào nhau hơn và từng thành phần được cấu trúc chắc chắn, ổn định và hiệu suất cao.

Đối với một hệ thống máy chủ lớn, một dự án được chia thành rất nhiều thành phần để giao cho từng đội nhóm phát triển. Các hệ thống này đa số giao tiếp với nhau thông qua giao thức HTTP.

Kiến trúc hướng dịch vụ ngày một được sử dụng rộng rãi là một điều tất nhiên khi mà nghiệp vụ của các hệ thống đương đại ngày một phức tạp. Đội ngũ phát triển hệ thống không chỉ dừng lại ở vài ngàn người mà trong đó, mỗi thành phần được phát triển dựa trên hàng trăm công nghệ đồ sộ trước đó. Làm cho độ phức tạp của dự án tăng lên quá mức kiểm soát của một người hoặc một nhóm người.

Kiến trúc hướng dịch vụ có thể xem như một chức năng của một khối đối tượng. Các đối tượng này khi xem như một khối, thì sẽ sinh ra các phương thức tương ứng. Cách thiết kế này có thể giúp cho dịch vụ được phát triển một cách rõ ràng và hiệu suất hơn.

Kiến trúc hướng dịch vụ cũng chỉ phù hợp với một số quy mô dự án. Sử dụng kiến trúc không đúng với quy mô sẽ làm giảm tiến độ phát triển. Ảnh hưởng đến chất lượng hệ thống và khả năng nâng cấp sau này.

Để đáp ứng cho nhu cầu phân tách, một số khái niệm, công nghệ, phương pháp được triển khai ứng dụng. Cần hiệu và sử dụng các phương pháp để đảm bảo khả năng bảo mật cho hệ thống.

1.2.5 Hệ quản trị cơ sở dữ liệu

Cơ sở dữ liệu NoSQL (phi quan hệ) hoạt động như thế nào? [Amab]

Cơ sở dữ liệu NoSQL sử dụng nhiều mô hình dữ liệu để truy cập và quản lý dữ liệu. Các loại cơ sở dữ liệu này được tối ưu hóa dành riêng cho các ứng dụng yêu cầu mô hình dữ liệu linh hoạt có lượng dữ liệu lớn và độ trễ thấp, có thể đạt được bằng cách giảm bớt một số hạn chế về tính nhất quán của dữ liệu của các cơ sở dữ liệu khác.

- Trong cơ sở dữ liệu quan hệ, hồ sơ về một cuốn sách thường được phân tách (hay còn gọi là "chuẩn hóa") và lưu trữ trong các bảng tách biệt nhau, còn mỗi quan hệ được quy định bằng các ràng buộc khóa ngoại và khóa chính. Trong ví dụ này, bảng Sách có các cột cho ISBN, Tên sách và Số phiên bản, bảng Tác giả có các cột cho ID tác giả và Tên tác giả và cuối cùng, bảng Tác giả–ISBN có các cột cho ID tác giả và ISBN. Mô hình quan hệ được thiết kế để cho phép các cơ sở dữ liệu này thực thi tính toàn vẹn tham chiếu giữa nhiều bảng trong cơ sở dữ liệu, được chuẩn hóa để giảm dư thừa và thường được tối ưu hóa cho mục đích lưu trữ.
- Trong cơ sở dữ liệu NoSQL, hồ sơ về một cuốn sách thường được lưu trữ dưới dạng văn bản JSON. Với từng quyển sách, mục, ISBN, Tên sách, Số phiên bản, Tên tác giả và ID tác giả được lưu trữ dưới dạng thuộc tính trong một văn bản duy nhất. Trong mô hình này, dữ liệu được tối ưu hóa cho việc phát triển trực quan và khả năng thay đổi quy mô theo chiều ngang.

Vì sao bạn nên sử dụng cơ sở dữ liệu NoSQL?[Amab]

Cơ sở dữ liệu NoSQL là lựa chọn cực kỳ thích hợp cho nhiều ứng dụng hiện đại, ví dụ như di động, web và trò chơi đòi hỏi phải sử dụng cơ sở dữ liệu cực kỳ thiết thực, linh hoạt, có khả năng thay đổi quy mô và hiệu năng cao để đem đến cho người dùng trải nghiệm tuyệt vời.

- **Cơ sở dữ liệu NoSQL:** là lựa chọn cực kỳ thích hợp cho nhiều ứng dụng hiện đại, ví dụ như di động, web và trò chơi đòi hỏi phải sử dụng cơ sở dữ

liệu cực kỳ thiết thực, linh hoạt, có khả năng thay đổi quy mô và hiệu năng cao để đem đến cho người dùng trải nghiệm tuyệt vời.

- **Khả năng thay đổi quy mô:** Cơ sở dữ liệu NoSQL thường được thiết kế để tăng quy mô bằng cách sử dụng các cụm phần cứng được phân phối thay vì tăng quy mô bằng cách bổ sung máy chủ mạnh và tốn kém. Một số nhà cung cấp dịch vụ đám mây xử lý các hoạt động này một cách không công khai dưới dạng dịch vụ được quản lý đầy đủ.
- **Hiệu năng cao:** Cơ sở dữ liệu NoSQL được tối ưu hóa theo các mô hình dữ liệu cụ thể và các mẫu truy cập giúp tăng hiệu năng cao hơn so với việc cố gắng đạt được mức độ chức năng tương tự bằng cơ sở dữ liệu quan hệ.
- **Cực kỳ thiết thực:** Cơ sở dữ liệu NoSQL cung cấp các API và kiểu dữ liệu cực kỳ thiết thực được xây dựng riêng cho từng mô hình dữ liệu tương ứng.

1.2.6 Quản lý mã nguồn

Git là một hệ thống kiểm soát phiên bản phân tán. Một mã nguồn mở miễn phí được thiết kế để xử lý mọi thứ từ các dự án nhỏ đến rất lớn với tốc độ và hiệu quả.

Git rất dễ học và đặc biệt hiệu suất. Nó vượt trội hơn các công cụ SCM như Subversion, CVS, Perforce và ClearCase với các tính năng như tối ưu phân nhánh cục bộ, thực hiện đóng gói và nhiều quy trình làm việc.

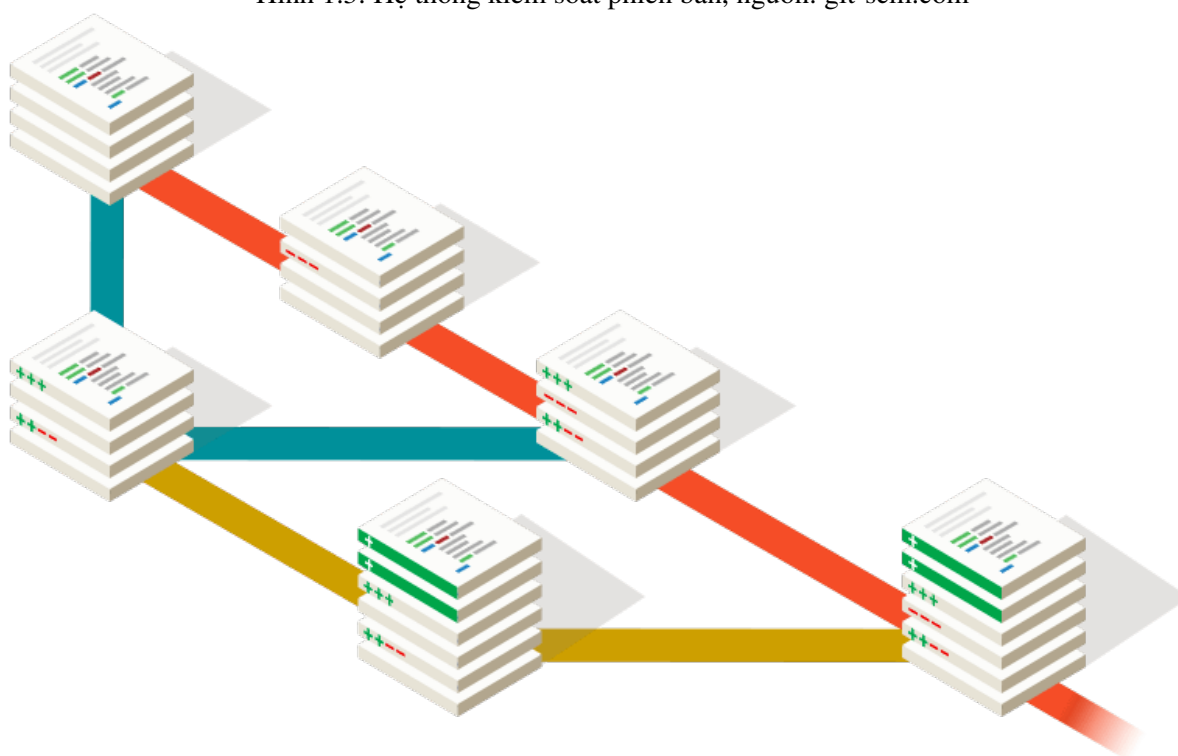
Phân nhánh và Hợp nhất

Tính năng Git thực sự khác biệt với SCM là mô hình phân nhánh của nó. Git cho phép và khuyến khích bạn có nhiều chi nhánh cục bộ có thể hoàn toàn độc lập với nhau. Việc tạo, hợp nhất và xóa các dòng phát triển đó mất vài giây.

Nhỏ và nhanh

Git rất nhanh. Với Git, gần như tất cả các hoạt động đều được thực hiện cục bộ, mang lại lợi thế về tốc độ rất lớn trên các hệ thống tập trung liên tục phải giao tiếp với máy chủ ở đâu đó. Git được xây dựng để hoạt động trên nhân Linux, có nghĩa là nó phải xử lý hiệu quả các kho lưu trữ lớn ngay từ đầu. Git được viết bằng C, giảm chi phí thời gian chạy liên quan đến các ngôn ngữ cấp cao hơn. Tốc độ và hiệu suất là mục tiêu thiết kế chính của Git ngay từ đầu.

Hình 1.3: Hệ thống kiểm soát phiên bản, nguồn: git-scm.com



Được phân phối

Một trong những tính năng hay nhất của bất kỳ hệ phân tán SCM nào, bao gồm Git, là nó được phân tán. Điều này có nghĩa là thay vì thực hiện "checkout" hiện tại của mã nguồn, bạn thực hiện "clone" toàn bộ kho lưu trữ.

1.3 Công cụ

1.3.1 Node.js

Là một môi trường chạy JavaScript theo hướng sự kiện bất đồng bộ, Node.js được thiết kế để xây dựng các ứng dụng mạng có thể mở rộng.

Một trong những lý do khiến Node.js trở nên phổ biến là nhờ sự phổ biến của Javascript. Tại thời điểm Javascript phát triển mạnh mẽ ở phía trình duyệt máy khách. Thì Node.js với hàng loạt các thư viện mạnh mẽ đã mang đến trải nghiệm phát triển liền mạch. Ngoài sự linh hoạt và quen thuộc đã nêu, hiệu suất của Node.js cũng khá tốt để phát triển một hệ thống máy chủ dẻo dai hơn.

1.3.2 PM2.js

Để phát hành ứng dụng máy chủ web lên một máy chủ ảo. Cần khởi chạy pm2 như một phần mềm quản lý tiến trình. pm2 giúp ta có thể khởi động lại, xóa hoặc chạy các tiến trình khi mà phiên kết nối giữa máy chủ và máy khách bị đóng.

Thay thế pm2, có thể sử dụng thư viện có sẵn của Node.js để lưu trữ mã tiến trình và khởi động hoặc xóa nó khi cần. Nhưng điều này tốn thời gian và độ hoàn thiện không đủ để thay thế pm2 khi nó còn là một mã nguồn mở miễn phí.

Khi thực hiện triển khai một hệ thống hướng dịch vụ với hàng trăm tiến trình. Nhu cầu kiểm tra theo dõi tình trạng các tiến trình tại một đầu mỗi tổng là hết sức cần thiết. PM2 cũng cung cấp gói tính phí để giải quyết vấn đề này. Tuy nhiên, so với các dịch vụ quản lý tiến trình khác thì cần phải cân nhắc.

Phát triển một hệ thống quản lý tiến trình tập trung cần hiểu các kiến thức về hệ thống phân tán. Điều này cũng không đơn giản khi các tiến trình hoặc các máy chủ ảo liên tục bị quá tải và khởi động lại. Lúc này các phương án dự phòng và khôi phục cần được thiết đặt và sử dụng.

1.3.3 React.js [Rea22]

Một thư viện JavaScript xây dựng giao diện người dùng

Tính trực quan

React giúp tạo các UI tương tác một cách dễ dàng. Thiết kế các khung nhìn đơn giản cho từng trạng thái trong ứng dụng của bạn, và React sẽ cập nhật và kết xuất đúng các thành phần phù hợp khi dữ liệu thay đổi.

Việc khai báo các khung nhìn tường minh sẽ khiến cho mã của bạn dễ sử dụng hơn và dễ dàng gỡ lỗi hơn.

Dựa trên khối cơ bản

Xây dựng các khối cơ bản và quản lý các trạng thái của riêng chúng, sau đó kết hợp chúng để tạo các giao diện người dùng phức tạp.

Vì khối cơ bản được viết bằng JavaScript thay vì các mẫu, bạn có thể dễ dàng truyền dữ liệu đa dạng qua ứng dụng của mình và tránh thao tác với DOM.

Dễ học, dễ làm

React không đưa ra các giả định về phần kỹ năng công nghệ của bạn, vì vậy bạn có thể phát triển các tính năng mới trong React mà không cần viết lại mã hiện có.

React cũng có thể kết xuất trên máy chủ bằng Node và xây dựng ứng dụng di động bằng cách sử dụng React Native.

1.3.4 Next.js

Next.js là một sự kết hợp giữa việc kết xuất khối cơ bản của React.js ở môi trường trình duyệt với môi trường máy chủ một cách linh hoạt. Next.js là một máy chủ Node.js sử dụng React.js nên cung cấp khả năng điều hướng. Đọc dữ liệu và truyền tải dữ liệu giữa hai môi trường một cách hết sức linh hoạt và mạnh mẽ. Một khối cơ sở của React.js được viết ra trong Next.js liền mạch đến mức nếu lập trình viên không nắm rõ kiến thức nền tảng và hiểu về Next.js sẽ không kiểm soát được rằng một cuộc dòng lệnh đó được thực thi và kết xuất ở máy chủ hay trình duyệt máy khách. Các phiên bản sau này của Next.js ra đời hạn chế sự liền mạch giữa hai môi trường để giảm bớt sự nhập nhằng cho lập trình viên. Next.js thực sự mạnh mẽ nhưng cũng không phải là lựa chọn duy nhất. Có rất nhiều thư viện tương tự ra đời và còn hiệu quả hơn cả Next.js.

Next.js cung cấp đầy đủ các khả năng để tạo ra một khối máy chủ web hoàn chỉnh. Nhưng đa số người dùng sử dụng nó như một dịch vụ kết xuất giao diện trong hệ thống kiến trúc hướng dịch vụ của họ.

Bởi khả năng kết xuất, điều hướng và truyền tải dữ liệu giữa hai môi trường linh hoạt. Trải nghiệm người dùng ở trình duyệt được cải thiện rõ rệt. Những vấn đề về tối ưu, cập nhật thông tin, gọi dữ liệu kết xuất phức tạp được giải quyết dễ dàng hơn.

1.3.5 GraphQL

REST API [Amaa]

REST là gì? Chuyển trạng thái đại diện (REST) là một kiến trúc phần mềm quy định các điều kiện về cách thức hoạt động của API. REST ban đầu được tạo ra như một hướng dẫn để quản lý giao tiếp trên một mạng phức tạp như Internet. Bạn có thể sử dụng kiến trúc dựa trên REST để hỗ trợ giao tiếp hiệu suất cao và đáng tin cậy trên quy mô lớn. Bạn có thể dễ dàng triển khai và sửa đổi REST, mang lại khả năng hiển thị và tính di động đa nền tảng cho bất kỳ hệ thống API nào.

GraphQL API

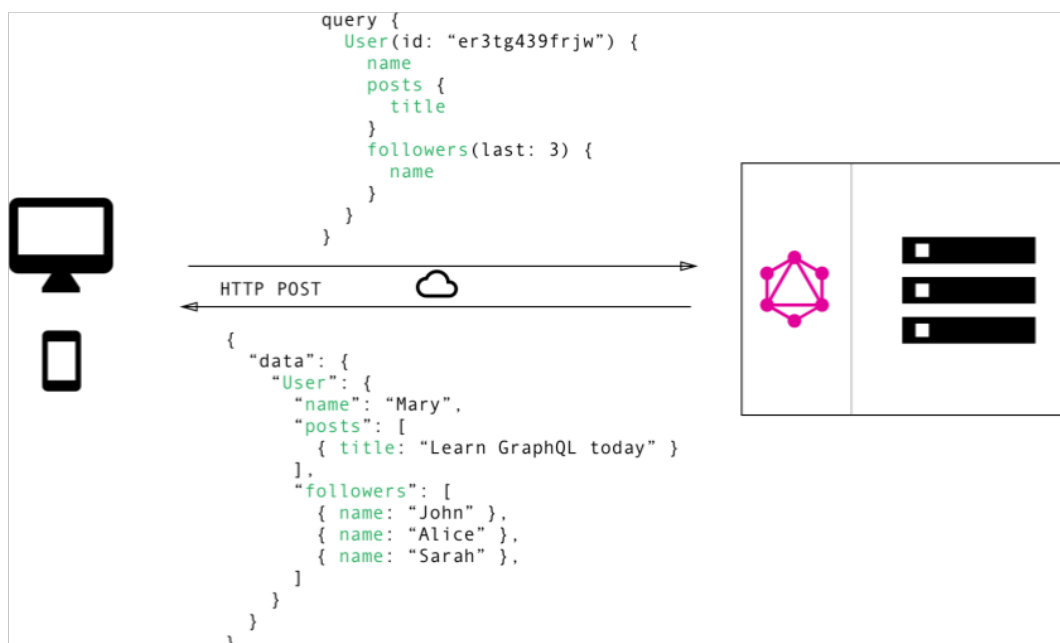
GraphQL là ngôn ngữ truy vấn cho các API và thời gian thực để thực hiện các truy vấn đó với dữ liệu hiện có của bạn. GraphQL cung cấp mô tả đầy đủ và dễ hiểu về dữ liệu trong API, cung cấp cho khách hàng sức mạnh để yêu cầu chính

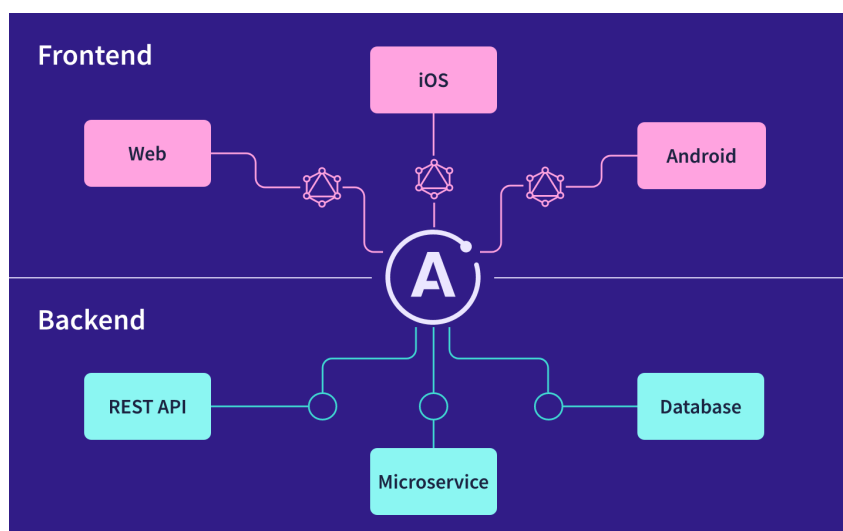
Hình 1.4: REST API



xác những gì họ cần và không cần gì hơn, giúp việc phát triển API dễ dàng hơn theo thời gian và cho phép các công cụ mạnh mẽ dành cho nhà phát triển.

Hình 1.5: GraphQL API





Hình 1.6:

1.3.6 Apollo Server

Nhờ sức mạnh của ngôn ngữ truy vấn dành cho API là GraphQL. Apollo Server thiết kế ra để dễ dàng khởi tạo khung sườn cho dự án sử dụng GraphQL để phân tích cú pháp và xử lý nghiệp vụ dễ dàng hơn.

Khác với REST API và cấu trúc MVC thông thường. GraphQL tổ chức xử lý API theo kiểu quy nạp. Cú pháp được phân tích thành cấu trúc cây và chia ra xử lý sau đó quy nạp ngược lên để trả kết quả về phía người dùng.

GraphQL cũng cho phép định kiểu các chức năng xử lý và hoạt động truy cập đến đối tượng nên có thể chia cấu trúc dự án hướng đối tượng với các hàm xử lý rất nhỏ. Mỗi hàm chức năng như vậy đều có thể là một nút trên cây truy vấn. Điều này giúp cho việc truy vấn và khởi chạy các hàm, liên kết quy nạp dữ liệu diễn ra hết sức linh hoạt. Nghĩa là bạn có thể lấy duy nhất một bảng ghi sau đó xử lý. Hoặc lấy danh sách bản ghi bao gồm việc xử lý đều được mà không cần phát triển một "Controller" mới như mô hình MVC thông thường.

Một số người nhờ vào sự linh hoạt khi thiết kế các nút xử lý của GraphQL để xử dụng chúng như một Gateway cho hệ thống có sẵn của họ. Tạo nên một bộ API thống nhất trên một tài liệu lập trình duy nhất giúp cho các đội ngũ có thể giao tiếp với nhau dễ dàng hơn và tài liệu được cập nhật triệt để hơn.

1.3.7 MongoDB

MongoDB là một mã nguồn mở để quản lý cơ sở dữ liệu dạng tài liệu thay vì dạng bảng như SQL. MongoDB được phát triển để mở rộng theo chiều ngang làm cho định kiểu dữ liệu trở nên linh hoạt hơn. Phát hành năm 2007, MongoDB đã được tiếp nhận và phổ biến khắp nơi trên thế giới.

Thay vì lưu dữ liệu vào bảng theo hàng hoặc cột như SQL, mỗi bản ghi trong MongoDB được mô tả như một tài liệu BSON, dạng mã hóa nhị phân dữ liệu. Ứng dụng có thể duyệt qua những thông tin theo định dạng JSON.

Lưu trữ database dạng tài liệu như vậy rất linh hoạt, chúng cho phép tùy biến trong cấu trúc của mỗi tài liệu và chỉ lưu một phần của tài liệu. Một tài liệu có thể có các dạng dữ liệu khác nhau. Từng trường dữ liệu cũng có khả năng ràng buộc giá trị như một cột của SQL. Ta cũng có thể đánh indexed để gia tăng tốc độ truy vấn.

No-SQL với những ưu điểm của nó rất thích hợp khi dùng để phát triển một dự chưa rõ ràng. Với một dự án quá lớn. Không nên chỉ sử dụng một loại quản trị cơ sở dữ liệu. Từng trường hợp cụ thể. Với kiến thức về ưu nhược điểm của từng loại ta nên lựa chọn phương án phù hợp hơn là so sách mà bỏ các vấn đề giải quyết ngay trong dự án hiện tại.

1.4 Dịch vụ

Luận án sử dụng một số dịch vụ của bên thứ ba để lược bớt sự phức tạp và khối lượng của dự án. Các vấn đề này xử lý rất đơn giản khi sử dụng dịch vụ bên ngoài. Sự phụ thuộc là không đáng kể. Cần đưa ra các phương án dự phòng khi phát triển. Cần phát triển các cầu nối đủ linh hoạt để sử dụng nhiều dịch vụ và lựa chọn giữa các nhà cung cấp trở nên dễ dàng hơn.

1.4.1 Dịch vụ lưu trữ

Dịch vụ lưu trữ được sử dụng tại luận án là dịch vụ lưu trữ tài nguyên đơn giản nhất. Chỉ có chức năng đăng tải và truy cập. Không hỗ trợ tìm kiếm, nén tệp, tối ưu hình ảnh.

1.4.2 Dịch vụ địa chỉ

Dịch vụ địa chỉ cung cấp danh sách tỉnh thành, quận huyện và các thông tin địa danh cấp nhỏ hơn để khách hàng khai báo thông tin địa chỉ. Các dịch vụ cung cấp thông tin đầy đủ, chính xác và được cập nhật thường xuyên.

1.4.3 Dịch vụ thư điện tử

Dịch vụ thư điện tử sử dụng là máy chủ thư điện tử. Việc cấu hình máy chủ thư điện tử và chi phí duy trì một máy chủ độc lập là không cần thiết.

Các nhà cung cấp dịch vụ thư điện tử cũng cung cấp giao diện người dùng đầy đủ tính năng giúp duyệt thư dễ dàng hơn.

1.4.4 Dịch vụ tên miền

Dịch vụ máy chủ tên miền giúp phân giải tên miền đến một địa chỉ ip cụ thể. Máy chủ tên miền sử dụng không bao gồm chứng thực SSL và cân bằng tải.

Máy chủ tên miền [Wik] chứa thông tin lưu trữ về một số tên miền. Hệ thống phân giải tên miền được vận hành bởi hệ thống dữ liệu phân tán, dạng Client-Server. Các node của hệ thống dữ liệu này là các máy chủ tên miền. Mỗi một tên miền sẽ có ít nhất một máy chủ tên miền chứa thông tin về tên miền đó. Các thông tin về máy chủ tên miền sẽ được lưu trữ trong các zone của tên miền. Có hai dạng Name Server là Primary và Secondary. Một Client sẽ ưu tiên hỏi Primary trước và thử lại với Secondary nếu Primary không thể trả lời thông tin về tên miền đó trong thời gian quy định.

1.4.5 Dịch vụ máy chủ ảo

Đề tài không bao gồm việc cấu hình mạng, phần cứng của máy chủ. Để triển khai mã nguồn dự án lên môi trường thực tế. Đề tài mô tả việc cấu hình mã nguồn dự án lên một máy chủ ảo (định nghĩa máy chủ mục 1.1.2).

Chương 2

Phân tích thiết kế

2.1 Mục tiêu

Đặc tả các yêu cầu nghiệp vụ của hệ thống, giúp quá trình phân tích thiết kế trở nên rõ ràng hơn. Nếu được tiếp tục phát triển bởi một đội nhóm. Thì tài liệu đặc tả là một phần không thể thiếu giúp lập trình viên lưu trữ nghiệp vụ, cấu trúc dự án cũ và lên kế hoạch phát triển tính năng cho tối ưu.

Xác định tài nguyên, nguyên liệu, nguồn lực của dự án (Xem mục 2.1). Xác định các yêu cầu đầu cuối. Đặc tả để tìm kiếm phương pháp (Xem mục ??). Từ tài nguyên nguồn lực và yêu cầu đặc tả. Ta có thể lên kế hoạch triển khai xây dựng phù hợp (Xem phân tích 2.2 và thiết kế 2.3).

Nội dung của phần phân tích 2.2 đã nêu rõ các vai trò và phân quyền của các loại người dùng trong hệ thống.

Chúng ta cần xác định rõ đối tượng cần phục vụ và đối tượng là khách hàng của hệ thống. Thông qua đối tượng cần phục vụ là người mua hàng như đã nêu ở mục tiêu. Cần có kế hoạch mở rộng đối tượng và cải tiến chất lượng hệ thống công nghệ. Dữ liệu người mua hàng, niềm tin của người mua hàng trên hệ thống được coi là nguồn lực tài nguyên. Hệ thống công nghệ là công vụ chuyển hóa tài nguyên đó thành giá trị doanh nghiệp. Tức là khả năng tạo đầu ra cho doanh nghiệp đối tác.

Các nhà sản xuất được coi là khách hàng của giá trị mà hệ thống công nghệ tạo ra. Họ không phải là khách mua hàng trên hệ thống. Họ đang mua chỗ đứng trên hệ thống để bán hàng của chính họ. Như vậy, cần xác định kế hoạch chăm sóc và tiếp cận các nhà sản xuất phù hợp.

2.1.1 Mô tả chung

Mô tả tổng quan về hoạt động của hệ thống. Ta bắt đầu từ việc liên tưởng đến mô hình bán hàng truyền thống. Khi mà các trang thương mại điện tử chưa ra đời. Người sản xuất sản phẩm đi tiếp thị và quảng bá sản phẩm của mình dưới nhiều hình thức khác nhau mà phổ biến nhất là các cửa hàng và đại lí. Ngành vận chuyển phát triển cùng với sự ra đời của các trang sàn thương mại điện tử giúp thay đổi quy trình bán hàng. Thay đổi cách tiếp thị và quảng bá dựa trên nền tảng tương tác trên không gian mạng.

Ngày nay, hình thức đó vẫn đang phát triển và vận hành rất tốt. Các nhà sản xuất trên khắp cả nước có thể tự đăng tải quảng bá sản phẩm của mình trên các sàn thương mại điện tử. Có rất nhiều công cụ ra đời để phụ giúp cho việc đó.

Trong bối cảnh đó, dự án ra đời để giải quyết vấn đề tạo trang thương mại điện tử cho chính nhà cung cấp. Và liên kết sản phẩm để phân phối thông tin giữa các nhà bán hàng dùng chung hệ thống dự án này. Khi một người bắt đầu đăng kí hộ kinh doanh hoặc doanh nghiệp để sản xuất và cung cấp sản phẩm ra thị trường. Thì để thiết lập một trang bán hàng riêng, người đó chỉ cần thuê dịch vụ tên miền ở một nhà cung cấp bất kì. Đăng ký tài khoản và chờ duyệt trở thành nhà bán hàng. Sau khi được duyệt, nhà sản xuất có thể đăng tải thông tin và cung cấp quyền cài đặt tên miền để liên kết trình bày dữ liệu của nhà sản xuất lên tại tên miền đó. Chi phí khởi tạo và duy trì là miễn phí. Khi doanh nghiệp thực sự cần và đã khai thác được nguồn lợi từ việc quảng bá trang thương mại điện tử. Hệ thống phần mềm có cơ chế mở rộng tài nguyên sử dụng cho nhà sản xuất.

Tiếp đến khi một nhà sản xuất, hoặc một cửa hàng có tài khoản nhà bán hàng trên hệ thống. Họ có nhu cầu đăng tải sản phẩm của mình tác trên trang thông tin của mình để tiếp thị sản phẩm cho khách hàng truy cập trang được đa dạng hơn. Người đó cần tạo một lời mời ký gửi thông tin đến nhà sản xuất sản phẩm đó. Sau khi được duyệt, các sản phẩm của nhà sản xuất đã hiện thì và có thể đặt hàng ở trên trang của đối tác.

Sau khi nhận được đơn đặt hàng, cùng với sản phẩm nằm ngoài doanh mục mà cửa hàng hoặc nhà sản xuất đó cung cấp. Người xử lí đơn có thể chuyển tiếp đơn đó cho đối tác xử lí như một hình thức dropshipping.

Khách hàng truy cập vào trang sẽ thấy thông tin như một trang thương mại điện tử thông thường, các trang thương mại điện tử này dùng chung dữ liệu nên một tài khoản có thể đăng nhập trên nhiều trang. Tạo ra sự đăng nhập liền mạch cải thiện trải nghiệm người dùng.

2.1.2 Yêu cầu tính năng

Các tính năng yêu cầu có thể được liệt kê ra như sau:

- Đăng nhập, đăng xuất, đăng ký, quên mật khẩu.
- Quản lý các thông tin liên quan đến sản phẩm và bài viết.
- Yêu cầu đối tác liên kết thông tin với nhau. Chấp nhận liên kết hoặc xóa.
- Khách hàng có thể xem và đặt sản phẩm ở các trang khác nhau.
- Có trang quản trị chung cho các nhà bán hàng xem đơn đặt hàng trên trang thương mại điện tử của họ.
- Nhà bán hàng có thể gửi thư điện tử cho những người có liên kết bạn bè.
- Hệ thống có thể gửi thư điện tử thông báo đơn hàng.
- Yêu cầu mở tài khoản cho nhà bán hàng nhanh và tự động. Hạn chế sự can thiệp về mặt kỹ thuật của nhà phát triển.
- Yêu cầu đo lường được hoạt động của người dùng hệ thống.
- Sản phẩm có thể có nhiều thuộc tính tùy chọn khi mua hàng.
- Có thể cài đặt hiển thị số lượng tồn kho của sản phẩm trên trang.
- Có thể tự tùy chỉnh hầu hết các nội dung trên trang ngoại trừ các chỉnh sửa về mặt đồ họa, giao diện, bố cục của trang.

2.1.3 Yêu cầu giao diện

- Giao diện cần được thiết kế hiện đại và dễ sử dụng.
- Tương thích ít nhất 80% các loại màn hình.
- Mức độ hoàn thiện tương thích màn hình ít nhất 70%.
- Thao tác nút bấm to rõ và hạn chế bị giật khi tải lại các thành phần của trang.
- Tăng các thành phần phản ứng với hành động của người dùng để tạo cảm giác hệ thống đang xử lý liền mạch không bị chậm chạp.

2.1.4 Yêu cầu hiệu suất

- Tất cả các tương tác của người dùng không được phản hồi chậm quá 3 giây.
- Tất cả phản hồi có thể đợi quá 5 giây cần được phản hồi lại tiến độ hoàn thành. Hoặc tình trạng truy vấn.
- Các thay đổi dữ liệu không được cập nhật chậm hơn 5 phút sau khi sửa đổi.

2.1.5 Yêu cầu thiết kế

- Hệ thống sử dụng các công nghệ còn được duy trì và phát triển.
- Hệ thống có thể chia nhỏ ra thành các đội nhóm khi trở nên quá lớn.
- Hệ thống có thể nâng cấp theo chiều ngang với các lựa chọn có chi phí hợp lí.
- Hệ thống không bị phụ thuộc và có phương pháp phát triển dự phòng.

2.1.6 Yêu cầu phi tính năng

- Chi phí duy trì gói miễn phí cho một trang thương mại ít tương tác nhỏ hơn 10% chi phí gói nâng cao phổ biến nhất.
- Chi phí gói nâng cao không vượt quá 2.500.000 VNĐ/năm.
- Thời gian khởi tạo cài đặt tên miền không quá 4 giờ làm việc (không bao gồm đợi cache dns).
- Mã hóa thông tin mật khẩu.
- Backup dữ liệu hằng tháng.
- Backup tiến trình thực thi ổn định.

2.1.7 Phụ lục

Về chiến lược truyền thông và ứng dụng. Hệ thống cần dành thời gian phát triển các tính năng theo dõi, đóng gói tính năng để thương mại. Chiến lược ứng dụng nhắm đến các nhà sản xuất nhỏ lẻ, hộ kinh doanh hoặc doanh nghiệp khởi nghiệp. Đồng thành nuôi dưỡng giá trị sử dụng sẽ cần một thời gian dài. Nếu một doanh nghiệp không phát triển, thì phần trăm sử dụng hệ thống là không đáng kể. Nên cũng không cần quá quan tâm về tình trạng sử dụng tài nguyên của các đơn vị đó. Đối với các doanh nghiệp trên đà hoặc đã phát triển. Cần

đánh giá đúng mức độ sử dụng và thương mại đúng cách để phát triển đội ngũ lập trình hệ thống.

Ngoài ra, trong quá trình phát triển dự án cũng phát sinh thêm một số tính năng nhỏ mà doanh nghiệp yêu cầu. Nằm ngoài phạm vi đề cập trong báo cáo nhưng có triển khai thực tế trong phần 3. Mô tả đơn giản qua câu chuyện Tí chăm công cho Tèo sau:

Nhân viên Tèo được phỏng vấn vào làm các công việc: Gia công, trực cửa hàng, đóng gói, giao hàng.

Tèo làm buổi sáng 4 tiếng gia công. Buổi chiều trực cửa hàng 3 tiếng. Đóng gói 1 tiếng. Làm xong tèo báo cáo vào bảng rồi đi về.

Tí buổi hôm sau chăm công cho Tèo 4 tiếng gia công được 80.000đ. 3 tiếng trực ca được 45.000đ và 1 tiếng đóng gói được 22.000đ. Tổng cộng 147.000đ

Rồi Tèo lại tiếp tục làm công việc gia công, nhưng hôm nay sản phẩm Tèo làm quen tay, Tèo muốn làm và báo cáo theo sản phẩm. Tèo làm được 10 sản phẩm trong 4 tiếng. Buổi chiều Tèo đi giao hàng. Xong việc Tèo báo cáo nhận lương và đi về.

Rồi ngày hôm sau đáng lẽ hôm nay Tèo được nhận 100.000đ gia công tính theo sản phẩm. Và 4 tiếng đi giao hàng được 120.000đ. Tổng cộng 220.000đ. Nhưng Tí bị ốm và không trả công được cho Tèo. Nên Tèo hôm đó phải nghỉ nhậu.

Tháng sau Tèo quyết định không nhậu nữa và muốn nhận lương một lần vào cuối tháng. Tèo cũng rất quyết tâm kiếm tiền nên quyết định tăng ca. Công việc khi tăng ca giống với công việc bình thường nhưng lương cao hơn. Tèo cũng hay mở điện thoại ra để kiểm tra xem mình đã tích góp được bao nhiêu tiền.

Tí cảm thấy Tèo rất nhiệt tình nên đã quyết định thưởng thêm cho Tèo một số tiền vào một số buổi tăng ca.

Nhưng trước ngày chốt lương 3 ngày. Tèo lại muốn mua một con robot. Nên Tèo quyết định chốt lương sớm và ứng trước một số tiền.

Tí thấy yêu cầu của Tèo là hợp lí và sẽ chuyển tiền trong vòng 3 ngày cho Tèo.

2.1.8 Vai trò

Người bán hàng

Người bán hàng là người có quyền công bố dữ liệu của họ cho mọi người truy cập thông qua một tên miền cụ thể.

Người bán hàng có thể mời người bán hàng khác, gộp dữ liệu sản phẩm của họ vào bày bán ở trang của mình. Hoặc cũng có thể tham gia đóng góp sản phẩm của mình vào các sàn bán hàng của những người bán hàng khác.

Người mua hàng

Người mua hàng là người thực hiện các hoạt động xem thông tin, tạo giỏ hàng, thêm sản phẩm vào giỏ hàng và đặt mua. Người mua hàng không cần thiết phải đăng nhập vào hệ thống. Chỉ cần để lại thông tin địa chỉ để giao hàng là được. Vì hiện tại, hai hình thức thanh toán chuyển khoản và thanh toán COD rất an toàn cho người bán. Nếu có rủi ro thì chi phí vận chuyển là không quá lớn.

Người mua hàng nếu chủ động lưu trữ thông tin đơn hàng hoặc các hoạt động để cá nhân hóa dữ liệu thì có thể đăng nhập trước khi thực hiện các thành động liên quan.

2.1.9 Hoạt động

Giống như các hệ thống khác, hoạt động được chia ra thành tạo, xem, sửa, xóa. Các hoạt động này được xác định là xảy ra ở tên miền nào, dữ liệu đối tượng gì, do ai có vai trò gì thực hiện.

Với cách xác định hoạt động như trên, có gần với khái niệm *phương thức* của đối tượng trong quan niệm thiết kế hướng đối tượng.

2.1.10 Phân quyền

Phân quyền không chỉ giúp cho hệ thống trở nên bảo mật hơn. Mà còn giúp cho đội ngũ phát triển làm việc chặt chẽ hơn với dữ liệu. Một dạng dữ liệu được phân quyền với yêu cầu đặc tả của khách hàng không thể truyền đạt hết cho tất cả những thành viên trong nhóm được. Điều đó gây ra các lỗi hổng đằng sau *Controller* trong mô hình MVC. Hệ thống xử dụng các nút xử lý mang theo thông tin chung đại diện cho người truy cập. Trong đó có thể phân tách định nghĩa được hoạt động và phân quyền được chặt chẽ hơn. Dưới đây là một số định nghĩa về phân quyền cần lưu ý:

Quyền đại diện

Khi thực hiện truy vấn dữ liệu, cần xác định nguồn gốc của truy vấn đến từ HTTP request nào. Quyền đại diện cho truy vấn được cấp cho tài khoản nhà bán hàng có tên miền khớp với tên miền chứa trong HTTP request đó. Tức là có thể lấy được ID của tài khoản đại diện cho HTTP request hiện tại.

Chỉ có tài khoản người bán hàng mới có thể đại diện cho một truy vấn. Khi người bán hàng truy vấn từ một trang không phải thuộc sở hữu của họ. Thì hệ thống thay vì căn cứ vào tên miền ở trong request hiện tại như cách cũ, nó sẽ trả về luôn ID của người bán hàng hiện tại. Tức là ghi đè quyền đại diện nếu người thực hiện truy vấn là một nhà bán hàng.

Quyền đại diện thường được sử dụng để xem dữ liệu như một nhà bán hàng tại một tên miền cụ thể. Nghĩa là khi người dùng truy vấn dữ liệu sản phẩm tại một domain nào đó. Thì hệ thống sẽ hỏi cơ sở dữ liệu ai đại diện cho truy vấn này để trả về dữ liệu tương ứng cho truy vấn đó.

Điều này cho phép cùng một hệ thống máy chủ giao diện khi cài đặt với các tên miền khác nhau sẽ trả về các kết quả khác nhau.

Phân quyền này dựa trên chứng thực headers của HTTP request. Tức là nếu ở ứng dụng mobile hoặc một nền tảng khác không chứng thực được headers. Người dùng không thể giả mạo headers để truy vấn thông tin của nhà bán hàng.

Nội dung và phương pháp chứng thực headers HTTP request nằm ngoài phạm vi đề cập của dự án.

Quyền hành động

Quyền sở hữu. Đối với quyền sở hữu. Ai tạo ra dữ liệu thì người đó mới có quyền xóa dữ liệu đó. Cho nên tất cả các bảng đều phải có ghi thông tin người tạo. Bởi vì trong quá trình phát triển dự án theo yêu cầu thực tế, mỗi liên hệ giữa các bảng có thể thay đổi nhưng quyền sở hữu giữa dữ liệu đó và người sở hữu không thay đổi. Nếu tạo ra dữ liệu mà không đăng nhập, không xác định được người sở hữu, thì dữ liệu sẽ thuộc về người đại diện cho truy cập (Xem chi tiết người đại diện của truy cập mục 2.1.10).

Chuyển quyền sở hữu. Khi tạo đơn đặt hàng, tạo nội dung, dữ liệu sản phẩm cho người khác. Phải chuyển quyền sở hữu sau đó. Sẽ có nghiệp vụ chạy ngầm để xét duyệt và ghi đè quyền sở hữu cho người thừa kế. Ghi thông tin người tạo vào dữ liệu vào một trường khác nếu cần. Nhưng bản chất dữ liệu đó đã chuyển quyền sở hữu rồi.

Nghĩa là, coi như người thừa kế (*người được chuyển*) tạo ra dữ liệu đó. sau khi chuyển người sở hữu cũ không có quyền xóa nữa.

Quyền xem. đa dạng hơn tùy trường hợp sử dụng.

Quyền xem được thiết đặt theo thứ tự: mặc định, bảng dữ liệu, trường dữ liệu. Nếu định nghĩa quyền xem cho một bảng dữ liệu cụ thể, thì hệ thống sẽ thực hiện định nghĩa đó thay cho quyền mặc định. Tương tự, nếu một trường có định nghĩa quyền xem cho riêng nó, thì khi truy vấn bảng dữ liệu, định nghĩa quyền chỉ áp dụng cho của bảng. Sau đó nếu truy cập hợp lệ vào trường đó thì mới trả về được dữ liệu của trường.

Ví dụ cho trường hợp phân quyền cho trường dữ liệu là khi truy vấn đến bảng người dùng, người xem có thể xem tên, hình ảnh đại diện nhưng không thể xem số điện thoại và mật khẩu đã mã hóa.

Quyền xem chia ra làm hai loại là phân quyền trực tiếp và phân quyền gián tiếp.

- Đối với phân quyền trực tiếp thì người tạo ra, người sở hữu dữ liệu toàn quyền xem với dữ liệu đó.
- Quyền xem gián tiếp là quyền cấp cho người xem dữ liệu mà không cần đăng nhập, hoặc đăng nhập nhưng không phải là nhà bán hàng.

Người xem gián tiếp được định nghĩa là khi truy cập dữ liệu dưới một tên miền, thì thông qua tên miền để lấy được quyền truy cập đến các dữ liệu được chia sẻ của: người sở hữu tên miền và người chia sẻ dữ liệu cho tên miền. Hoặc có thể hiểu là người mua hàng đã thông qua tên miền để truy cập đến dữ liệu của nhà bán hàng. Hoặc ngược lại căn cứ theo tên miền tìm nhà bán hàng đại diện để phục vụ dữ liệu cho người dùng.

Quyền tạo Khi tạo dữ liệu, đối với một số bảng dữ liệu bắt buộc đăng nhập thì người dùng phải đăng nhập, hoặc là nhà bán hàng thì mới có thể tạo.

Cũng có trường hợp các loại dữ liệu không cần đăng nhập vẫn tạo được để tạo ra sự thuận tiện cho người mua hàng. Ví dụ thêm một sản phẩm vào giỏ hàng. Thực tế khi mua hàng tại các siêu thị, người mua hàng có thể yêu cầu mở thẻ thành viên hoặc là không.

Những dữ liệu không đăng nhập có thể xem bởi tất cả mọi người, nhưng những sản phẩm có thông tin người tạo cần được xem xét cho phép truy cập bởi một số người liên quan chẳng hạn như đơn hàng được tạo trên cửa hàng. Người chủ cửa hàng cần quyền xem đơn hàng đó.

Dưới đây là các mô tả cụ thể hơn về phân quyền:

Người dùng Người dùng có thể nhìn thấy thông tin cá nhân cơ bản của chính họ và của những người đã chấp nhận bạn bè.

Lời mời kết bạn Người tạo ra lời mời và người được mời có thể nhìn thấy lời mời đó.

Hợp đồng Hợp đồng tương tự như lời mời kết bạn chỉ người tạo ra và đối tác của người đó được xem dữ liệu.

Các tương tác Các tương tác được xem công khai cho tất cả mọi người nhưng chỉ người tạo ra mới được chỉnh sửa.

Thông báo Thông báo từ hệ thống đến người dùng chỉ được xem bởi chính người đó. Trong trường hợp gửi thông báo cho nhau thì người gửi và người nhận được xem thông báo đó.

Nhóm Nhóm được xem bởi người tham gia nhóm và người đã chấp nhận lời mời tham gia nhóm. Lời mời tham gia nhóm tương tự với lời mời kết bạn.

Thống kê truy cập Thống kê theo dõi truy cập của một cá nhân chỉ được truy cập quản lí bởi chính cá nhân đó.

Các bảng công việc Khi giữa các tài khoản người dùng và tài khoản nhà bán hàng có kí hợp đồng lao động, người dùng có thể tạo bảng để chấm công khi làm việc cho một nhà bán hàng nào đó. Nên nhà bán hàng và người chấm công có thể xem. Yêu cầu chốt công tạo ra bởi người chấm công nhưng không ai có thể xóa. Nhà bán hàng có thể cập nhật hoặc ghi chú vào bảng chốt công.

Thông tin công khai Các thông tin công khai là những thông tin được tạo ra bởi nhà bán hàng và công khai truy cập thông qua một tên miền cụ thể. Đối với các truy cập không có thông tin tên miền. Người dùng đó không có quyền truy cập vào dữ liệu công khai. Ngoại trừ trường hợp người dùng đó đăng nhập với tài khoản nhà bán hàng (xem thêm **quyền xem** tại mục 2.1.10).

2.2 Phân tích hoạt động

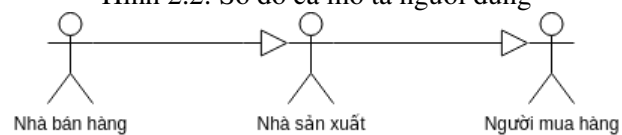
Phân tích hoạt động bao gồm các sơ đồ mô tả hành vi, kiến trúc dựa theo ngôn ngữ đặc tả. Các sơ đồ sử dụng ngôn ngữ đặc tả UML 2.5. Các mô tả thiết kế cụ thể hơn dành gần với công cụ triển khai nằm ở phần mô tả chi tiết. Công cụ sử dụng để vẽ sơ đồ là diagrams.net.

2.2.1 Sơ đồ ca sử dụng

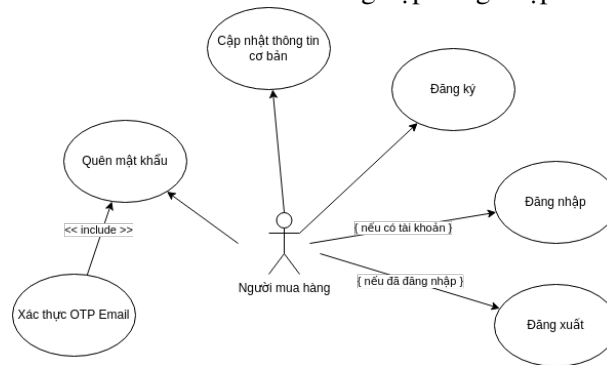
Hình 2.1: Sơ đồ ca sử dụng tổng quát



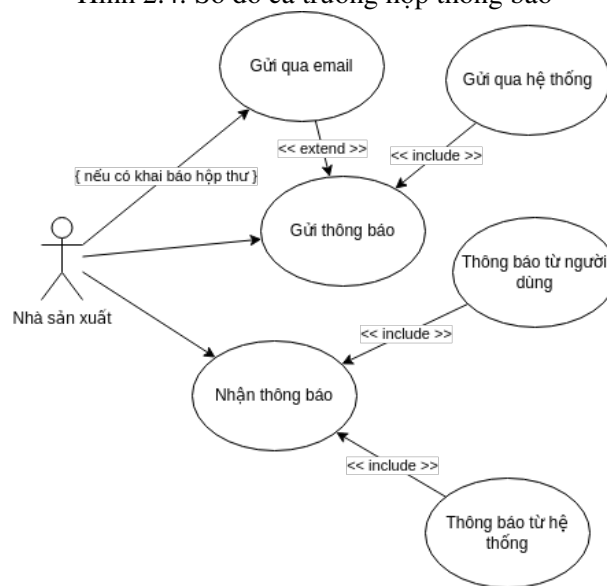
Hình 2.2: Sơ đồ ca mô tả người dùng



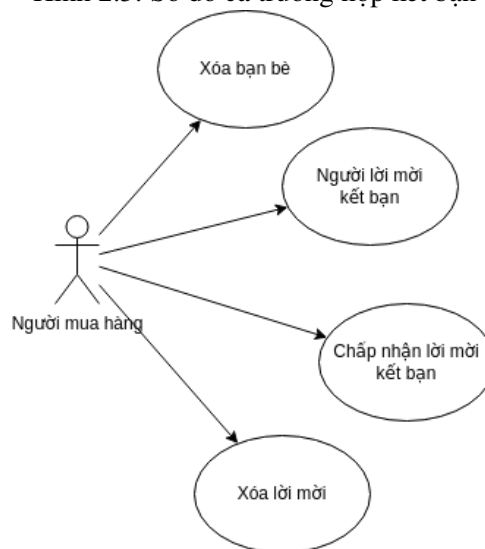
Hình 2.3: Sơ đồ ca trường hợp đăng nhập



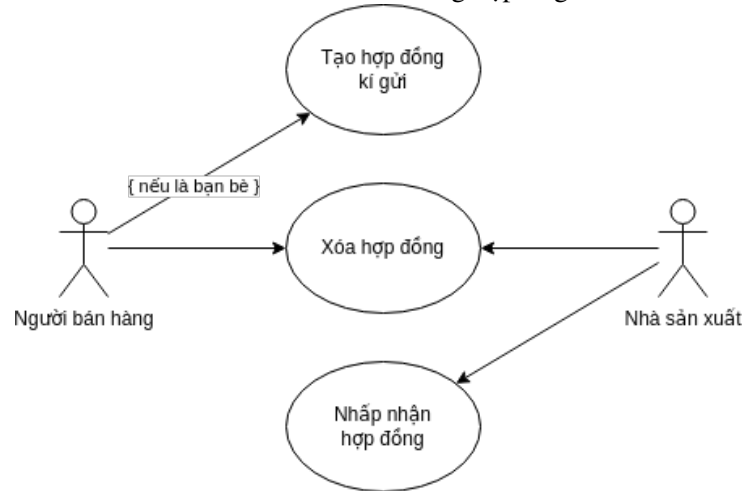
Hình 2.4: Sơ đồ ca trường hợp thông báo



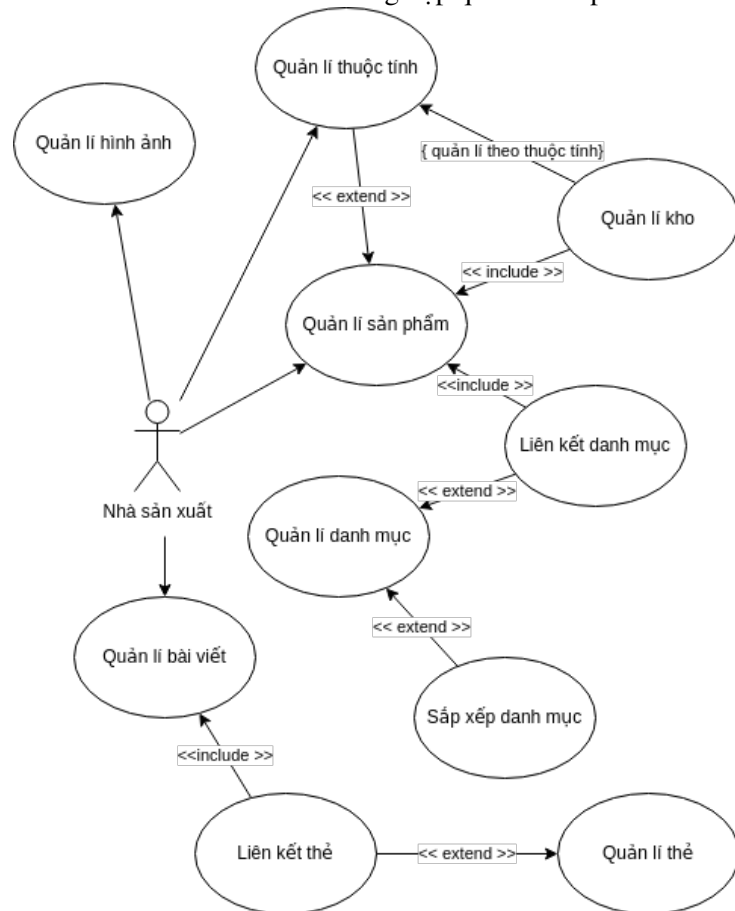
Hình 2.5: Sơ đồ ca trường hợp kết bạn



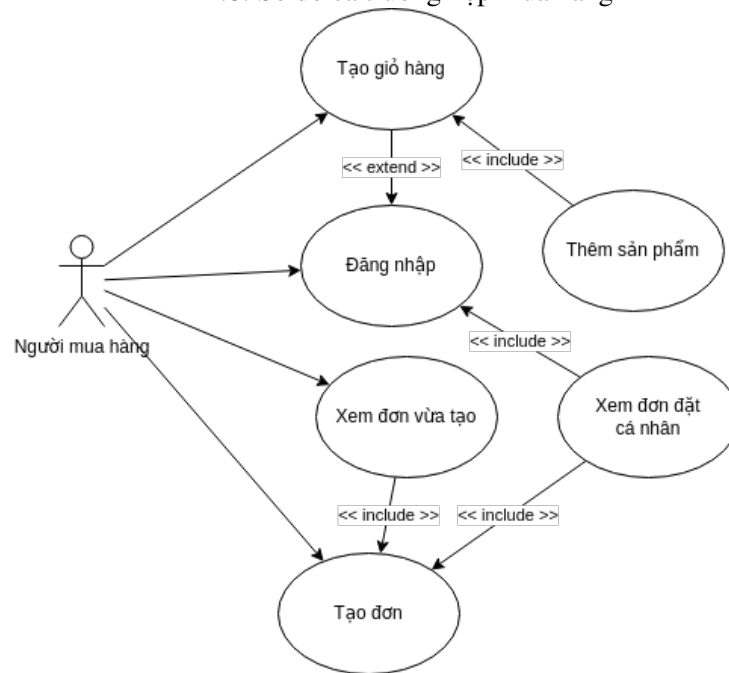
Hình 2.6: Sơ đồ ca trường hợp kí gửi



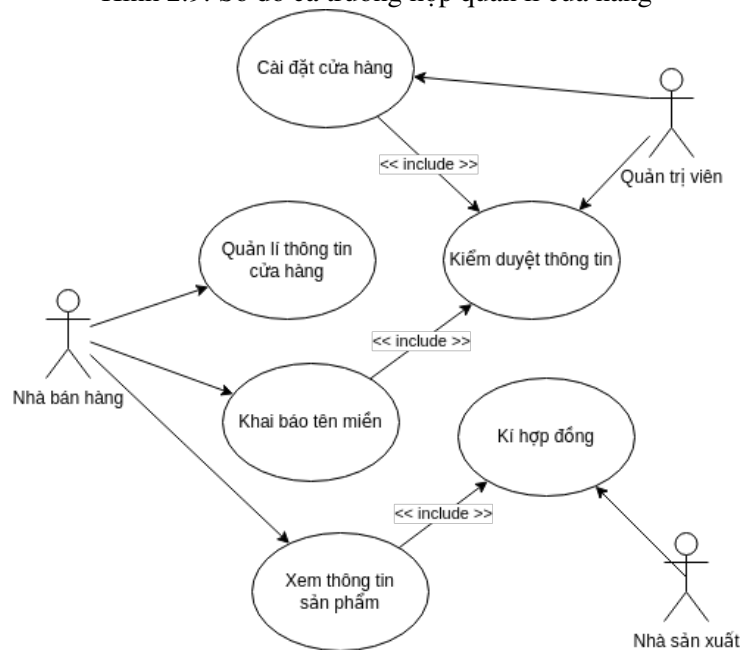
Hình 2.7: Sơ đồ ca trường hợp quản lí sản phẩm



Hình 2.8: Sơ đồ ca trường hợp mua hàng

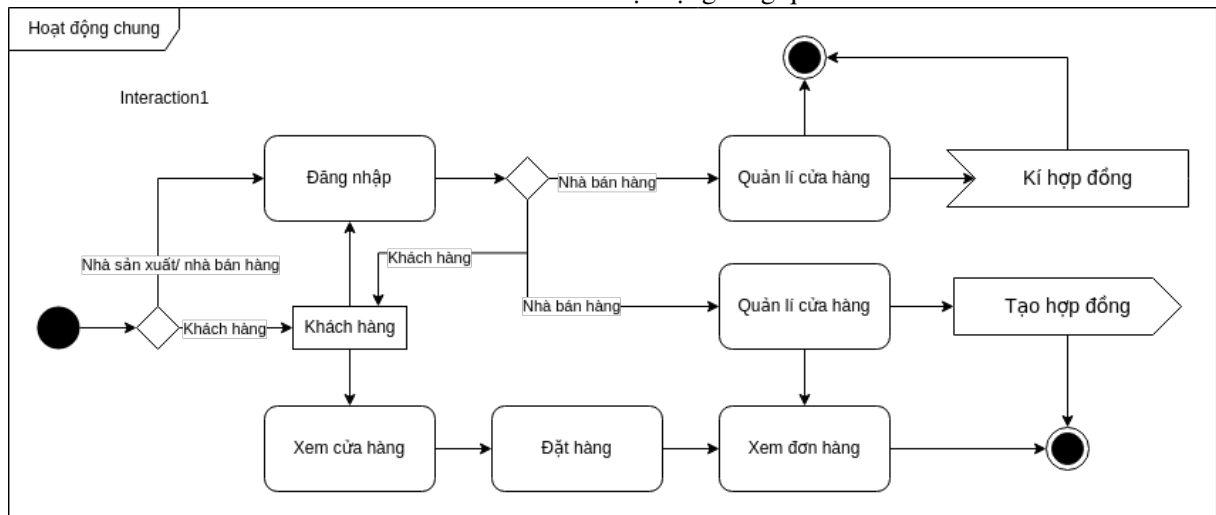


Hình 2.9: Sơ đồ ca trường hợp quản lí cửa hàng

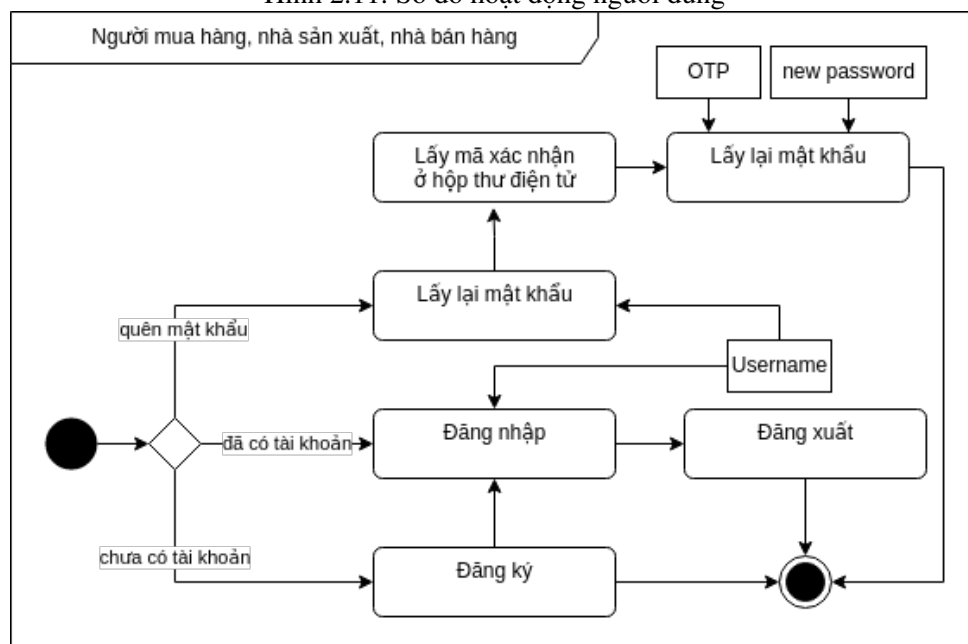


2.2.2 Sơ đồ hoạt động

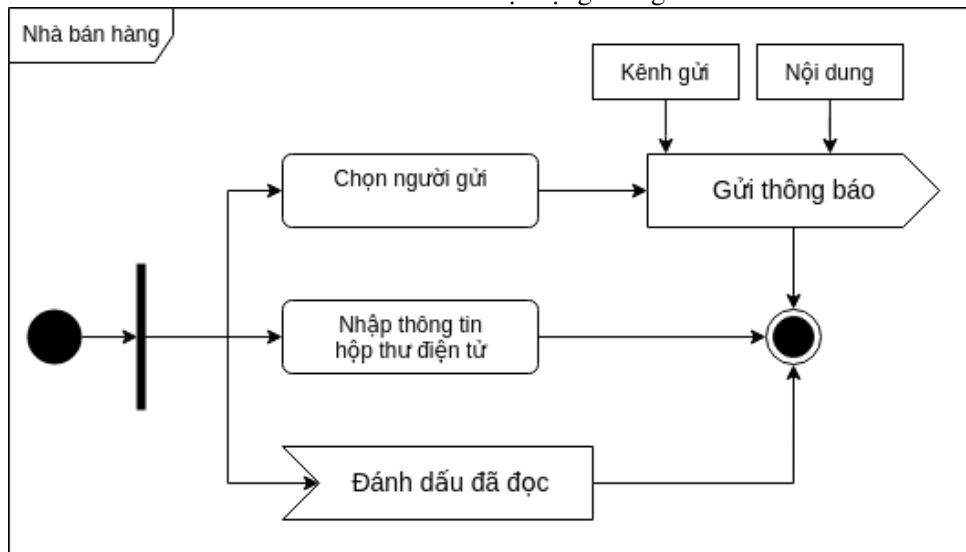
Hình 2.10: Sơ đồ hoạt động tổng quát



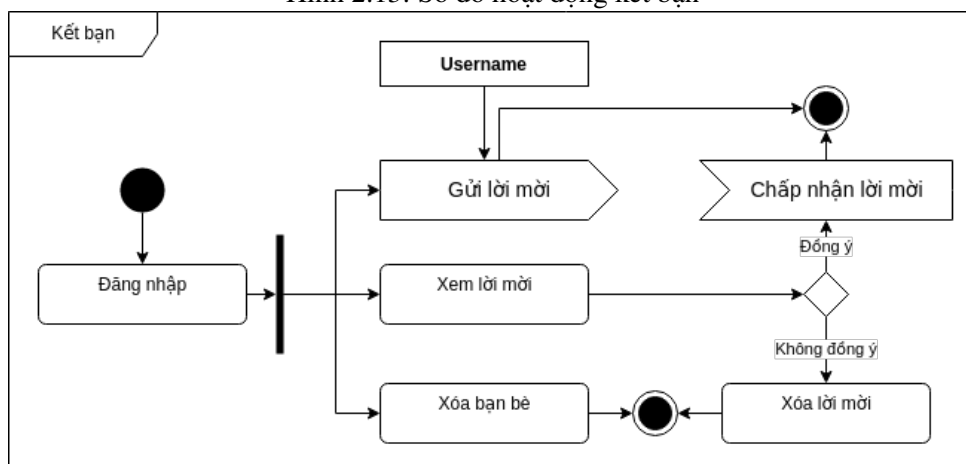
Hình 2.11: Sơ đồ hoạt động người dùng



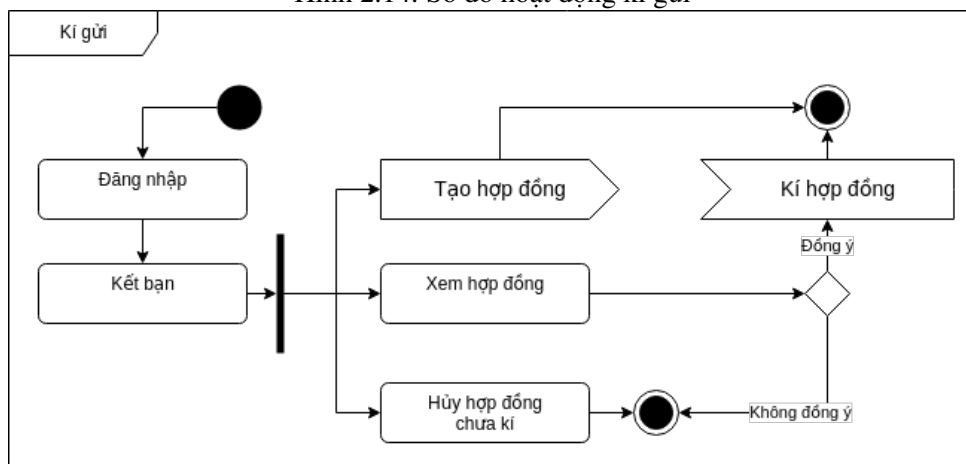
Hình 2.12: Sơ đồ hoạt động thông báo



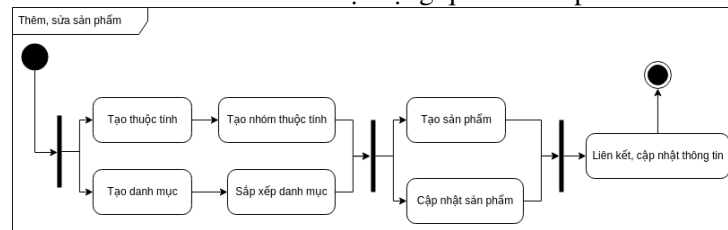
Hình 2.13: Sơ đồ hoạt động kết bạn



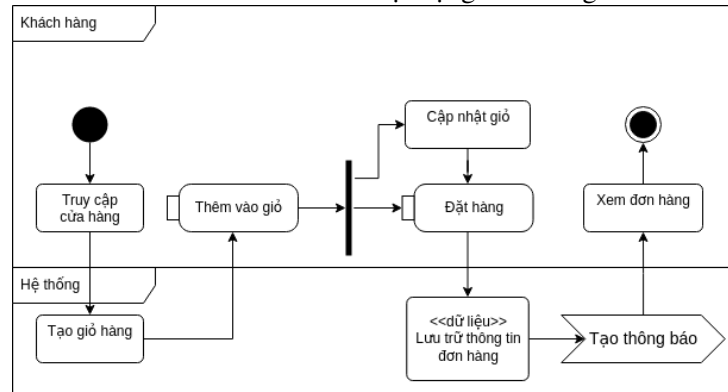
Hình 2.14: Sơ đồ hoạt động kí gửi



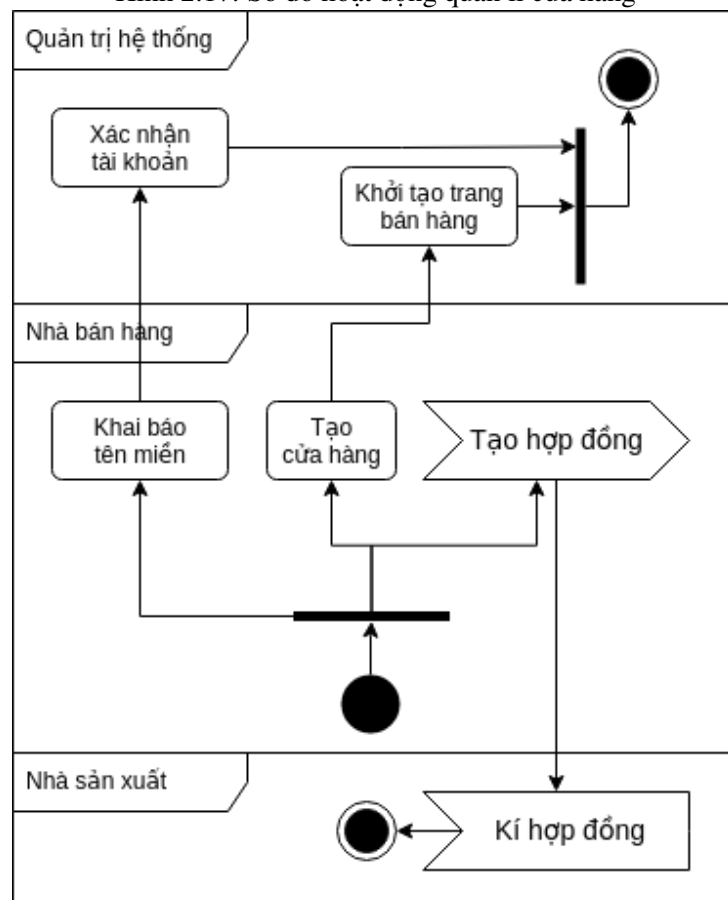
Hình 2.15: Sơ đồ hoạt động quản lí sản phẩm



Hình 2.16: Sơ đồ hoạt động mua hàng



Hình 2.17: Sơ đồ hoạt động quản lí cửa hàng



2.2.3 Sơ đồ trạng thái

2.2.4 Sơ đồ tuần tự

2.3 Thiết kế cấu trúc

2.3.1 Sơ đồ lớp

2.3.2 Sơ đồ đóng gói

2.3.3 Sơ đồ khối thành phần

2.3.4 Sơ đồ phát hành

Chương 3

Triển khai và đánh giá

3.1 Triển khai

3.1.1 Nội quy, hướng dẫn, yêu cầu

Kiến thức cơ bản

Thư viện bậc cao

3.1.2 Kiến thức khác

Một trong những vấn đề khi làm việc nhóm là sự hiểu nhau giữa các thành viên trong nhóm. Mình có thể khắc phục bằng cách trao đổi nhiều hơn. Xác nhận nhiều hơn. Nhưng khi nhóm trở nên lớn. Thành viên trong nhóm cũng có sự thay đổi liên tục. Hoặc khung giờ làm việc, lịch trình ngày làm việc của mọi người khác nhau. Thì việc này dần trở nên khó khăn hơn. Một chức năng phát triển xong cách khoảng thời gian dài sau đó nâng cấp. Nếu cùng một người thì đây là giao tiếp với chính bản thân mình.

Vậy để giải quyết vấn đề này, mọi người cần viết viết và viết. Viết đúng là tốn thời gian và làm chậm năng suất của mình. Nhưng đối với các vấn đề trên. Viết là cách giải quyết.

Viết bất cứ cái gì có thể viết. Cách thức xác định cái gì cần viết bao gồm:

- Nếu như mình không làm công việc này nữa. Thì chỉ lại cho ai, chỉ những gì. Suy ra viết những gì học được.
- Nếu như ai đó cần giúp mình, thì mình cần chỉ họ đọc gì, hiểu gì để làm. Suy ra viết những gì hiểu được trong từng trường hợp.
- Nếu như ai đó thay đổi công việc mình đã làm, họ cần lưu ý những gì. Suy ra viết những dự định hoặc rủi ro lường trước với công việc mình đang làm.

Báo cáo kiểm soát công việc

Dành cho thay đổi, thêm mới tính năng

Thay đổi hoặc thêm mới gì
Mô tả hướng thay đổi, lý do thay đổi.

Dành cho sửa lỗi

Nguyên nhân

Nguyên nhân gây lỗi.

Điều tra

Các thay đổi trong quá khứ gây ra lỗi hoặc điều tra các chức năng liên quan đến thay đổi.

Các hiểu nhầm hoặc yêu cầu thay đổi gây ra lỗi.

Link pull request gây ra lỗi.

Hướng thay đổi

Hướng thực hiện thay đổi

Phạm vi can thiệp

Màn hình

Nhóm components

Schema

Kết quả

Chụp ảnh màn hình, kết quả API

Lưu ý

3.2 Mô tả chi tiết

Mô tả chi tiết thuộc về tài liệu phát triển chứ, là kết quả khi phát triển theo phân tích thiết kế. Mô tả chi tiết giúp đội ngũ phát triển dễ dàng ghi lại và trao đổi các thông tin của dự án.

3.2.1 Dữ liệu

Tên	Kiểu	Mô tả
to	Virtual
isAccepted	Checkbox
key	Text

Bảng 3.1: Mô tả bảng ContractConsignment.

Tên	Kiểu	Mô tả
content	Markdown
status	Select
price	Virtual

Bảng 3.2: Mô tả bảng ContractLabor.

Tên	Kiểu	Mô tả
content	Text

Bảng 3.3: Mô tả bảng InteractionComment.

Tên	Kiểu	Mô tả
emoji	Select

Bảng 3.4: Mô tả bảng InteractionReaction.

Tên	Kiểu	Mô tả
status	Checkbox
item	MongoId

Bảng 3.5: Mô tả bảng Interaction.

Tên	Kiểu	Mô tả
username	Text
password	Text
host	Text
port	Integer
secure	Checkbox
name	Text

Bảng 3.6: Mô tả bảng NotificationMailer.

Tên	Kiểu	Mô tả
chanel	Select
subject	Text
text	Text
seen	Checkbox

Bảng 3.7: Mô tả bảng Notification.

Tên	Kiểu	Mô tả
username	Text
to	Virtual
isAccepted	Checkbox
key	Text
consignment	Virtual

Bảng 3.8: Mô tả bảng Relationship.

Tên	Kiểu	Mô tả
isAccepted	Checkbox
key	Text

Bảng 3.9: Mô tả bảng TeamInvitation.

Tên	Kiểu	Mô tả
name	Text
description	Text

Bảng 3.10: Mô tả bảng Team.

Tên	Kiểu	Mô tả
username	Text
password	Password
phone	Text
email	Text
name	Text
fullname	Text
avatar	File
about	Text
domain	Text
isAdmin	Checkbox
isSeller	Checkbox
isActive	Checkbox
forgotAt	DateTime

Bảng 3.11: Mô tả bảng User.

Tên	Kiểu	Mô tả
listKey	Text
count	Integer
of	MongoId

Bảng 3.12: Mô tả bảng View.

Tên	Kiểu	Mô tả
price	Currency

Bảng 3.13: Mô tả bảng WorkPaid.

Tên	Kiểu	Mô tả
title	Text
price	Currency
unit	Select

Bảng 3.14: Mô tả bảng WorkType.

Tên	Kiểu	Mô tả
amount	Integer
content	Text
confirmAt	CalendarDay

Bảng 3.15: Mô tả bảng Work.

Tên	Kiểu	Mô tả
name	Text
slogan	Text
image	File
description	Text
url	Text
type	Select
size	Virtual

Bảng 3.16: Mô tả bảng Banner.

Tên	Kiểu	Mô tả
phone	Text
name	Text
address	Text
email	Text
note	Text
message	Text
isDefault	Checkbox

Bảng 3.17: Mô tả bảng Contact.

Tên	Kiểu	Mô tả
store	Text
logo	File
slogan	Text
address	Text
phone	Text
email	Text
intro	Editor
contact	Editor
twitter	Text
instagram	Text
pinterest	Text
youtube	Text
googlePlus	Text
googleMap	Text
zalo	Text
greeting	Text
pageId	Text
pixelId	Text
gtag	Text
shipMoneySupport	Integer
ship	Editor
transfer	Editor
color	Color
colorMode	Select
ordering	Checkbox
moit	Text
mst	Text
language	Select
translations	Translate

Bảng 3.18: Mô tả bảng Page.

Tên	Kiểu	Mô tả
value	Text
file	File

Bảng 3.19: Mô tả bảng ProductAttributeValue.

Tên	Kiểu	Mô tả
label	Text
name	Text
language	Select
translations	Translate

Bảng 3.20: Mô tả bảng ProductAttribute.

Tên	Kiểu	Mô tả
name	Text
url	Slug
language	Select
translations	Translate

Bảng 3.21: Mô tả bảng ProductBrand.

Tên	Kiểu	Mô tả
sale	Integer
price	Integer
percent	Virtual
isInCart	Virtual
quantity	Integer

Bảng 3.22: Mô tả bảng ProductCartItem.

Tên	Kiểu	Mô tả
-----	------	-------

Bảng 3.23: Mô tả bảng ProductCart.

Tên	Kiểu	Mô tả
name	Text
description	Editor
file	File
prioritize	Integer
url	Slug
root	Checkbox
language	Select
translations	Translate

Bảng 3.24: Mô tả bảng ProductCategory.

Tên	Kiểu	Mô tả
code	Text
type	Select
value	Integer
name	Text
description	Text
condition	Integer
image	File
url	Slug
language	Select
translations	Translate

Bảng 3.25: Mô tả bảng ProductDiscount.

Tên	Kiểu	Mô tả
name	Text
url	Slug
language	Select
translations	Translate

Bảng 3.26: Mô tả bảng ProductHashtag.

Tên	Kiểu	Mô tả
value	Text
color	Select

Bảng 3.27: Mô tả bảng ProductOrderStatus.

Tên	Kiểu	Mô tả
code	Text
isExport	Checkbox
payment	Select
saving	Integer
total	Integer
notification	MongoId

Bảng 3.28: Mô tả bảng ProductOrder.

Tên	Kiểu	Mô tả
quantity	Integer
image	File

Bảng 3.29: Mô tả bảng ProductStock.

Tên	Kiểu	Mô tả
image	File
images	Images
name	Text
price	Currency
sale	Currency
percent	Virtual
status	Select
description	Editor
detail	File
guide	Editor
isDraft	Checkbox
isOutOfStock	Checkbox
sku	Text
gs1	Text
url	Slug
sold	Virtual
language	Select
translations	Translate

Bảng 3.30: Mô tả bảng Product.

Tên	Kiểu	Mô tả
item	MongoId
listKey	Text
fieldName	Text
lang	Text
content	Text

Bảng 3.31: Mô tả bảng Translate.

Tên	Kiểu	Mô tả
file	File
address	Text

Bảng 3.32: Mô tả bảng UploadFile.

Tên	Kiểu	Mô tả
file	File
alt	Text

Bảng 3.33: Mô tả bảng UploadImage.

Tên	Kiểu	Mô tả
title	Text
body	Markdown
prioritize	Integer

Bảng 3.34: Mô tả bảng FAQ.

Tên	Kiểu	Mô tả
name	Text
image	File
description	Markdown
content	Markdown

Bảng 3.35: Mô tả bảng Feature.

Tên	Kiểu	Mô tả
name	Text
image	File
root	Checkbox
description	Markdown
prioritize	Integer
color	Color
url	Slug

Bảng 3.36: Mô tả bảng PostHashtag.

Tên	Kiểu	Mô tả
title	Text
thumbnail	File
content	Markdown
prioritize	Integer
embed	Text
description	Text
keywords	Text
url	Slug
body	Text

Bảng 3.37: Mô tả bảng Post.

Tên	Kiểu	Mô tả
name	Text
image	File
description	Text
content	Markdown

Bảng 3.38: Mô tả bảng Service.

Tên	Kiểu	Mô tả
name	Text
profile	Text
description	Text
image	File

Bảng 3.39: Mô tả bảng Testimonial.

3.2.2 API.

Lấy dữ liệu

allUsers Lấy tất cả bản ghi trong danh sách *User*. Truy vấn *allUsers* cho phép tìm kiếm, giới hạn và lọc các kết quả. Chi tiết xem bên dưới:

```

1  query {
2    allUsers {
3      id
4      # ...
5      # orthers field
6    }
7  }
```

_allUsersMeta Tương tự như *allUsers*. *_allUsersMeta* trả về chỉ tổng số bản ghi hợp lệ vào *count* mà không phải là các dữ liệu của từng bản ghi cụ thể.

```

1  query {
2    _allUsersMeta {
3      count
4    }
5  }
```

User Truy cập chính xác đến bản ghi trong danh sách *User*. Truy vấn này bắt buộc có *\$id*

```

1  query {
2    User(where: { id: $id }) {
3      name
4      # ...
5      # orthers field
6    }
7  }
```

_UsersMeta Trả về thông tin của chính danh sách *User*, chẳng hạn như trạng thái phân quyền.

Thêm, sửa, xóa

Có 6 phương thức

- *createUser*
- *createUsers*
- *updateUser*
- *updateUsers*
- *deleteUser*
- *deleteUsers*

createUser Thêm một *User* vào danh sách. Yêu cầu có tham số *data*

```
1  mutation {
2    createUser(data: { name: "Huy" }) {
3      id
4      # ...
5      # orthers field
6    }
7  }
```

createUsers Tạo nhiều *Users*. Đầu vào tương tự như *createUser* nhưng ở đây có dạng là *[data, data,...]*

```
1  mutation {
2    createUsers(data: [{ data: { name: "Huy" } },
3      { data: { name: "Ngoc" } }]) {
4      id
5      name
6      # ...
7      # orthers field
8    }
9  }
```

updateUser Cập nhật *User* thông qua ID.

```
1  mutation {
2    updateUser(id: ID, data: { name: "Tran" }) {
3      id
4      # ...
5      # orthers field
6    }
7  }
```

```
7 }
```

updateUsers Cập nhật nhiều *Users* thông qua ID. Tham số truyền vào tương tự như *createUser*. Cần được bao trong bảng và truyền vào biến *data* như ví dụ

```
1 mutation {
2   updateUser(data: [{ id: ID, data: { name: "Tran
3     " } }]) {
4     id
5     # ...
6     # orthers field
7   }
}
```

deleteUser Xóa một *User* thông qua ID

```
1 mutation {
2   deleteUser(id: ID) {
3     id
4     # ...
5     # orthers field
6   }
7 }
```

deleteUsers Xóa nhiều *User* thông qua danh sách ID

```
1 mutation {
2   deleteUsers(ids: [ID]) {
3     id
4     # ...
5     # orthers field
6   }
7 }
```

Lọc, giới hạn, sắp xếp

Bên trong tham số của truy vấn chúng ta có thể thêm:

- where
- search

- skip
- first
- sortBy

where Giới hạn số kết quả trả về đúng với điều kiện truyền vào.

```
1 query {  
2   allUsers (where: { name_starts_with_i: "A" }) {  
3     id  
4     # ...  
5     # orthers field  
6   }  
7 }
```

Lọc trường là một Relationship where

- relatedList_every: whereInput
- relatedList_some: whereInput
- relatedList_none: whereInput
- relatedList_is_null: Boolean

Lọc trường là một String where

- Field: String
- Field_not: String
- Field_contains: String
- Field_not_contains: String
- Field_starts_with: String
- Field_not_starts_with: String
- Field_ends_with: String
- Field_not_ends_with: String
- Field_i: String
- Field_not_i: String

- Field_contains_i: String
- Field_not_contains_i: String
- Field_starts_with_i: String
- Field_not_starts_with_i: String
- Field_ends_with_i: String
- Field_not_ends_with_i: String
- Field_in: [String]
- Field_not_in: [String]

Lọc trường là một ID *where*

- Field: ID
- Field_not: ID
- Field_in: [ID!]
- Field_not_in: [ID!]

Lọc trường là một Integer *where*

- Field: Int
- Field_not: Int
- Field_lt: Int
- Field_lte: Int
- Field_gt: Int
- Field_gte: Int
- Field_in: [Int]
- Field_not_in: [Int]

Toán tử Có hai toán tử

- AND: [whereInput]
- OR: [whereInput]

```
1  query {
2    allUsers (
3      where: { OR: [{ name_starts_with_i: "A" },
4        { email_starts_with_i: "A" }] }
5    ) {
6      id
7      # ...
8      # orthers field
9    }
  }
```

search Sẽ tìm kiếm danh sách theo trường name

```
1  query {
2    allUsers (search: "Huy") {
3      id
4      # ...
5      # orthers field
6    }
7  }
```

sortBy Sắp xếp kết quả

Kiểu được cung cấp (lấy ví dụ cho bảng *User*)

```
1  enum SortUsersBy {
2    id_ASC
3    id_DESC
4    name_ASC
5    name_DESC
6    email_ASC
7    email_DESC
8  }
```

sortBy cho phép một hoặc nhiều giá trị tham gia vào điều kiện sắp xếp.

```
1 query {
2   allUsers(sortBy: name_DESC) {
3     id
4     # ...
5     # orthers field
6   }
7 }
```

```
1 query {
2   allUsers(sortBy: [name_DESC, email_ASC]) {
3     id
4     # ...
5     # orthers field
6   }
7 }
```

first Giới hạn số kết quả trả về trong truy vấn. Sau khi áp dụng các tham số sau: skip, sortBy, where and search.

```
1 query {
2   allUsers(first: 10) {
3     id
4     # ...
5     # orthers field
6   }
7 }
```

skip Bỏ qua các số bản ghi đầu tìm thấy theo chỉ định. Nó được áp dụng trước khi first, nhưng sau sortBy, where and search.

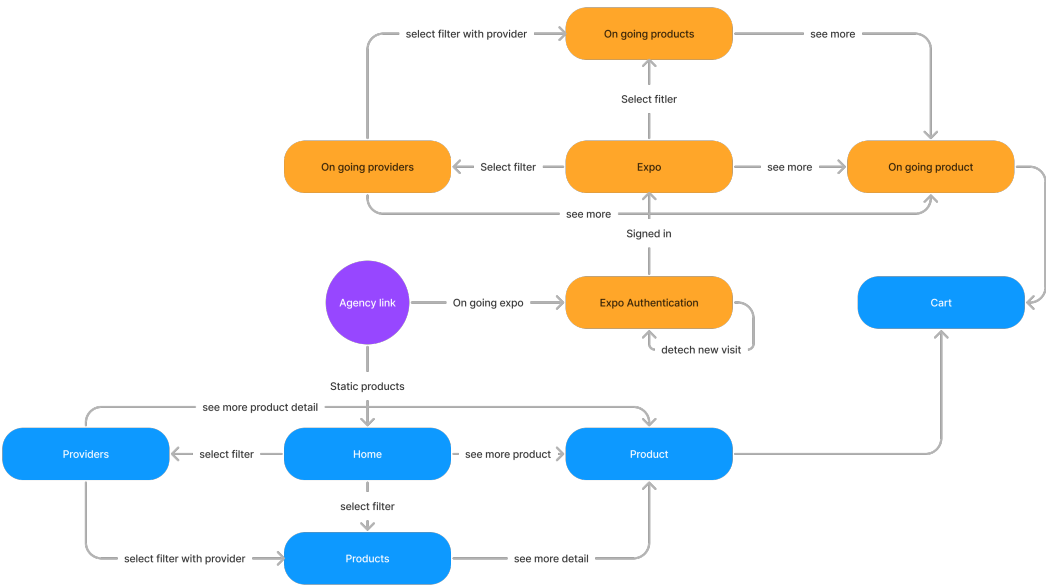
Nếu số lượng *skip* lớn hơn số lượng kết quả hợp lệ, một mảng rỗng sẽ được trả về.

```
1 query {
2   allUsers(skip: 10) {
3     id
4     # ...
5     # orthers field
6   }
7 }
```


3.2.3 Quản lí mã nguồn

- occopee
- gateway
- accounts
- docs
- sellers
- Cài đặt
- Khởi chạy

Hình 3.1: State diagram [Monne]



Backend

3.3 Kết quả đạt được

fashion.occopee.com

3.3.1 Dự án đã phát hành thực tế

Trạng thái ổn định

shopwheydanang.com mayberoom.vn b21shoemaker.store hypesneaker.vn yen-saodatquang.vn

Chưa hoàn thiện, đang phát triển thêm

japan-apple.jp madeuniform.vn khoailangtradoa.com ocopee.com

3.3.2 Dự án đang thử nghiệm

fashion.ocadopee.com

3.4 Những vấn đề hạn chế

3.5 Hướng phát triển

Kết luận

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Tài liệu tham khảo

- [Amaa] Amazon. Api restful là gì?
- [Amab] Amazon. Nosql là gì?
- [Fak] Kirill Fakhroutdinov. The unified modeling language.
- [Han22] TS. Le Thi My Hanh. *Giáo trình môn lập trình hướng đối tượng*. 09/2022, tham khảo ngày 30/06/2022.
- [Monne] MongoDB. Why use mongodb? what we offer and when to use it, 2007, Accessed June 20, 2022) for mongo's define.
- [moz22] mozilla. Trình duyệt web là gì?, truy cập ngày 30/06/2022.
- [ngi] nginx. What is a web server?
- [Rea22] React.js. React thư viện javascript xây dựng giao diện người dùng, 2021, truy cập ngày June 20, 2022.
- [Wik] Wikipedia. Domain name system.

Phụ lục

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.