

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng phần mềm quản lý bán hàng đa kênh

VŨ VĂN DŨNG

dung.vv160721@sis.hust.edu.vn

**Ngành Công nghệ thông tin và truyền thông
Chuyên ngành An toàn thông tin**

Giảng viên hướng dẫn: PGS. TS. Lã Thế Vinh

Chữ kí GVHD

Khoa: Kỹ thuật máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 08/2022

ĐỀ TÀI TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Vũ Văn Dũng

Điện thoại liên lạc: 0326200500

Email: dung.vv160721@sis.hust.edu.vn

Lớp: ATTT-K61

Hệ đào tạo: Đại học chính quy

Đề án tốt nghiệp (ĐATN) được thực hiện tại: Trường Đại học Bách

Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 04/06/2020 đến ngày 08/08/2021

2. Mục đích nội dung của ĐATN

Xây dựng, đề xuất và triển khai phần mềm quản lý bán hàng đa kênh. Sử dụng các phương pháp, công nghệ để cải thiện hiệu năng và khả năng mở rộng của hệ thống.

3. Các nhiệm vụ cụ thể của ĐATN

Tìm hiểu đánh giá về thực trạng phần mềm quản lý bán hàng

Phân tích xây dựng các chức năng cơ bản của phần mềm

Thực nghiệm, đánh giá và tìm hiểu các công nghệ xây dựng, tối ưu phần mềm

4. Lời cam đoan của sinh viên

Tôi – Vũ Văn Dũng – cam kết ĐATN này là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của PGS. TS. Lã Thế Vinh. Các kết quả nêu trong ĐATN là trung thực, không sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày 08 tháng 08 năm 2022

Tác giả ĐATN

Vũ Văn Dũng

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ

Hà Nội, ngày 08 tháng 08 năm 2022

Giảng viên hướng dẫn

PGS. TS. Lã Thế Vinh

LỜI CẢM ƠN

Tôi xin chân thành cảm ơn toàn thể thầy cô trong trường Đại học Bách Khoa Hà Nội đặc biệt là các thầy cô trong Viện Công nghệ thông tin và truyền thông đã chỉ dạy cho tôi kiến thức về các môn đại cương, chuyên ngành, giúp tôi có nền tảng và kiến thức chung lẫn chuyên ngành trong suốt những năm học tập và rèn luyện tại trường.

Tôi xin chân thành cảm ơn thầy hướng dẫn khóa luận của tôi là thầy PGS. TS. Lã Thế Vinh, đã tận tình chỉ bảo, hướng dẫn, động viên tôi trong suốt quá trình thực hiện đồ án tốt nghiệp này. Tôi xin chân thành cảm ơn, các anh chị cấp trên, các bạn đồng nghiệp trong đội ngũ nhân viên phát triển ứng dụng của Công ty Pancake Việt Nam.

Tôi cũng xin chân thành cảm ơn gia đình, người thân và bạn bè đã quan tâm, động viên và giúp đỡ tôi trong suốt thời gian học tập và thực hiện đồ án tốt nghiệp này.

Tôi xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Phần mềm quản lý bán hàng là một hệ thống được sử dụng để giảm chi phí hoạt động đồng thời thúc đẩy hoạt động kinh doanh. Lợi ích của phần mềm quản lý bán hàng bao gồm giao nhiệm vụ cho nhân viên, quản lý luồng công việc, và theo dõi tiến độ công việc,... cùng điểm qua những lợi ích tuyệt vời từ việc sử dụng phần mềm quản lý bán hàng.

Trong phạm vi đồ án, tác giả đề xuất xây dựng một phần mềm quản lý bán hàng đa kênh dựa trên cơ sở là những phần mềm đã có, từ đó nâng cấp bằng những công nghệ mới giúp cải thiện hiệu năng của phần mềm nhằm nâng cao tính bảo mật dữ liệu và khả năng mở rộng của toàn bộ hệ thống.

Nội dung đồ án gồm những phần sau:

Chương 1: Giới thiệu đề tài

Đặt vấn đề về tầm quan trọng của phần mềm, từ đó đưa ra mục tiêu và hướng tiếp cận cho đồ án

Chương 2: Khảo sát và phân tích yêu cầu

Đưa ra các khảo sát về thực trạng của phần mềm và phân tích các chức năng cần có

Chương 3: Công nghệ sử dụng

Các công nghệ nền tảng để xây dựng phần mềm

Chương 4: Thực nghiệm và đánh giá

Quá trình thực nghiệm, đánh giá và nhận xét kết quả thu được

Chương 5: Các giải pháp và đóng góp nổi bật

Các công nghệ, giải pháp giúp tối ưu, cải thiện hiệu năng phần mềm

Chương 6: Kết luận và hướng phát triển

Tổng kết đóng góp của đồ án, khó khăn trong quá trình thực hiện và đưa ra hướng phát triển trong tương lai.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu, phạm vi đề tài và hướng giải pháp	1
1.3 Bố cục đồ án	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	3
2.1 Khảo sát hiện trạng	3
2.2 Tổng quan chức năng	4
2.2.1 Biểu đồ use case tổng quát	5
2.2.2 Biểu đồ use case phân rã Manage account	6
2.2.3 Biểu đồ use case phân rã Manage orders	7
2.2.4 Biểu đồ use case phân rã Manage products	8
2.2.5 Biểu đồ use case phân rã Manage warehouses.....	9
2.2.6 Biểu đồ use case phân rã Manage customers	10
2.3 Đặc tả chức năng	10
2.3.1 Đặc tả use case Manage orders	10
2.3.2 Đặc tả use case Mange products	11
2.3.3 Đặc tả use case Manage warehouses	11
2.3.4 Đặc tả use case Manage customers	12
2.4 Yêu cầu phi chức năng	12
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	13
3.1 ReactJS.....	13
3.2 Elixir.....	16
3.3 PosgresSQL	18

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNG GIÁ	20
4.1 Thiết kế kiến trúc.....	20
4.2 Phân tích thiết kế giao diện.....	22
4.3 Phân tích cơ sở dữ liệu	25
4.3.1 Xác định các thực thể trong hệ thống	25
4.3.2 Mô tả chi tiết các thực thể	26
4.4 Minh họa các chức năng chính	31
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	37
5.1 Tìm kiếm bằng Elasticsearch	37
5.1.1 Bài toán	37
5.1.2 Giải pháp	37
5.1.3 Kết quả đạt được	39
5.2 Hiển thị danh sách dữ liệu lớn với React Virtualized	39
5.2.1 Bài toán	39
5.2.2 Giải pháp	39
5.3 Quản lý hàng đợi với RabbitMQ.....	40
5.3.1 Bài toán	40
5.3.2 Giải pháp	40
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	42
6.1 Đóng góp của đồ án	42
6.2 Hướng phát triển.....	42
6.3 Kết luận	42
TÀI LIỆU THAM KHẢO.....	43

DANH MỤC HÌNH VẼ

Hình 2.1	Use case tổng quát hệ thống	5
Hình 2.2	Use case phân rã Manage account	6
Hình 2.3	Use case phân rã Manage orders	7
Hình 2.4	Use case phân rã Manage products	8
Hình 2.5	Use case phân rã Manage warehouses	9
Hình 2.6	Use case phân rã Manage customers	10
Hình 3.1	DOM ảo và DOM thực	14
Hình 3.2	JSX	15
Hình 3.3	Chức năng chính của Elixir	16
Hình 3.4	Lợi ích khi sử dụng Elixir	17
Hình 4.1	Mô hình kiến trúc MVC	20
Hình 4.2	Ví dụ về lược đồ (schema) trong Model	22
Hình 4.3	Ví dụ về View	22
Hình 4.4	Kiến trúc của Redux	24
Hình 4.5	Sơ đồ quan hệ giữa các thực thể	25
Hình 4.6	Thực thể users	26
Hình 4.7	Thực thể shops	27
Hình 4.8	Thực thể orders	28
Hình 4.9	Thực thể customers	29
Hình 4.10	Thực thể products	30
Hình 4.11	Giao diện quản lý đơn hàng	31
Hình 4.12	Giao diện tạo đơn hàng	32
Hình 4.13	Giao diện quản lý sản phẩm	33
Hình 4.14	Chi tiết thông tin khách hàng	33
Hình 4.15	Giao diện tích hợp kênh bán	34
Hình 4.16	Giao diện tích hợp thanh toán	34
Hình 4.17	Cấu hình nhân viên	35
Hình 4.18	Giao diện thống kê	36
Hình 5.1	Cơ chế hoạt động của Elasticsearch	38
Hình 5.2	Cơ chế hoạt động của RabbitMQ	40

DANH MỤC BẢNG BIỂU

Bảng 4.1	Ý nghĩa thuộc tính thực thể users	26
Bảng 4.2	Ý nghĩa thuộc tính thực thể shops	27
Bảng 4.3	Ý nghĩa thuộc tính thực thể orders	28
Bảng 4.4	Ý nghĩa thuộc tính thực thể customers	29
Bảng 4.5	Ý nghĩa thuộc tính thực thể products	30

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
DBMS	Hệ thống quản trị cơ sở dữ liệu (Database Management System)
DOM	Mô hình Các Đối tượng Tài liệu (Document Object Model)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
JSON	Một dạng dữ liệu tuân theo quy luật (JavaScript Object Notation)

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Ngày nay, thương mại điện tử ở Việt Nam đang được Nhà nước và Chính phủ rất quan tâm và đặt mục tiêu thúc đẩy phát triển mạnh mẽ. Với sự phát triển vượt bậc của công nghệ cả trong lẫn ngoài nước, càng có thêm nhiều lý do để các doanh nghiệp quan tâm đến ứng dụng thương mại điện tử. Thương mại điện tử làm cho một tổ chức kinh doanh không bị tụt hậu trong cuộc đua công nghệ. Tuy nhiên, hiện nay Việt Nam chưa có nhiều doanh nghiệp ứng dụng đúng đắn thương mại điện tử vào hoạt động kinh doanh của mình.

Hiện nay, các kỹ thuật và công cụ để thiết lập ứng dụng vào thương mại điện tử vẫn chưa được áp dụng rộng rãi trong các doanh nghiệp. Không chỉ vậy, gần đây với sự bùng nổ của mua sắm online thì việc quản lý khách hàng cũng như quản lý sản phẩm là một thách thức lớn đối với nhiều chủ doanh nghiệp bán hàng trên nhiều nền tảng cùng một lúc.

Bài toán đặt ra ở đây là một nền tảng có tập chung có thể quản lý mọi thông tin từ các nguồn mạng xã hội, sàn thương mại điện tử đổ về sẽ giúp doanh nghiệp tiết kiệm nhiều thời gian quản lý và chi phí điều hành doanh nghiệp.

1.2 Mục tiêu, phạm vi đề tài và hướng giải pháp

Thương mại điện tử là một lĩnh vực mới trong vài năm trở lại đây nên các ứng dụng quản lý bán hàng đa kênh hiện nay còn khá hạn chế. Bên cạnh đó, để quản lý tệp khách mua hàng, chăm sóc lại khách đã mua cũng như quản lý tồn kho của sản phẩm trong cùng một ứng dụng hiện nay chưa có nhiều ứng dụng đáp ứng được yêu cầu của doanh nghiệp.

Vì vậy cách giải quyết những vấn đề tồn đọng vừa đưa ra là một phần mềm quản lý, lớn hơn là hệ sinh thái các phần mềm bổ trợ cho nhau nhằm phục vụ, quản lý về mọi mặt thông tin về buôn bán thương mại điện tử. Điều này sẽ giúp doanh nghiệp tối ưu hóa được quy trình làm việc, tiết kiệm chi phí và nhân công, từ đó cải thiện doanh thu và lợi nhuận.

Để xây dựng một phần mềm quản lý cũng như hệ thống quản lý bán hàng như trên cần một hệ thống lớn, luôn sẵn sàng và có khả năng mở rộng ngoài ra cần một đội ngũ lớn có nhiều kinh nghiệm. Trên thực tế, phần mềm quản lý bán hàng đòi hỏi một hệ thống phục vụ realtime có khả năng chịu lỗi, và có khả năng mở rộng lớn. Vì vậy đội ngũ của tôi đã lựa chọn một số công nghệ đáp ứng các yêu cầu trên và có cộng đồng đông đảo nhiều lập trình viên sử dụng. Ở phần này, chúng tôi đã

lựa chọn một hệ thống lưu trữ cơ sở dữ liệu có khả năng mở rộng theo chiều ngang để tối ưu hóa tốc độ truy vấn và có khả năng phân tích realtime, công nghệ này sẽ được nhắc đến cụ thể ở chương sau.

Ngoài ra, việc lựa chọn ngôn ngữ lập trình phù hợp, phần mềm quản lý source code cũng là những yếu tố quan trọng trong quá trình xây dựng phần mềm của đội ngũ chúng tôi. Ngôn ngữ lập trình được chọn xây dựng và biên dịch dựa trên nền tảng máy ảo để tối ưu tốc độ và đáp ứng khả năng xử lý lỗi của hệ thống.

1.3 Bố cục đề án

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về khảo sát hiện trạng của phần mềm từ đó đưa ra các phân tích và đánh giá về yêu cầu của phần mềm quản lý bán hàng. Từ đó khái quát hóa các tính năng chính mà phần mềm cần có. Trong chương này cũng gồm các Use case và đặc tả chức năng nhằm phân tích chi tiết một số chức năng quan trọng của phần mềm.

Trong chương 3 là những công nghệ nền tảng được sử dụng trong đề án. Các công cụ này được lựa chọn dựa trên tính phù hợp, đề cao khả năng mở rộng của hệ thống.

Kiến trúc hệ thống đóng vai trò vô cùng quan trọng khi triển khai một hệ thống lớn nên việc chọn một kiến trúc phù hợp sẽ được nêu rõ trong Chương 4 của đề án. Ngoài ra chương này còn trình bày thiết kế chi tiết về cơ sở dữ liệu, các thư viện chính mà hệ thống sử dụng.

Chương 5 là những đóng góp nổi bật, những công nghệ hay mà đề án đã sử dụng giúp cải thiện được nhiều vấn đề mà các phần mềm khác còn tồn đọng.

Cuối cùng là Chương 6 nêu ra kết luận và trình bày những hướng phát triển mới cho đề án.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

Trong thời đại công nghệ 4.0, khi xu hướng mua hàng online trở thành hình thức thương mại phổ biến nhất, các doanh nghiệp bắt đầu gặp phải những vấn đề khó khăn trong hoạt động kinh doanh trực tuyến. Điều này đòi hỏi sự hỗ trợ của các công cụ với tính năng tiện ích vượt trội, và đây chính là lúc phần mềm quản lý bán hàng ra đời.

Phần mềm quản lý bán hàng là hệ thống các công cụ được lập trình với những tính năng thông minh ưu việt, có khả năng hỗ trợ doanh nghiệp thực hiện toàn bộ các công đoạn của quá trình bán hàng và quản lý bán hàng một cách hiệu quả, toàn diện và nhanh chóng nhất. Bao gồm:

Quản lý hàng hóa, kho hàng và tình trạng xuất – nhập kho hàng

Quản lý hoạt động bán hàng trên cả nền tảng online lẫn offline

Tạo lập và quản lý đơn hàng

Quản lý thông tin và hoạt động của nhân viên cũng như các đối tác

Quản lý thông tin khách hàng và các kế hoạch khuyến mãi, hậu mãi

Thống kê lịch sử mua hàng, tồn kho, báo cáo tài chính

Với các công cụ và cách thức xử lý truyền thống, doanh nghiệp phải mất quá nhiều thời gian, nhân lực và chi phí cho hoạt động quản lý bán hàng nhưng vẫn không tránh khỏi các sai sót. Nhưng giờ đây, bạn có thể hoàn toàn yên tâm vì mọi số liệu đều được xử lý một cách tự động, với độ chính xác tuyệt đối.

Đối với doanh nghiệp, hệ thống quản lý bán hàng là một trong những công cụ hỗ trợ đắc lực cho quá trình phát triển kinh doanh. Không chỉ giúp tiết kiệm tối đa thời gian và chi phí cho công tác cập nhật, theo dõi, thống kê, xử lý số liệu,... mỗi ngày, mà nó còn đảm bảo hiệu quả trong việc tính toán một khối lượng dữ liệu khổng lồ chỉ trong thời gian ngắn.

Bên cạnh đó, doanh nghiệp còn có thể sử dụng phần mềm để tập trung tất cả các nguồn lực về một mối để lưu trữ thông tin và quản lý tập trung hiệu quả hơn. Từ đó, nâng cao sự gắn kết và phối hợp giữa các phòng ban, giúp quá trình vận hành kinh doanh diễn ra mượt mà và mang lại hiệu suất cao nhất.

Một ưu điểm nổi bật của phần mềm quản lý bán hàng là tính linh hoạt và khả năng áp dụng không giới hạn. Đặc biệt, doanh nghiệp không cần hệ thống máy

móc công kênh, các phương thức tính toán bằng não lồi thời hay lo sợ thất lạc dữ liệu nữa. Vì tất cả đều được xử lý nhanh gọn và lưu trữ với độ bảo mật cao nhất.

Một phần mềm quản lý kinh doanh không chỉ dừng lại ở các tính năng quản lý cơ bản. Nó cần đáp ứng được nhiều yêu cầu hơn thế nữa. Ví dụ như tích hợp phương thức thanh toán thông minh, hình thức vận chuyển đa dạng hay khả năng kết nối với các thiết bị khác để tối đa hóa hiệu quả bán hàng tại chỗ.

Ưu điểm nổi bật khi sử dụng phần mềm Pancake POS

Với những tính năng ưu việt vừa liệt kê bên trên, sở hữu giải pháp quản lý bán hàng Pancake POS sẽ mang đến cho doanh nghiệp rất nhiều lợi ích thiết thực.

Đầu tiên và dễ thấy nhất, chúng ta có thể bỏ qua hệ thống máy móc công kênh, tốn kém, cũng như các phương thức quản lý truyền thống rườm rà nhưng hiệu quả mang lại không khả quan là mấy.

Các tính năng có sẵn trong Pancake POS sẽ giúp doanh nghiệp quản lý toàn bộ hoạt động kinh doanh dễ dàng, nhanh chóng hơn chỉ với vài thao tác đơn giản trên giao diện phần mềm thân thiện, trực quan và dễ sử dụng. Từ đó, tiết kiệm tối đa công sức và nguồn nhân lực, đồng thời đẩy mạnh hiệu quả bán hàng đến mức cao nhất.

2.2 Tổng quan chức năng

Các chức năng chính trong phần mềm quản lý bán hàng Pancake POS:

Quản lý cửa hàng: Mỗi người dùng được tạo nhiều cửa hàng, có thể đặt tên, sửa xóa thông tin cửa hàng. Cấu hình các thuộc tính đặc thù như quốc gia kinh doanh, tiền tệ, phương thức thanh toán...

Quản lý đơn hàng: Cho phép tạo đơn thủ công, hoặc cấu hình tạo đơn tự động, đồng bộ đơn hàng từ website, Facebook, các sàn thương mại điện tử. Đơn hàng có thể liên kết với các đơn vị vận chuyển và cập nhật trạng thái theo hành trình.

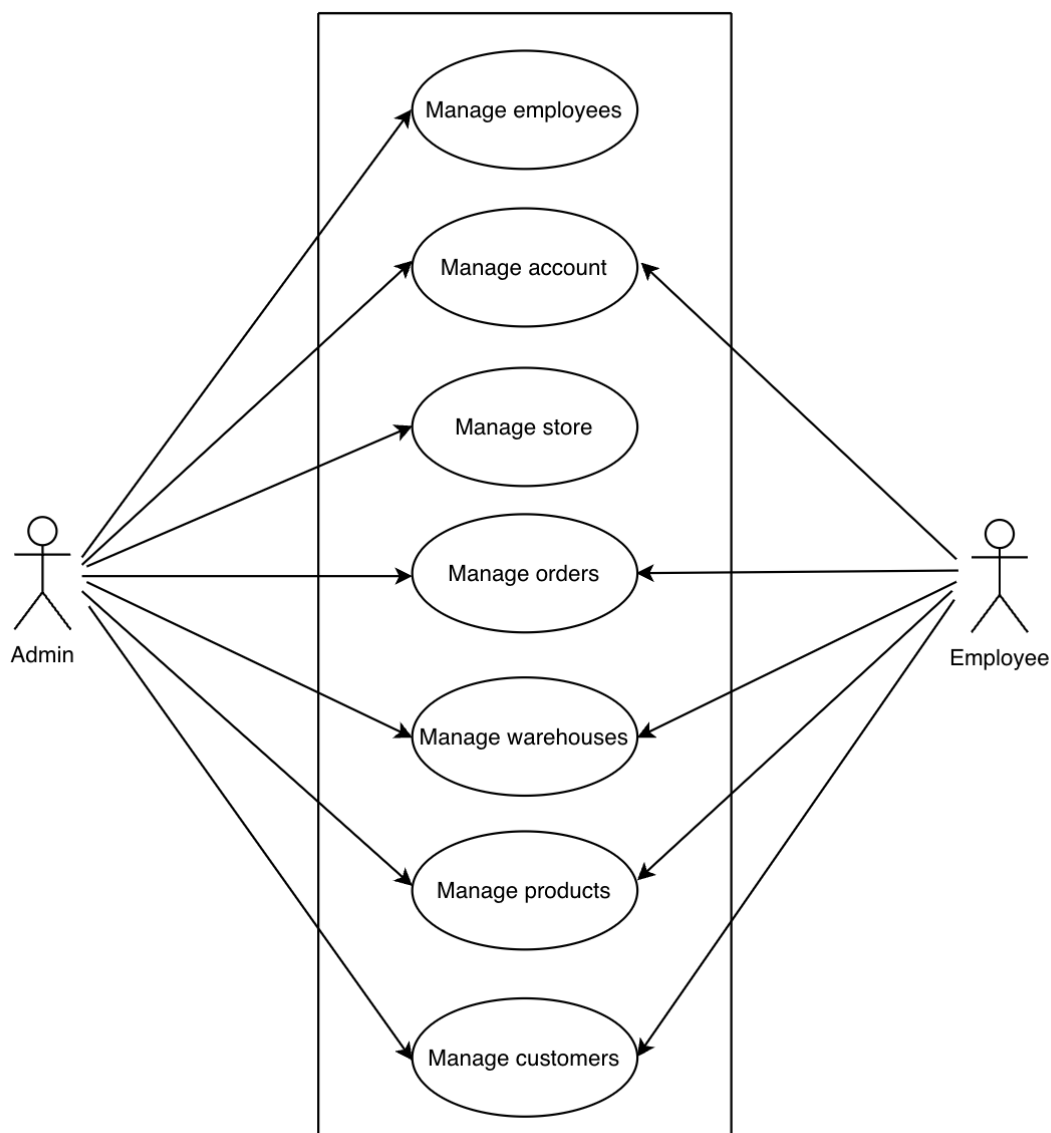
Quản lý khách hàng của cửa hàng: Lưu trữ thông tin những khách đã mua hàng như tên, ngày sinh, số điện thoại. Mỗi khách hàng gồm một hoặc nhiều đơn hàng. Khách hàng sẽ bao gồm với những tính năng quản lý công nợ, quản lý điểm thưởng.

Quản lý sản phẩm: Cho phép nhập liệu thông tin, thuộc tính sản phẩm, danh mục. Có thể nhập liệu từ các nguồn excel, đồng bộ với các sàn thương mại điện tử Shopee, Lazada, TikTok. Sản phẩm liên kết các thông tin tồn kho, lưu lại log xuất nhập kho

Quản lý nhân viên: Nhân viên gồm các thông tin tên, ID, email, các quyền của nhân viên, ngoài ra nhân viên còn thuộc quản lý của các bộ phận và các kho

Quản lý kho: Gồm các thông tin cơ bản như tên, địa chỉ, ID. Có những thao tác thêm, sửa, xóa. Kho gồm các sản phẩm thuộc kho, có thể tạo phiếu xuất nhập và kiểm kho, báo cáo tồn kho theo ngày. Quản lý hệ thống lô kệ trong kho.

2.2.1 Biểu đồ use case tổng quát



Hình 2.1: Use case tổng quát hệ thống

Các tác nhân gồm người dùng và nhân viên:

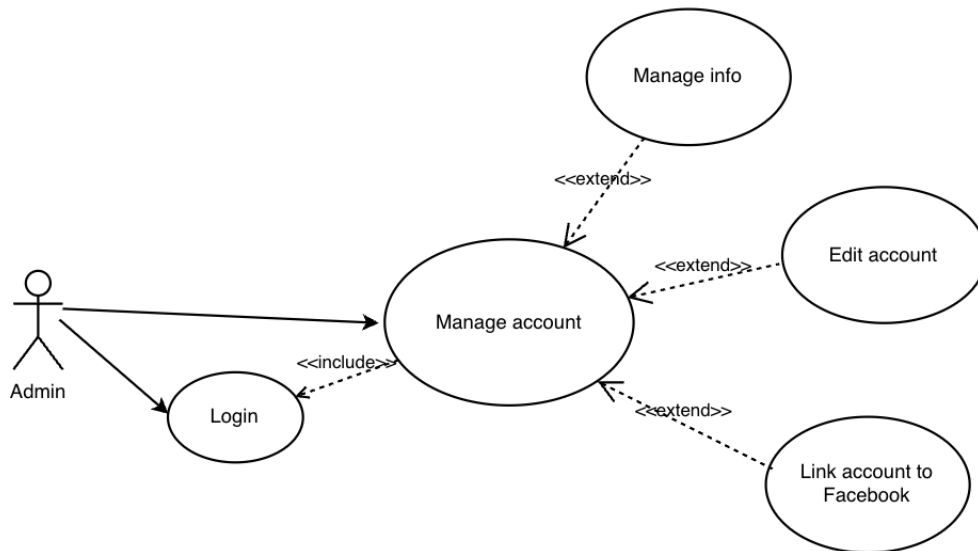
Người dùng thường là quản trị viên. Sau khi đăng nhập vào hệ thống sẽ có thông tin quản lý về người dùng, người dùng có quyền quản lý các cửa hàng, đơn hàng, kho, sản phẩm, khách hàng. Người dùng có quyền quản lý nhân viên và phân quyền cho các nhân viên.

Nhân viên có hầu như các quyền như người dùng, nhưng sẽ phụ thuộc vào quyền

mà người dùng cài đặt theo mỗi cửa hàng cho từng nhân viên. Có thể nhân viên có quyền quản lý đơn hàng ở cửa hàng nhưng lại không có quyền quản lý kho hàng.

Một số use case chính của hệ thống là: Quản lý nhân viên, quản lý tài khoản, quản lý đơn hàng, quản lý cửa hàng, quản lý kho, quản lý sản phẩm và quản lý khách hàng. Các chức năng của use case này đã được mô tả qua trong phần Tổng quan chức năng 2.2

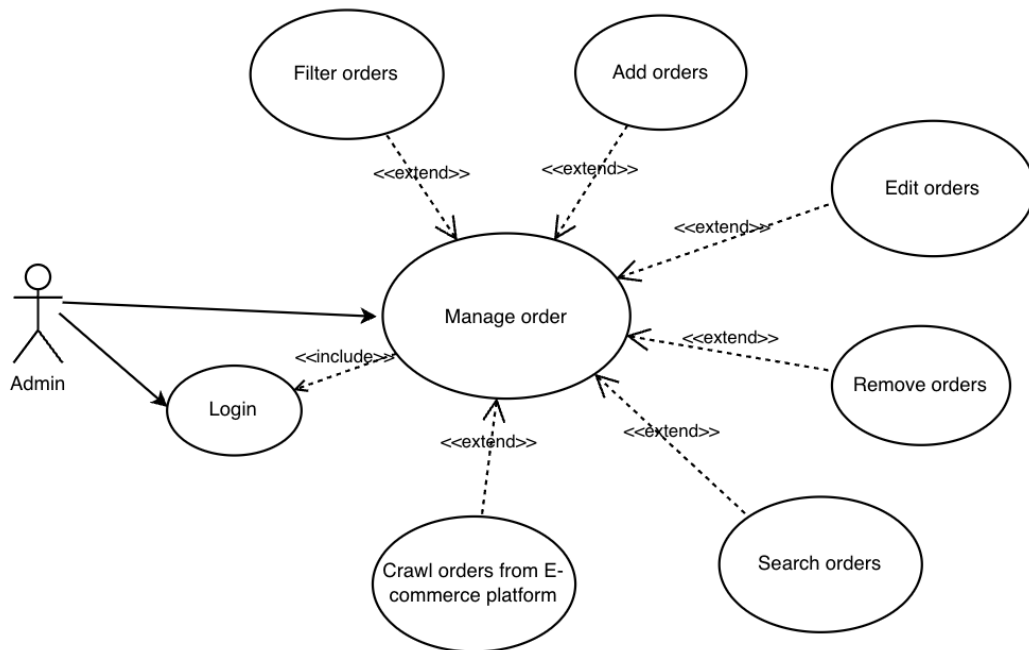
2.2.2 Biểu đồ use case phân rã Manage account



Hình 2.2: Use case phân rã Manage account

Sau khi đăng nhập chủ cửa hàng (Admin) có thể quản lý các thông tin về tài khoản. Sửa thông tin tài khoản như tên, avatar, ngôn ngữ. Liên kết với tài khoản Facebook

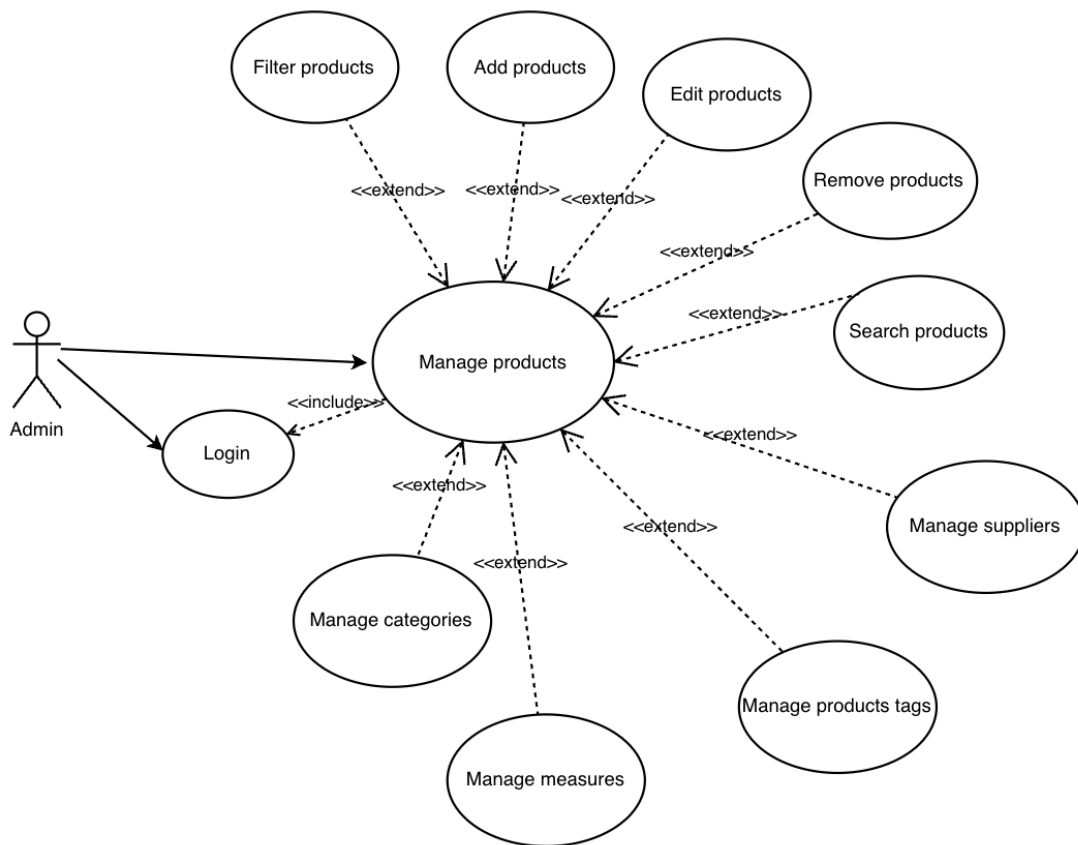
2.2.3 Biểu đồ use case phân rã Manage orders



Hình 2.3: Use case phân rã Manage orders

Quản lý đơn hàng bao gồm các tính năng cơ bản thêm, sửa, xóa, lọc tìm kiếm đơn hàng. Đồng bộ đơn hàng từ các nền tảng thương mại điện tử. Tác nhân ở đây là nhân viên của cửa hàng có thể là Admin hoặc employee có quyền.

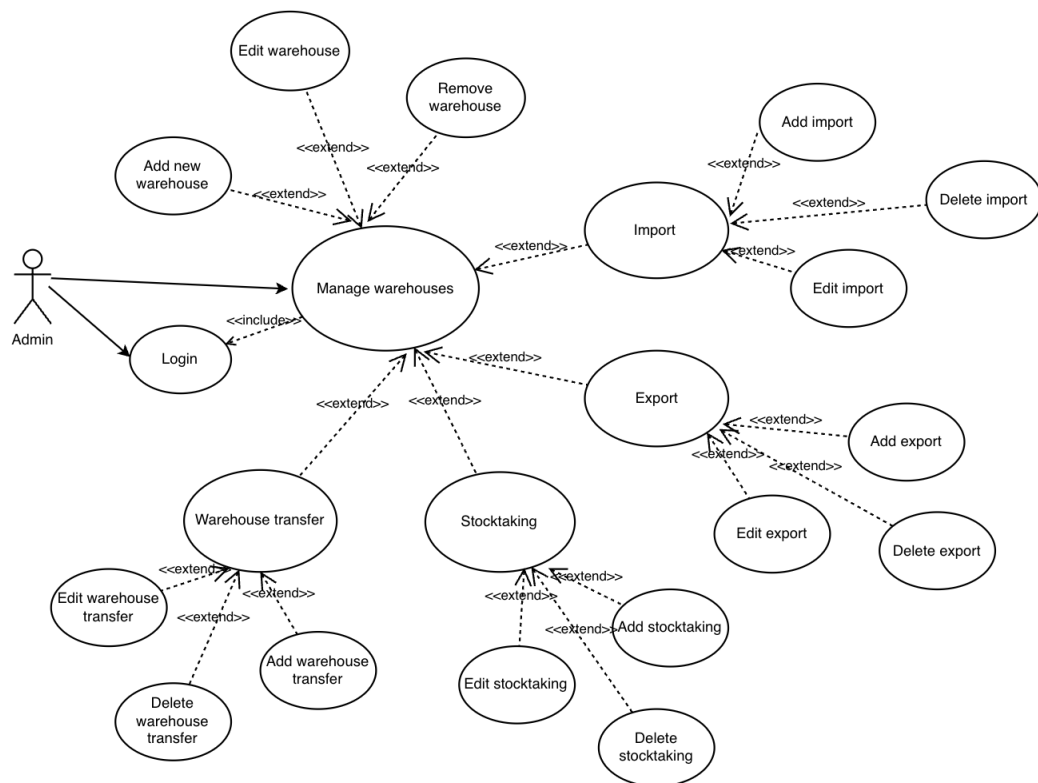
2.2.4 Biểu đồ use case phân rã Manage products



Hình 2.4: Use case phân rã Manage products

Tác nhân tham gia vào use case quản lý sản phẩm là chủ cửa hàng (Admin) hoặc nhân viên trong cửa hàng nếu được Admin phân quyền quản lý sản phẩm. Quản lý sản phẩm gồm các hành động thêm, sửa, xóa, lọc, tìm kiếm sản phẩm. Quản lý sản phẩm bao gồm quản lý danh mục, đơn vị tính, thẻ sản phẩm và đơn vị cung cấp.

2.2.5 Biểu đồ use case phân rã Manage warehouses



Hình 2.5: Use case phân rã Manage warehouses

Quản lý kho hàng gồm tác nhân Admin là nhân viên có quyền trong cửa hàng. Bao gồm các chức năng thêm sửa xóa kho hàng. Ngoài ra quản lý kho hàng bao gồm:

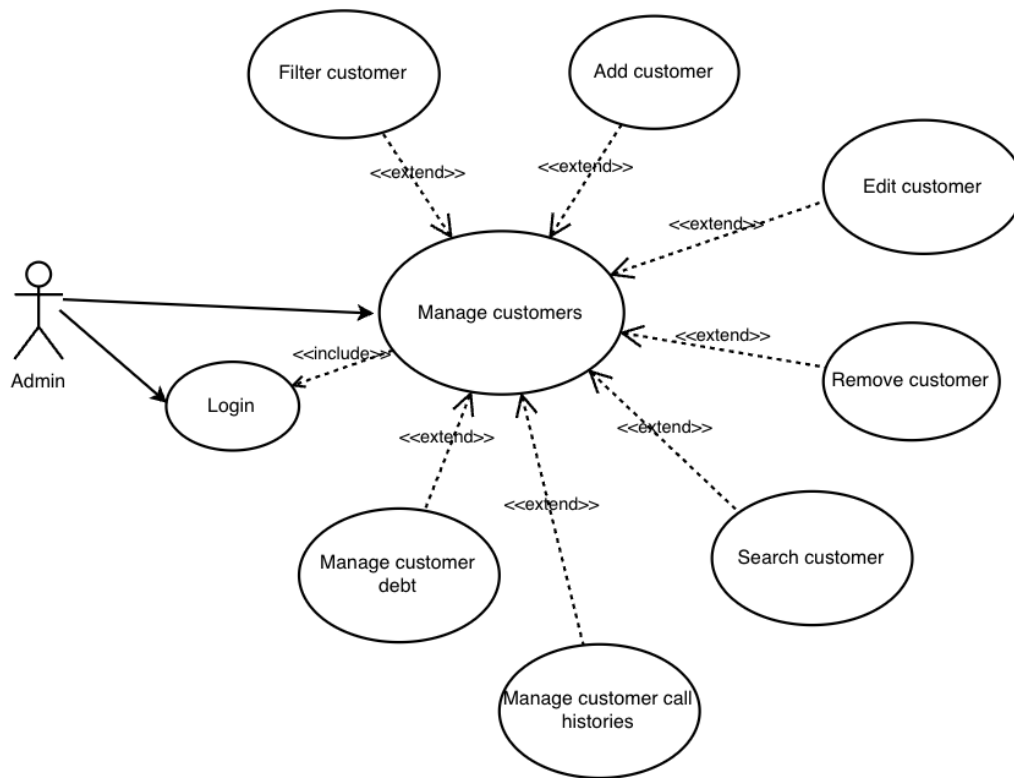
Quản lý nhập hàng: Thêm, sửa, xóa phiếu nhập hàng

Quản lý xuất hàng: Thêm, sửa, xóa phiếu xuất hàng

Quản lý kiểm hàng: Thêm, sửa, xóa phiếu kiểm hàng

Quản lý chuyển kho: Thêm, sửa, xóa phiếu chuyển giữa các kho

2.2.6 Biểu đồ use case phân rã Manage customers



Hình 2.6: Use case phân rã Manage customers

Quản lý khách hàng cũng bao gồm tác nhân là chủ cửa hàng và nhân viên có quyền quản lý. Sau khi đăng nhập thì tác nhân có quyền thêm, sửa, xóa khách hàng. Ngoài ra quản lý khách hàng bao gồm quản lý công nợ khách hàng và quản lý danh sách lịch sử gọi cho khách hàng.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Manage orders

Tên Use case: Manage orders

Tác nhân: Quản trị viên

Tiền điều kiện: Cần phải đăng nhập vào hệ thống

Luồng sự kiện: Use case này bắt đầu khi quản trị viên chọn phần quản lý thông tin đơn hàng.

(1) Hiện thị danh sách đơn hàng của cửa hàng mà quản trị viên đã truy cập, có thể thực hiện các thao tác thêm, sửa, xóa, lọc, tìm kiếm đơn hàng

(2) Người quản lý nhấn vào danh sách đơn hàng để xem chi tiết thông tin của

đơn hàng đó.

2.3.2 Đặc tả use case Mange products

Tên Use case: Manage products

Tác nhân: Quản trị viên

Tiền điều kiện: Cần phải đăng nhập vào hệ thống

Luồng sự kiện: Use case này bắt đầu khi quản trị viên chọn phần quản lý thông tin sản phẩm.

Sự kiện chính:

(1) Hiển thị danh sách sản phẩm của cửa hàng mà quản trị viên đã truy cập, có thể thực hiện các thao tác thêm, sửa, xóa, lọc, tìm kiếm sản phẩm

(2) Người quản lý nhấn vào danh sách sản phẩm để xem chi tiết thông tin của sản phẩm đó.

Sự kiện phát sinh:

(1) Quản trị viên thao tác bấm vào quản lý danh mục, nhà cung cấp, thẻ sản phẩm, đơn vị tính để cấu hình cho sản phẩm.

2.3.3 Đặc tả use case Manage warehouses

Tên Use case: Manage warehouses

Tác nhân: Quản trị viên

Tiền điều kiện: Cần phải đăng nhập vào hệ thống

Luồng sự kiện: Use case này bắt đầu khi quản trị viên chọn phần quản lý thông tin kho hàng. Hoặc quản trị viên thực hiện các thao tác xuất nhập hàng, kiểm kho, chuyển kho.

Sự kiện chính:

(1) Hiển thị danh sách kho hàng của cửa hàng mà quản trị viên đã truy cập, có thể thực hiện các thao tác thêm, sửa, xóa, tìm kiếm kho hàng

(2) Người quản lý nhấn vào danh sách kho hàng để xem chi tiết thông tin của kho hàng đó.

Sự kiện phát sinh:

(1) Quản trị viên thao tác bấm vào quản lý xuất nhập hàng, kiểm kho, chuyển kho

(2) Tạo, sửa, xóa các phiếu điều chỉnh kho hàng

2.3.4 Đặc tả use case Manage customers

Tên Use case: Manage customers

Tác nhân: Quản trị viên

Tiền điều kiện: Cần phải đăng nhập vào hệ thống

Luồng sự kiện: Use case này bắt đầu khi quản trị viên chọn phần quản lý thông tin khách hàng.

Sự kiện chính:

(1) Hiện thị danh sách khách hàng của cửa hàng mà quản trị viên đã truy cập, có thể thực hiện các thao tác thêm, sửa, xóa, lọc, tìm kiếm khách hàng

(2) Người quản lý nhấn vào danh sách khách hàng để xem chi tiết thông tin của khách hàng đó.

Sự kiện phát sinh:

(1) Quản trị viên thao tác bấm vào quản lý công nợ khách hàng hoặc quản lý lịch sử cuộc gọi từ khách hàng

2.4 Yêu cầu phi chức năng

Phù hợp với các kích thước màn hình khác nhau (máy tính xách tay, điện thoại di động, ...).

Các chức năng phải rõ ràng và dễ sử dụng.

Mã nguồn phải được viết khoa học để dễ bảo trì.

Đảm bảo an toàn dữ liệu người mua hàng của cửa hàng.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 ReactJS

React là một thư viện của JavaScript để xây dựng giao diện người dùng. Nhằm mục đích hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, sử dụng trạng thái (state) và được sử dụng nhiều lần.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM.

Tại sao lựa chọn React

Dễ dàng tạo các ứng dụng động: React giúp tạo các ứng dụng web động dễ dàng hơn vì nó yêu cầu ít mã hóa hơn và cung cấp nhiều chức năng hơn.

Cải thiện hiệu suất: React sử dụng Virtual DOM, do đó tạo các ứng dụng web nhanh hơn. Virtual DOM so sánh trạng thái trước đó của các thành phần và chỉ cập nhật các mục trong DOM thực đã được thay đổi.

Các thành phần có thể tái sử dụng: Các thành phần là khối xây dựng của bất kỳ ứng dụng React nào và một ứng dụng đơn lẻ thường bao gồm nhiều thành phần. Các thành phần này có logic và điều khiển của chúng và chúng có thể được sử dụng lại trong toàn bộ ứng dụng, do đó làm giảm đáng kể thời gian phát triển của ứng dụng.

Luồng dữ liệu một chiều: React tuân theo luồng dữ liệu một chiều. Điều này có nghĩa là khi thiết kế một ứng dụng React, các nhà phát triển thường lồng các thành phần con vào bên trong các thành phần mẹ. Vì dữ liệu chảy theo một hướng nên việc gỡ lỗi trở nên dễ dàng hơn và biết được vị trí xảy ra sự cố trong ứng dụng tại thời điểm được đề cập.

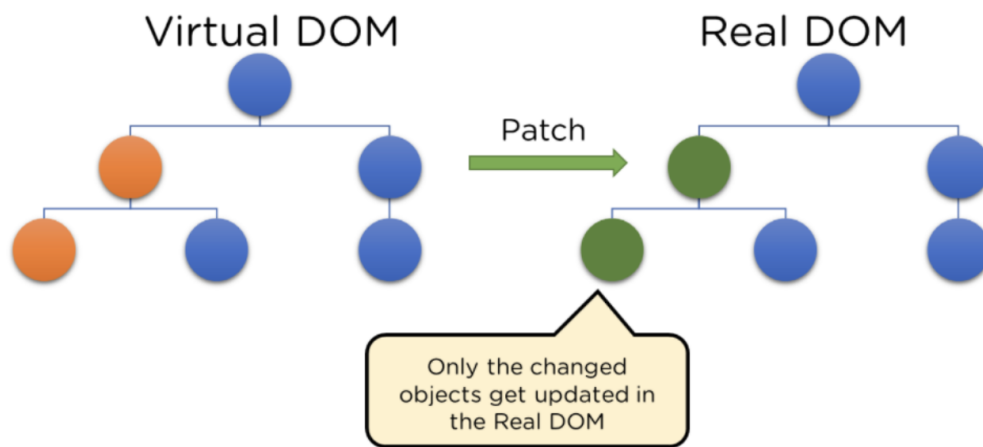
Nó có thể được sử dụng để phát triển cả ứng dụng web và ứng dụng dành cho thiết bị di động

Các công cụ chuyên dụng để gỡ lỗi dễ dàng: Facebook đã phát hành một tiện ích mở rộng của Chrome có thể được sử dụng để gỡ lỗi các ứng dụng React. Điều này làm cho quá trình gỡ lỗi các ứng dụng web React nhanh hơn và dễ dàng hơn.

Virtual Document Object Model (DOM)

DOM ảo (VDOM) là một khái niệm lập trình trong đó một biểu diễn lý tưởng hay còn gọi là “ảo” của giao diện người dùng được lưu trong bộ nhớ và được đồng

bộ hóa với DOM “thực”. Thao tác DOM thực về cơ bản chậm hơn so với thao tác DOM ảo. Khi trạng thái của một đối tượng thay đổi, Virtual DOM chỉ cập nhật đối tượng đó trong DOM thực thay vì tất cả chúng.



Hình 3.1: DOM ảo và DOM thực

Document Object Model (DOM) coi một tài liệu XML hoặc HTML như một cấu trúc cây, trong đó mỗi nút là một đối tượng đại diện cho một phần của tài liệu.

Khi trạng thái của một đối tượng thay đổi trong ứng dụng React, VDOM sẽ được cập nhật. Sau đó, nó so sánh trạng thái trước đó của nó và sau đó chỉ cập nhật các đối tượng đó trong DOM thực thay vì cập nhật tất cả các đối tượng.

Điều này làm cho mọi thứ di chuyển nhanh chóng, đặc biệt là khi so sánh với các công nghệ front-end khác phải cập nhật từng đối tượng ngay cả khi chỉ một đối tượng thay đổi trong ứng dụng web.

Vì React sử dụng liên kết dữ liệu một chiều, nên tất cả các hoạt động luôn ở dạng mô-đun và nhanh chóng. Hơn nữa, luồng dữ liệu một chiều có nghĩa là chúng ta thường lồng các thành phần con bên trong các thành phần mẹ khi phát triển một dự án React.

JSX



Hình 3.2: JSX

JSX là một phần mở rộng cú pháp JavaScript. Đó là một thuật ngữ được sử dụng trong React để mô tả giao diện người dùng trông như thế nào. Bạn có thể viết cấu trúc HTML trong cùng một tệp như mã JavaScript bằng cách sử dụng JSX.

```
const name = 'Simplilearn';
```

```
const greet = <h1>Hello, name</h1>;
```

Đoạn mã trên cho thấy cách JSX được triển khai trong React. Nó không phải là một chuỗi hay HTML. Thay vào đó, nó nhúng HTML vào mã JavaScript.

Components

Các component là các khối xây dựng bao gồm một ứng dụng React đại diện cho một phần của giao diện người dùng. React tách giao diện người dùng thành nhiều component, giúp việc gỡ lỗi trở nên dễ tiếp cận hơn và mỗi component có một bộ thuộc tính và chức năng riêng. Dưới đây là một số tính năng của Component:

Khả năng sử dụng lại: Một component được sử dụng trong một khu vực của ứng dụng có thể được sử dụng lại trong một khu vực khác. Điều này giúp đẩy nhanh quá trình phát triển.

Các component lồng nhau: Một component có thể chứa một số component khác.

Phương thức kết xuất: Ở dạng tối thiểu, một component phải xác định một phương thức kết xuất chỉ định cách thành phần kết xuất với DOM.

Thuộc tính props: Một thành component có thể nhận được props. Đây là những thuộc tính được cha của nó truyền vào để chỉ định các giá trị cho component con.

Props và State

Props là viết tắt của Properties (thuộc tính) trong React, cho phép người dùng truyền các đối số hoặc dữ liệu cho các thành phần. Các props này giúp làm cho các thành phần năng động hơn. Props trong một component là chỉ đọc và không thể

thay đổi.

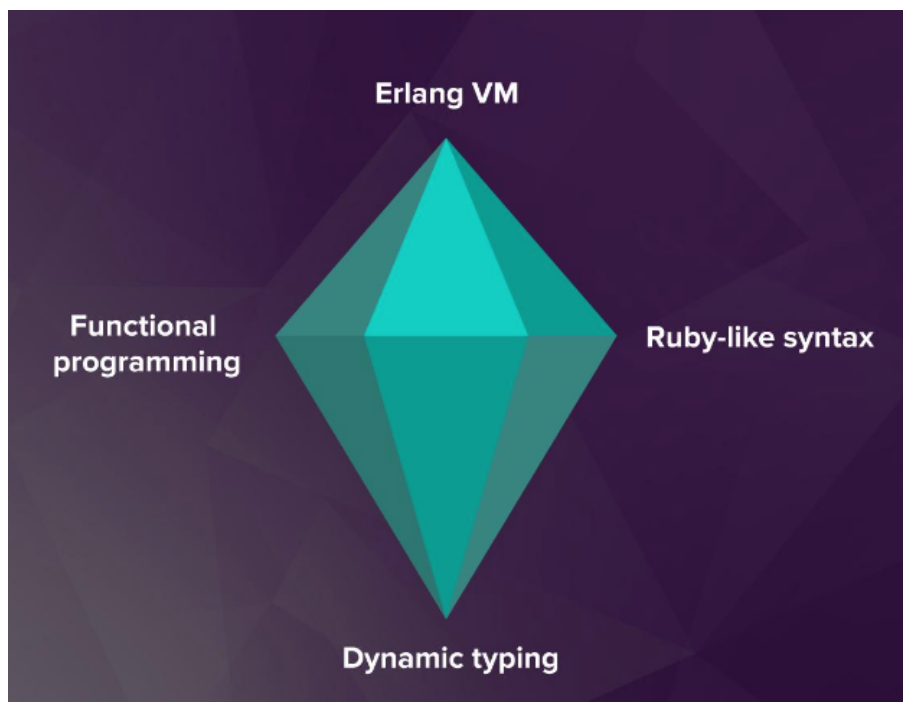
State là một đối tượng lưu trữ các giá trị thuộc tính cho những thuộc tính được gán cho một thành phần có thể thay đổi theo thời gian. State có thể được thay đổi do hành động của người dùng hoặc các thay đổi trong mạng. React kết xuất thành phần đó cho trình duyệt bất cứ khi nào trạng thái của một đối tượng thay đổi.

3.2 Elixir

Elixir là một ngôn ngữ chức năng (functional), được nhập động (dynamically typed), được xây dựng trên máy ảo của Erlang và biên dịch thành mã bytecode của Erlang.

Ngôn ngữ lập trình Elixir kết hợp lập trình chức năng với trạng thái bất biến và cách tiếp cận dựa trên tác nhân để đồng thời theo một cú pháp gọn gàng, hiện đại. Và nó chạy trên máy ảo Erlang phân tán, hiệu suất cao, công nghiệp.

Các tính năng chính của Elixir



Hình 3.3: Chức năng chính của Elixir

Được xây dựng trên máy ảo Erlang. Elixir có quyền truy cập vào tất cả các công cụ đồng thời mà Erlang có quyền truy cập, khiến nó trở thành một trong những ngôn ngữ hiện đại mạnh mẽ nhất để xây dựng các hệ thống phân tán, có thể mở rộng.

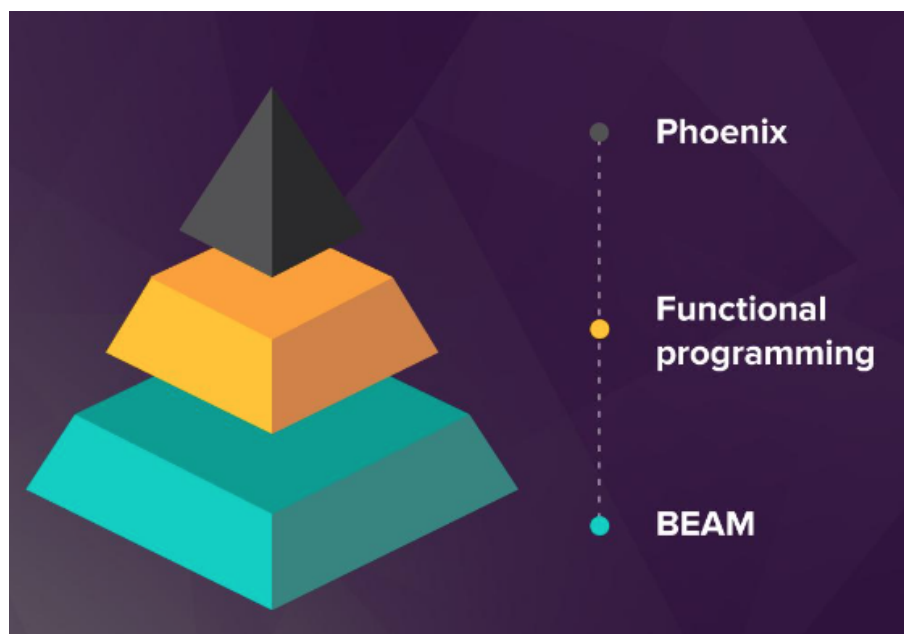
Elixir là một trong những ngôn ngữ định hướng năng suất và ngắn gọn nhất hiện có, nhưng nó thiếu hiệu suất và tính đồng thời. Erlang VM giải quyết cả hai vấn đề

này.

Cấu trúc dữ liệu không thay đổi giúp đồng thời khá nhiều, và đối sánh mẫu rất tốt để viết mã khai báo.

Có kiểu gõ động (dynamically typed). Điều này làm tăng tốc độ phát triển cho các ứng dụng web đơn giản. Các kiểu tĩnh có thể được giới thiệu trong Elixir thông qua các tệp kiểu chữ.

Lợi ích khi sử dụng Elixir



Hình 3.4: Lợi ích khi sử dụng Elixir

BEAM

Elixir được xây dựng dựa trên BEAM, Erlang VM, và nó có chung những điểm trừu tượng đã làm cho Erlang trở thành một trong những lựa chọn tốt nhất cho các ứng dụng đồng thời, khối lượng lớn. tương tự với Erlang, Elixir được thừa hưởng các đặc điểm sau:

Tính đồng thời (Concurrency). Elixir sử dụng các luồng thực thi nhẹ. Chúng được tách biệt, chạy trên tất cả các CPU và giao tiếp thông qua các tin nhắn. Cùng với tính bất biến của dữ liệu đến từ bản chất chức năng của ngôn ngữ, điều này làm cho việc viết các chương trình đồng thời trong Elixir trở nên bớt khó khăn hơn.

Khả năng mở rộng (Scalability). Các tiến trình tương tự này cho phép mở rộng quy mô hệ thống một cách dễ dàng, theo chiều ngang hoặc theo chiều dọc.

Độ tin cậy (Reliability). Elixir có một cách tiếp cận độc đáo để chịu lỗi. Trong khi mọi thứ đôi khi không thể tránh khỏi lỗi trong quá trình sản xuất, các quy trình

nhẹ có thể được khởi động lại nhanh chóng bởi hệ thống giám sát. Điều này giải quyết hầu hết các lỗi không phải do lỗi nghiêm trọng trong quá trình phát triển phần mềm.

Functional programming

Lập trình hàm là một mô hình lập trình xử lý các chương trình như đánh giá các hàm toán học và tránh những thứ như dữ liệu có thể thay đổi và trạng thái thay đổi. Hỗ trợ và tận dụng những thứ như đối sánh mẫu, các hàm bậc cao hơn và kiểu viết mã khai báo, và điều đó đi kèm với nhiều lợi ích:

Khả năng bảo trì tốt hơn

Kiểm tra và gỡ lỗi rõ ràng hơn

Xử lý dễ dàng hơn các chương trình đồng thời

Trừu tượng cấp cao giúp viết và hiểu mã đơn giản hơn

Phoenix Framework

Phoenix dễ sử dụng với mọi kiến trúc MVC web framework, và nó cũng được sao lưu bởi BEAM. Nó cho phép một người tạo ra các ứng dụng web một cách hiệu quả trong khi không ảnh hưởng đến tốc độ hoặc khả năng bảo trì.

Ngoài ra, các quy trình của Elixir / Erlang phù hợp để giải quyết các vấn đề về tính đồng thời và khả năng mở rộng trong các dịch vụ web. Mặc dù các kênh Phoenix có thể đạt được 2 triệu kết nối websocket trên một hộp duy nhất, nhưng dung lượng có thể được nhân lên bằng cách phân phối lưu lượng đến nhiều nút hơn.

3.3 PostgreSQL

PostgreSQL là một cơ sở dữ liệu quan hệ. Nó lưu trữ các điểm dữ liệu trong các hàng, với các cột là các thuộc tính dữ liệu khác nhau. Một bảng lưu trữ nhiều hàng liên quan. Cơ sở dữ liệu quan hệ là loại cơ sở dữ liệu phổ biến nhất được sử dụng. Nó khác biệt với việc tập trung vào tích hợp và khả năng mở rộng. Nó hoạt động với nhiều công nghệ khác và phù hợp với các tiêu chuẩn cơ sở dữ liệu khác nhau, đảm bảo nó có thể mở rộng được.

Tại sao sử dụng PostgreSQL

Một cơ sở dữ liệu cấp doanh nghiệp, PostgreSQL tự hào có các tính năng phức tạp như Kiểm soát đồng thời nhiều phiên bản (MVCC), khôi phục tại thời điểm, không gian bảng, sao chép không đồng bộ, giao dịch lồng nhau, sao lưu trực tuyến / nóng, trình lập kế hoạch / tối ưu hóa truy vấn phức tạp và ghi nhật ký trước cho khả năng chịu lỗi.

PostgreSQL hoạt động trên hầu hết các hệ điều hành phổ biến - hầu như tất cả

các bản phân phối Linux và Unix, Windows, Mac OS X. Bản chất nguồn mở của nó giúp dễ dàng nâng cấp hoặc mở rộng. Trong PostgreSQL, bạn có thể xác định kiểu dữ liệu của riêng mình, xây dựng các hàm tùy chỉnh và thậm chí viết mã bằng ngôn ngữ lập trình khác mà không cần biên dịch lại cơ sở dữ liệu.

Một cơ sở dữ liệu đáng tin cậy

PostgreSQL không chỉ là quan hệ, mà còn là đối tượng-quan hệ và hỗ trợ các cấu trúc phức tạp và nhiều loại dữ liệu được tích hợp sẵn và do người dùng xác định.

Nó cung cấp dung lượng dữ liệu lớn và được tin cậy vì tính toàn vẹn của dữ liệu. Điều này mang lại cho nó một số lợi thế so với các cơ sở dữ liệu SQL mã nguồn mở khác như MySQL, MariaDB và Firebird. PostgreSQL đi kèm với nhiều tính năng nhằm giúp các nhà phát triển xây dựng ứng dụng, quản trị viên để bảo vệ tính toàn vẹn của dữ liệu và xây dựng môi trường chịu lỗi.

Cung cấp cho người dùng một số lượng lớn các chức năng. Những điều này giúp các lập trình viên tạo các ứng dụng mới, quản trị viên bảo vệ tính toàn vẹn của dữ liệu tốt hơn và các nhà phát triển xây dựng các môi trường linh hoạt và an toàn. PostgreSQL cung cấp cho người dùng khả năng quản lý dữ liệu, bất kể cơ sở dữ liệu lớn như thế nào.

The extensible database

Một cơ sở dữ liệu có khả năng mở rộng

Ngoài việc miễn phí và mã nguồn mở, PostgreSQL có khả năng mở rộng cao. Có hai lĩnh vực mà PostgreSQL tỏa sáng khi người dùng cần cấu hình và kiểm soát cơ sở dữ liệu của họ. Đầu tiên, nó tuân thủ ở mức độ cao với các tiêu chuẩn SQL. Điều này làm tăng khả năng tương tác của nó với các ứng dụng khác.

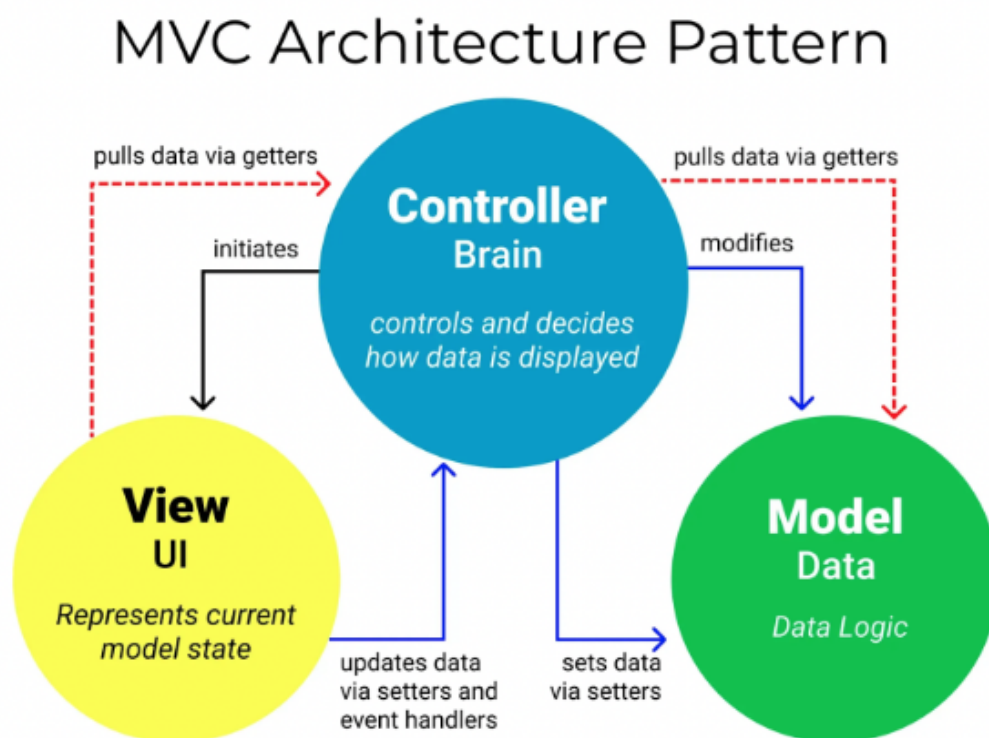
Thứ hai, PostgreSQL cung cấp cho người dùng quyền kiểm soát siêu dữ liệu. PostgreSQL có thể mở rộng được vì hoạt động của nó là theo hướng danh mục. Một điểm khác biệt chính giữa PostgreSQL và các hệ thống cơ sở dữ liệu quan hệ tiêu chuẩn là PostgreSQL lưu trữ nhiều thông tin hơn trong danh mục của nó: không chỉ thông tin về bảng và cột, mà còn thông tin về kiểu dữ liệu, hàm, phương thức truy cập, v.v.

Người dùng có thể sửa đổi các bảng này và vì PostgreSQL dựa trên hoạt động của nó trên các bảng này, điều này có nghĩa là PostgreSQL có thể được mở rộng bởi người dùng. Để so sánh, các hệ thống cơ sở dữ liệu thông thường chỉ có thể được mở rộng bằng cách thay đổi các thủ tục được mã hóa cứng trong mã nguồn hoặc bằng cách tải các mô-đun do nhà cung cấp DBMS viết riêng.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNG GIÁ

4.1 Thiết kế kiến trúc

Kiến trúc MVC



Hình 4.1: Mô hình kiến trúc MVC

MVC (Model-View-Controller) là một mẫu thiết kế kiến trúc khuyến khích tổ chức ứng dụng được cải tiến thông qua việc tách biệt các mối quan tâm. Nó chia một ứng dụng tương tác thành ba thành phần: Model / View và Controller.

Nó thực thi cách ly dữ liệu kinh doanh (Models) khỏi giao diện người dùng (Views), với thành phần thứ ba (Controllers) quản lý logic, người dùng nhập và điều phối cả mô hình và chế độ xem theo cách truyền thống. Mục tiêu của MVC là giúp cấu trúc các mối quan tâm riêng biệt của một ứng dụng thành ba phần:

Model chịu trách nhiệm quản lý dữ liệu của ứng dụng. Nó nhận đầu vào của người dùng từ bộ điều khiển. Model chứa các phương thức để xử lý dữ liệu đến từ database.

View có nghĩa là bản trình bày của mô hình ở một định dạng cụ thể. Là nơi sẽ hiển thị các dữ liệu đến với người dùng. Phần View chứa tất cả các chức năng mà người dùng có thể tương tác được như nút bấm hoặc một thao tác kéo thả.

Controller phản hồi đầu vào của người dùng và thực hiện các tương tác trên các đối tượng mô hình dữ liệu. Bộ điều khiển nhận đầu vào, xác nhận tùy chọn và sau đó chuyển đầu vào cho mô hình. Phần Controller được ví như bộ não của ứng dụng. Controller sử dụng để liên kết hai phần model và view với nhau bằng cách cập nhật các dữ liệu ở phần model từ các event của phần view.

Mục đích của việc lựa chọn kiến trúc MVC

Như đã đề cập ở trên, kiến trúc MVC giúp chúng ta phân tách các khía cạnh khác nhau của ứng dụng (logic đầu vào, logic nghiệp vụ và GUI), đồng thời cung cấp một khớp nối lỏng lẻo giữa các phần tử này. Do đó, logic thông tin có thể tái sử dụng nhiều nhất thuộc về model, GUI thuộc về view. Logic đầu vào thuộc controller. Sự tách biệt này giúp chúng ta quản lý khi xây dựng một ứng dụng phức tạp vì nó cho phép ta tập trung vào một khía cạnh của việc triển khai tại một thời điểm.

MVC tách các thành phần khác nhau của một ứng dụng, các nhà phát triển có thể làm việc song song trên các thành phần khác nhau mà không ảnh hưởng hoặc chặn lẫn nhau.

Khả năng tái sử dụng. Chế độ xem giống nhau (hoặc tương tự) cho một ứng dụng có thể được cấu trúc lại cho một ứng dụng khác với dữ liệu khác vì chế độ xem chỉ đơn giản là xử lý cách dữ liệu được hiển thị cho người dùng.

Khả năng mở rộng. Nếu ứng dụng của chúng ta bắt đầu gặp vấn đề về hiệu suất do truy cập cơ sở dữ liệu chậm, bạn có thể nâng cấp phần cứng chạy cơ sở dữ liệu mà không bị ảnh hưởng đến các thành phần khác. Vì các thành phần có mức độ phụ thuộc vào nhau thấp, việc thực hiện các thay đổi đối với một (để sửa lỗi hoặc thay đổi chức năng) không ảnh hưởng đến các thành phần khác

Bản chất của khuôn khổ MVC là có sự liên kết thấp giữa các model, view hoặc controller.

Áp dụng kiến trúc MVC vào ứng dụng

Phần Model chủ yếu xử lý việc xác thực dữ liệu, lược đồ của nó, mối quan hệ của nó với các mô hình khác và cách nó được trình bày. Việc chỉ định lược đồ (schema) trong model nó cho phép chúng ta dễ dàng tạo các trường ảo, là các trường không được lưu giữ trong cơ sở dữ liệu.

```
defmodule HelloPhoenix.User do
  use HelloPhoenix.Web, :model

  schema "users" do
    field :name, :string
    field :email, :string
    field :password, :string, virtual: true
    field :password_hash, :string
  end
end
```

Hình 4.2: Ví dụ về lược đồ (schema) trong Model

Ở đây chúng ta khai báo bốn trường trong bảng users: name, email, password, và password-hash. Và "password" là được cài đặt là trường ảo.

Điều này cho phép chúng ta thiết lập và lấy trường này mà không cần lưu các thay đổi vào cơ sở dữ liệu. Điều này rất hữu ích vì chúng ta sẽ có logic để chuyển đổi mật khẩu đó thành một hàm băm, mà chúng ta sẽ lưu vào trường "password-hash" và sau đó lưu vào cơ sở dữ liệu.

```
defmodule HelloWorld.HelloView do
  use HelloWorld.Web, :view

  def render("index.html", _assigns) do
    raw("<h1>Hello, World!</h1>")
  end
end
```

Hình 4.3: Ví dụ về View

Phần View chịu trách nhiệm kết xuất các mẫu và cung cấp các chức năng giúp dữ liệu thô dễ sử dụng hơn cho các mẫu. Ở đây trong hàm render có trả về JSON cho front-end xử lý dữ liệu.

Controller xử lý các logic thêm, sửa, xóa dữ liệu, các hàm xử lý tính toán thông qua việc gọi model để lấy các trường từ database.

4.2 Phân tích thiết kế giao diện

Để xây dựng một ứng dụng web hoàn toàn hoạt động hiệu quả, các mã code và thành phần phải được viết và kiểm tra một cách cẩn thận. Code phải rõ ràng và đơn giản để đọc và dễ hiểu. Qua đó cần chọn lựa các thư viện và công cụ thích hợp với dự án.

Figma

Figma là công cụ được ra mắt vào năm 2016, với giao diện thân thiện và tính dễ sử dụng, Figma đã nhanh chóng nổi lên và trở thành một công cụ thiết kế giao

diện người dùng phổ biến trong cộng đồng công nghệ toàn cầu. Một số thương hiệu lớn sử dụng Figma cho tới thời điểm hiện tại có thể kể đến như Microsoft, Twitter, GitHub, Dropbox...

Khác với những công cụ thiết kế trước đây, Figma được thiết kế trên nền tảng đám mây. Đây là công cụ có tính năng tương tự như Sketch, ưu điểm nổi bật của công cụ là khả năng tương tác thời gian thực, cho phép hỗ trợ làm việc nhóm tốt hơn. Designer, Leader, Developer và thậm chí là cả khách hàng có thể tương tác, trao đổi trực tiếp với nhau thông qua tính năng comment giúp quá trình làm việc trở nên nhanh chóng.

Webpack

Webpack là một module bundler cho phép viết bất kỳ định dạng module nào (cũng có thể kết hợp), sau đó biên dịch cho trình duyệt. Webpack phân tách mã dựa trên cách nó được sử dụng trong ứng dụng và với phân tích trách nhiệm theo module này, việc quản lý, gỡ lỗi, xác minh và kiểm tra mã trở nên dễ dàng.

Một số tính năng nổi bật của Webpack là:

Giúp cho cho project dễ dàng quản lý.

Theo dõi các thay đổi và chạy lại các tác vụ.

Phân chia các module và chỉ load khi cần; Đóng gói tất cả file nguồn thành một file duy nhất, nhờ vào loader mà có thể biên dịch các loại file khác nhau.

Có thể chạy chuyển đổi Babel sang ES5, cho phép bạn sử dụng các tính năng JavaScript mới nhất mà không cần lo lắng về sự hỗ trợ của trình duyệt.

Có thể chạy như một development webserver.

Ant Design

Ant Design là tập hợp các components của React được xây dựng theo chuẩn thiết kế của Ant UED Team. Tương tự như chuẩn Material Design, Ant cung cấp hầu hết các component thông dụng trong ứng dụng web hiện đại, như Layout, Button, Icon, DatePicker,...

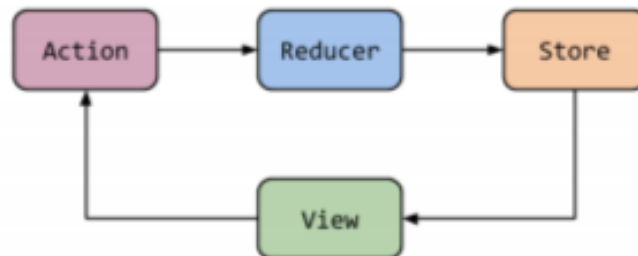
Redux

Redux là một kho chứa trạng thái mà có thể dự đoán được cho các ứng dụng JavaScript. Nó giúp bạn viết các ứng dụng hoạt động nhất quán, chạy trong các môi trường khác nhau (client, server và native) và dễ kiểm tra.

Nói một cách đơn giản, Redux là một công cụ quản lý trạng thái. Mặc dù nó chủ yếu được sử dụng để React, nhưng nó có thể được sử dụng với bất kỳ khung hoặc thư viện JavaScript nào khác. Nó có dung lượng nhẹ ở 2KB, vì vậy bạn không phải

lo lắng về việc nó làm cho kích thước nội dung ứng dụng của bạn lớn hơn.

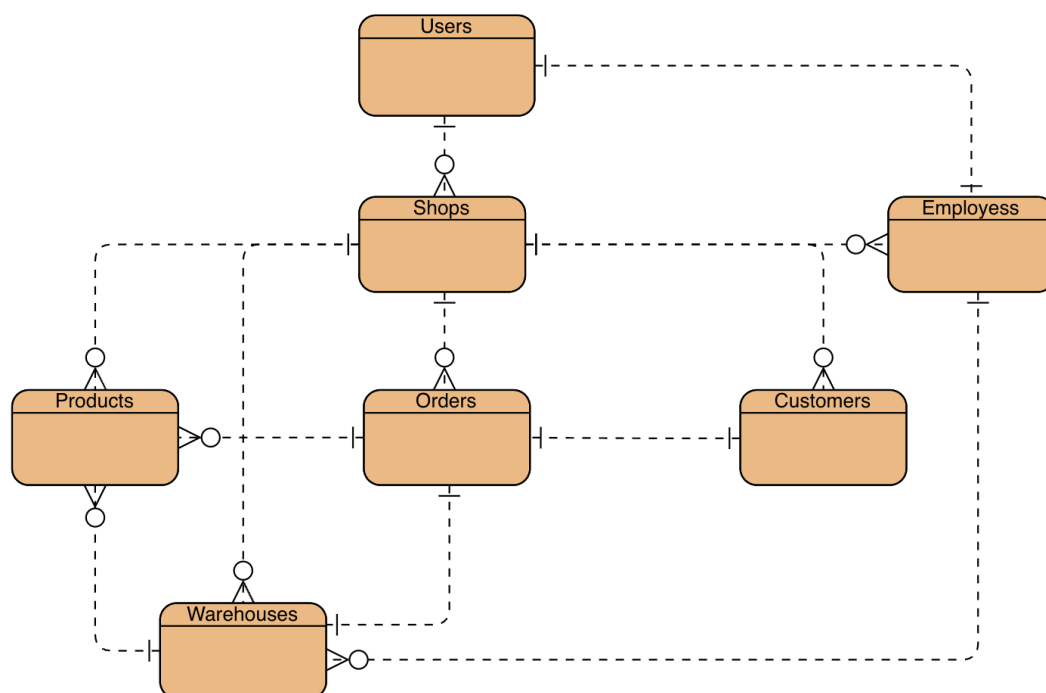
Với Redux, trạng thái của ứng dụng của bạn được lưu trong một store và mỗi thành phần có thể truy cập vào bất kỳ trạng thái nào mà nó cần từ store này. Hãy đi sâu hơn một chút để biết lý do tại sao bạn có thể cần một công cụ quản lý state.



Hình 4.4: Kiến trúc của Redux

Chúng ta có một component view (React) gửi một hành động để lưu trữ, store sẽ chuyển hành động đó cho reducer với thông tin về trạng thái hiện tại. Sau đó, reducer sẽ gửi lại trạng thái mới dựa trên hành động, sau đó store sẽ cập nhật trạng thái của nó. Sau khi cập nhật store, chế độ xem thành phần sẽ nhận dữ liệu mới và hiển thị lại. Store thường chứa trạng thái của nó với kiểu không thay đổi được để ngăn việc thay đổi dữ liệu bên ngoài Redux.

4.3 Phân tích cơ sở dữ liệu



Hình 4.5: Sơ đồ quan hệ giữa các thực thể

Qua sơ đồ quan hệ giữa các thực thể phía trên chúng ta có thể thấy mối quan hệ tương quan giữa các thực thể có trong hệ thống. Một Users có thể quản lý nhiều Shops. Một Users tương ứng với một Employees có trong Shops. Một Shops sẽ bao gồm nhiều Employees, Products, Orders, Customers, Warehouses. Và tương ứng với một Orders chỉ có một Customers và một Warehouses. Một Orders có thể có nhiều Products.

4.3.1 Xác định các thực thể trong hệ thống

Ta xác định được các thực thể trong cơ sở dữ liệu, dưới đây là vài thực thể chính của ứng dụng:

users: Lưu trữ thông tin về người dùng.

shops: Lưu trữ thông tin, cấu hình về các gian hàng.

orders: Lưu trữ thông tin về đơn hàng.

customers: Lưu trữ thông tin về khách mua hàng.

products: Lưu trữ thông tin về sản phẩm.

4.3.2 Mô tả chi tiết các thực thể

Thực thể users

Thực thể users lưu trữ thông tin về người dùng. Các thuộc tính của thực thể users bao gồm:

Name	users	Primary	id	Column name... (KF)			
#	column_name	data_type	is_nullable	check	column_default	foreign_key	comment
1	id	uuid	NO	NULL	NULL	EMPTY	NULL
2	name	varchar(255)	YES	NULL	NULL	EMPTY	NULL
3	email	varchar(255)	YES	NULL	NULL	EMPTY	NULL
4	fb_id	varchar(32)	YES	NULL	NULL	EMPTY	NULL
5	access_token	text	YES	NULL	NULL	EMPTY	NULL
6	password_hash	varchar(255)	YES	NULL	NULL	EMPTY	NULL
7	phone_number	varchar(255)	YES	NULL	NULL	EMPTY	NULL
8	group	varchar(64)	YES	NULL	'USER':character varying	EMPTY	NULL
10	inserted_at	timestamp(0)	NO	NULL	NULL	EMPTY	NULL
11	updated_at	timestamp(0)	NO	NULL	NULL	EMPTY	NULL
13	shipper_balance	int4	YES	NULL	0	EMPTY	NULL
14	app_id	int4	YES	NULL	1	EMPTY	NULL
15	admin_tag	jsonb	YES	NULL	NULL	EMPTY	NULL
16	locale	varchar(255)	YES	NULL	NULL	EMPTY	NULL
17	is_intl	bool	YES	NULL	false	EMPTY	NULL
18	stripe_token	varchar(255)	YES	NULL	NULL	EMPTY	NULL
19	third_party_meta_data	jsonb	YES	NULL	NULL	EMPTY	NULL
20	timezone	varchar(255)	YES	NULL	NULL	EMPTY	NULL
21	signatures	jsonb	YES	NULL	NULL	EMPTY	NULL
22	avatar_url	text	YES	NULL	NULL	EMPTY	NULL
23	country	varchar(255)	YES	NULL	NULL	EMPTY	NULL

Hình 4.6: Thực thể users

Ý nghĩa một số thuộc tính quan trọng của thực thể:

Tên thuộc tính	Mô tả
id	Mã người dùng
name	Tên đầy đủ của người dùng
fb_id	Facebook ID khi khách đang nhập bằng Facebook
access_token	phiên đăng nhập
password_hash	Mật khẩu đã được Hash
phone_number	Số điện thoại
avatar_url	Đường dẫn ảnh đại diện của người dùng
inserted_at	Ngày tạo tài khoản
updated_at	Ngày sửa tài khoản
locale	Khu vực của người dùng
timezone	Múi giờ

Bảng 4.1: Ý nghĩa thuộc tính thực thể users

Thực thể shops

Thực thể Shops lưu trữ thông tin, cấu hình về các gian hàng. Các thuộc tính của thực thể user bao gồm:

Name: shops Primary: id						
#	column_name	data_type	is_nullable	check	column_default	foreign_key
1	id	int8	NO	NULL	nextval('shops_id_seq'::regclass)	EMPTY
2	name	varchar(255)	YES	NULL	NULL	EMPTY
3	is_created_by_page	bool	NO	NULL	true	EMPTY
4	creator_id	uuid	YES	NULL	NULL	EMPTY
5	fifo_lifo	fifo_lifo	YES	NULL	'lifo'::fifo_lifo	EMPTY
6	inserted_at	timestamp(0)	NO	NULL	NULL	EMPTY
7	updated_at	timestamp(0)	NO	NULL	NULL	EMPTY
8	use_lock_order	bool	YES	NULL	false	EMPTY
9	business_id	int8	YES	NULL	NULL	EMPTY
10	business_secret_key	varchar(255)	YES	NULL	NULL	EMPTY
11	tags	_varchar	YES	NULL	ARRAY[]::character varying[]	EMPTY
12	tag_colors	_varchar	YES	NULL	ARRAY[]::character varying[]	EMPTY
13	reward_point_settings	jsonb	YES	NULL	NULL	EMPTY
14	is_merging	bool	YES	NULL	false	EMPTY
15	limit_view_order	bool	YES	NULL	false	EMPTY
16	is_snappy_intergration	bool	YES	NULL	true	EMPTY
17	auto_send_order_to_snappy	bool	YES	NULL	false	EMPTY
18	auto_tracking_messages	_jsonb	YES	NULL	ARRAY[]::jsonb[]	EMPTY
19	settings	jsonb	YES	NULL	'{}'::jsonb	EMPTY
20	auto_assign_order	bool	YES	NULL	false	EMPTY
21	limit_time_to_reassign	int4	YES	NULL	15	EMPTY
22	day_offs	varchar(255)	YES	NULL	""::character varying	EMPTY
23	working_time	jsonb	YES	NULL	NULL	EMPTY
24	holiday	_varchar	YES	NULL	ARRAY[]::character varying[]	EMPTY
25	assign_online_user	bool	YES	NULL	true	EMPTY
26	limit_pending_order	int4	YES	NULL	10	EMPTY
27	show_discount_detail	bool	YES	NULL	NULL	EMPTY
28	currency	varchar(255)	YES	NULL	NULL	EMPTY

Hình 4.7: Thực thể shops

Ý nghĩa một số thuộc tính quan trọng của thực thể:

Tên thuộc tính	Mô tả
id	Mã cửa hàng
name	Tên cửa hàng
create_id	Người tạo cửa hàng
inserted_at	Ngày tạo tài khoản
updated_at	Ngày sửa tài khoản
fifo_lifo	phiên đăng nhập
settings	Thông tin cấu hình cửa hàng
currency	Tiền tệ
timezone	Múi giờ cửa hàng
fifo_lifo	Hình thức xuất kho

Bảng 4.2: Ý nghĩa thuộc tính thực thể shops

Thực thể orders

Thực thể Orders lưu trữ thông tin về các đơn hàng của người dùng. Các thuộc tính của thực thể user bao gồm:

Name	orders	Primary	id	Column name... (%F)		
#	column_name	data_type	is_nullable	check	column_default	foreign
1	id	int8	NO	NULL	nextval('orders_id_seq'::regclass)	EMPTY
2	note	text	YES	NULL	NULL	EMPTY
3	note_print	text	YES	NULL	NULL	EMPTY
4	bill_full_name	text	YES	NULL	NULL	EMPTY
5	bill_phone_number	text	YES	NULL	NULL	EMPTY
6	status	int4	YES	NULL	NULL	EMPTY
7	display_id	int4	YES	NULL	NULL	EMPTY
8	display_id_original	int4	YES	NULL	NULL	EMPTY
9	display_id_original_prefix	varchar(255)	YES	NULL	NULL	EMPTY
10	shipping_address	jsonb	YES	NULL	NULL	EMPTY
11	shop_id	int8	YES	NULL	NULL	EMPTY
12	page_id	varchar(64)	YES	NULL	NULL	EMPTY
13	conversation_id	varchar(128)	YES	NULL	NULL	EMPTY
14	warehouse_id	uuid	YES	NULL	NULL	EMPTY
15	creator_id	uuid	YES	NULL	NULL	EMPTY
16	last_editor_id	uuid	YES	NULL	NULL	EMPTY
17	discount	int4	YES	NULL	0	EMPTY
18	prepaid	int4	YES	NULL	0	EMPTY
19	received_at_shop	bool	YES	NULL	false	EMPTY
20	is_removed	bool	YES	NULL	false	EMPTY

Hình 4.8: Thực thể orders

Ý nghĩa một số thuộc tính quan trọng của thực thể:

Tên thuộc tính	Mô tả
id	Mã đơn hàng
note	Ghi chú
bill_full_name	Tên người mua hàng
inserted_at	Ngày tạo đơn hàng
updated_at	Ngày sửa đơn hàng
bill_phone_number	Số điện thoại người mua hàng
status	Trạng thái đơn hàng
display_id	Mã hiển thị đơn hàng
shipping_address	Địa chỉ người mua
shop_id	Đơn hàng thuộc cửa hàng
warehouse_id	Đơn hàng thuộc kho hàng
is_removed	Kiểm tra trạng thái xóa
cod	Tiền COD của đơn hàng
discount	Tiền giảm giá

Bảng 4.3: Ý nghĩa thuộc tính thực thể orders

Thực thể customers

Thực thể customers lưu trữ thông tin về các khách hàng của cửa hàng. Các thuộc tính của thực thể user bao gồm:

Name	customers	Primary	id	Column name... (ZF)			
#	column_name	data_type	is_nullable	check	column_default	foreign_key	comment
1	id	uuid	NO	NULL	NULL	EMPTY	NULL
2	fb_id	varchar(255)	YES	NULL	NULL	EMPTY	NULL
3	global_id	varchar(255)	YES	NULL	NULL	EMPTY	NULL
4	name	text	YES	NULL	NULL	EMPTY	NULL
5	banned_count	int4	YES	NULL	0	EMPTY	NULL
7	reported_count	int4	YES	NULL	0	EMPTY	NULL
8	profile_info	jsonb	YES	NULL	NULL	EMPTY	NULL
9	profile_updated_at	timestamp(0)	YES	NULL	NULL	EMPTY	NULL
10	psid	varchar(255)	YES	NULL	NULL	EMPTY	NULL
11	username	varchar(255)	YES	NULL	NULL	EMPTY	NULL
12	platform	varchar(255)	YES	NULL	NULL	EMPTY	NULL
14	page_id	varchar(255)	YES	NULL	NULL	EMPTY	NULL
15	page_ids	_varchar	YES	NULL	NULL	EMPTY	NULL
16	gender	varchar(255)	YES	NULL	NULL	EMPTY	NULL
17	job	varchar(255)	YES	NULL	NULL	EMPTY	NULL
18	lives_in	varchar(255)	YES	NULL	NULL	EMPTY	NULL
19	relationship_status	varchar(255)	YES	NULL	NULL	EMPTY	NULL
20	inserted_at	timestamp(0)	YES	NULL	NULL	EMPTY	NULL
21	updated_at	timestamp(0)	YES	NULL	NULL	EMPTY	NULL

Hình 4.9: Thực thể customers

Ý nghĩa một số thuộc tính quan trọng của thực thể:

Tên thuộc tính	Mô tả
id	Mã khách hàng
name	Tên khách hàng
fb_id	ID khách hàng trên Facebook qua ứng dụng
global_id	ID khách hàng trên Facebook
inserted_at	Ngày tạo khách hàng
updateted_at	Ngày sửa khách hàng
bill_phone_number	Số điện thoại khách hàng
platform	Phân loại khách hàng
page_id	Phân loại khách theo trang
shipping_address	Địa chỉ khách hàng
shop_id	Khách hàng thuộc cửa hàng
gender	Giới tính khách hàng
profile_info	Thông tin thêm khách hàng

Bảng 4.4: Ý nghĩa thuộc tính thực thể customers

Thực thể products

Thực thể products lưu trữ thông tin về các thông tin sản phẩm của cửa hàng. Các thuộc tính của thực thể products bao gồm:

Name	products	Primary	id	Column name... (%F)			
#	column_name	data_type	is_nullable	check	column_default	foreign_key	
1	id	uuid	NO	NULL	NULL	EMPTY	→
2	name	varchar(255)	YES	NULL	NULL	EMPTY	→
3	product_attributes	_jsonb	YES	NULL	NULL	EMPTY	→
4	description	text	YES	NULL	NULL	EMPTY	→
5	offline_id	varchar(255)	YES	NULL	NULL	EMPTY	→
6	removed	bool	NO	NULL	false	EMPTY	→
7	display_id	int4	YES	NULL	NULL	EMPTY	→
8	display_id_original	int4	YES	NULL	NULL	EMPTY	→
9	display_id_original_prefix	varchar(255)	YES	NULL	NULL	EMPTY	→
10	shop_id	int8	YES	NULL	NULL	EMPTY	→
11	category_ids	_int4	YES	NULL	NULL	EMPTY	→
12	inserted_at	timestamp(0)	NO	NULL	NULL	EMPTY	→
13	updated_at	timestamp(0)	NO	NULL	NULL	EMPTY	→
14	custom_id	varchar(255)	YES	NULL	NULL	EMPTY	→
15	creator_id	uuid	YES	NULL	NULL	EMPTY	→
16	last_editor_id	uuid	YES	NULL	NULL	EMPTY	→
17	shop_warranty_policy_id	int4	YES	NULL	NULL	EMPTY	→
18	warranty_period	int4	YES	NULL	NULL	EMPTY	→
19	third_parties	_varchar	YES	NULL	NULL	EMPTY	→
20	webcms_id	int8	YES	NULL	NULL	EMPTY	→

Hình 4.10: Thực thể products

Ý nghĩa một số thuộc tính quan trọng của thực thể:

Tên thuộc tính	Mô tả
id	Mã sản phẩm
name	Tên sản phẩm
product_attributes	Thuộc tính của sản phẩm
description	Mô tả sản phẩm
inserted_at	Ngày tạo sản phẩm
updated_at	Ngày sửa sản phẩm
removed	Kiểm tra trạng thái đã xóa
display_id	Mã hiển thị
shop_id	Sản phẩm thuộc cửa hàng
category_ids	Danh mục sản phẩm
creator_id	Người tạo

Bảng 4.5: Ý nghĩa thuộc tính thực thể products

4.4 Minh họa các chức năng chính

Quản lý đơn hàng

ĐƠN HÀNG							<input type="text"/> Mã đơn / Mã vận chuyển / Tên / Địa chỉ / Số điện thoại / Ghi chú		<input type="text"/> 99+	<input type="text"/> 13
<div> <input type="button" value="Tất cả"/> <input type="button" value="Tại quầy"/> <input type="button" value="Online"/> <input type="button" value="Thêm lọc"/> </div>										
<div> Tất cả 15976 Mới 4560 <input checked="" type="radio"/> Cần xử lý 494 Chờ xác nhận 1 Chờ hàng 673 Đã đặt hàng 4 Đã xác nhận 1404 >> </div>										
<input type="checkbox"/>	ID	Mã vận đơn	VC	Thẻ	Nguồn QC Webcake	Khách hàng	Trạng thái			
<input type="checkbox"/>	Pancake20904					aaaaaaaaaaszW2	Mới			
<input type="checkbox"/>	Pancake20903	S3808507.MB4-04-D17.14...	GHTK ✓			aaaaaaaaaaszW2	Đang đóng hàng			
<input type="checkbox"/>	Pancake20902	242860511	ShipRocket ✓			Levera Admin	Đã hủy			
<input type="checkbox"/>	Pancake20901					Levera Admin	Mới			
<input type="checkbox"/>	Pancake20900	test	GHTK ✓			Levera Admin	Mới			
<input type="checkbox"/>	Pancake20899			Thêm thẻ		Levera Admin	Mới			
<input type="checkbox"/>	Pancake20898					Nguyễn Phan Phúc	Mới			
<input type="checkbox"/>	AF913110696					Lam Lam	Mới			
<input type="checkbox"/>	AF913110695					Trương Thu Hương	Mới			
<input type="checkbox"/>	Pancake20895	123123	SuperShip ✓				Chờ hàng			
<input type="checkbox"/>	Pancake20832	asdas	SuperShip ✓			Levera Admin	Chờ hàng			
<input type="checkbox"/>	Pancake20831		SuperShip			Levera Admin	Mới			
<input type="checkbox"/>	Pancake20830		SuperShip			Levera Admin	Mới			
<input type="checkbox"/>	Pancake20829	S3808507.HN2-G38.1850...	GHTK ✓	• Đã đổi soát		Levera Admin	Hoàn một phần			
<input type="checkbox"/>	Pancake20828		SuperShip	• Đã đổi soát		Hội Hóng Hớt	Hoàn một phần			
<input type="checkbox"/>	Pancake20827		Aha	• Đã đổi soát		Hội Hóng Hớt	Đã hoàn			
<input type="checkbox"/>	Pancake20826	S3808507.BO.MN1-36-A1...	GHTK ✓	• Delay lấy hàng		harrisonsigideon2	Chờ hàng			
<input type="checkbox"/>	Pancake20825		Aha	• Đã đổi soát		Haha	Đã hoàn			
<input type="checkbox"/>	Pancake20822	PNCKE100248213875319...	Ninja van Phi ✓			hadbad	Đã hủy			
<input type="checkbox"/>	Pancake20824		SuperShip			Phùng Thế Hùng	Mới			
<input type="checkbox"/>	Pancake20823		SuperShip			Phùng Thế Hùng	Mới			
<input type="checkbox"/>	Pancake20821		SuperShip			Phùng Thế Hùng	Chờ hàng			
<input type="checkbox"/>	Pancake20820		SuperShip			Phùng Thế Hùng	Đã xác nhận			
<input type="checkbox"/>	Pancake20819		SuperShip			Phùng Thế Hùng	Đã xác nhận			
<input type="checkbox"/>	Pancake20818	GA43BP3K	GHN ✓			abc	Chờ hàng			
<input type="checkbox"/>	Pancake20817		GHN			Phùng Thế Hùng	Đã xác nhận			
<input type="checkbox"/>	Pancake20816		SuperShip			Phùng Thế Hùng	Mới			
<input type="checkbox"/>	Pancake20815		SuperShip			Khoa 12 12 tv123456	Mới			
<input type="checkbox"/>	Pancake20814	SPXVN026850168267	Shopee Xpress ✓			abc	Chờ hàng			
<input type="checkbox"/>	Pancake20813		SuperShip			Aditya	Mới			
COD: 14.733.551.431 đ Trả trước: 15.464.083.025 đ Cước VC: 187.282.400 đ Phí VC thu khách: 305.621.635 đ <div> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/> <input type="button" value="533"/> <input type="button" value="30 / trang"/> </div>										

Hình 4.11: Giao diện quản lý đơn hàng

Quản lý đơn hàng gồm các đơn hàng có trong cửa hàng, có thể lựa chọn các thông tin quan trọng để hiển thị ra bảng. Phía trên cùng là thanh tìm kiếm, lọc đơn hàng và các nút để thao tác với đơn hàng như: In đơn, cập nhật hàng loạt, xuất excel...

Giao diện tạo đơn hàng

TẠO ĐƠN

Đơn mới +

SẢN PHẨM Zalo Kho Nữ hoàng Anh ...

Sản phẩm Nhập mã, tên sản phẩm hoặc Barcode ☐ Còn hàng (F9)

Giỏ hàng trống

Nhập mã

Thanh toán VND

☐ Miễn phí giao hàng ☐ Chỉ thu phí hoàn

☐ Tính thuế đơn hàng

Phí vận chuyển 30.000 đ

Giảm giá đơn hàng 0 đ

Tiền chuyển khoản 0 đ

Tiền quét thẻ 0 đ

Momo 0 đ

QRPay 0 đ

Tiền khách đưa 0 đ

Phụ thu 0 đ

Điểm thưởng 0

Khách hàng nợ 0 đ

Tiền trả lại khách 0 đ

Tiền chuyển khoản trả khách 0 đ

Tổng số tiền 30.000 đ

Giảm giá 0 đ

Sau giảm giá 30.000 đ

Cần thanh toán 30.000 đ

Đã thanh toán 0 đ

Thông tin Mã tùy chỉnh

Tạo lúc: 03/08/2022 17:10

NV xử lý: Thêm NV xử lý

NV chăm sóc: Thêm NV chăm sóc

Marketer: Thêm NV Marketer

Thêm thẻ

Khách hàng

Online Tại quầy Giải tính: Chưa có

Tên khách hàng... Số điện thoại Ngày sinh

Địa chỉ email

Chưa có thông tin

Nhận hàng Chọn địa chỉ

Dự kiến nhận hàng Chọn ngày

Tên người nhận Số điện thoại

Địa chỉ Quốc gia

Tỉnh thành... Quận huy... Phường xã

Vận chuyển J&T

Kết nối để tự động đẩy đơn sang J&T và cập nhật hành trình vận đơn nhanh hơn [Click vào đây!](#)

Mã vận đơn Phí 0

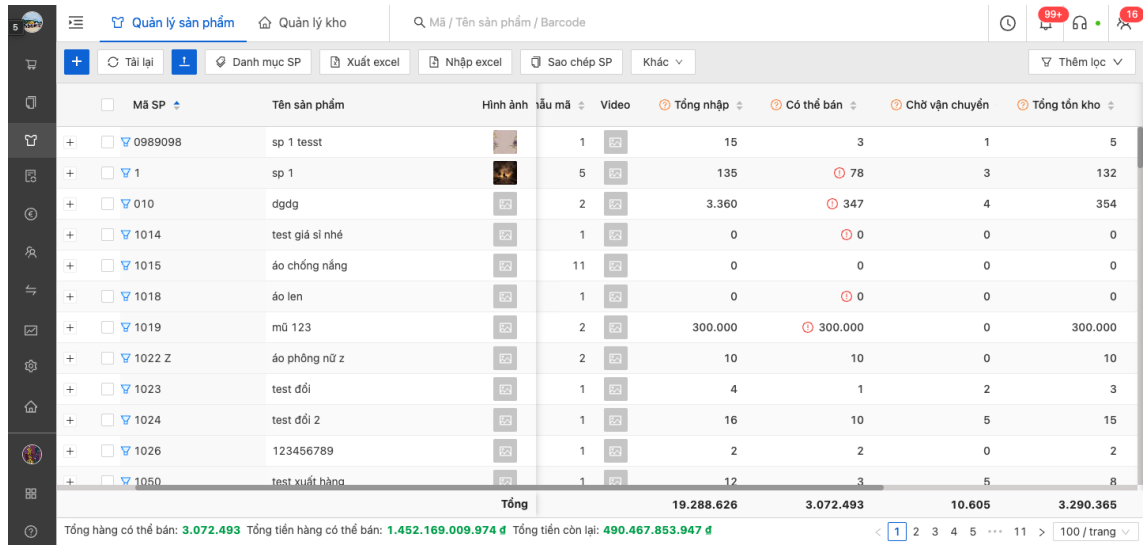
Tổng số tiền: 30.000 đ
COD: 30.000 đ

+ Tạo đơn trùng lặp Mới In (F4) Lưu (F2)

Hình 4.12: Giao diện tạo đơn hàng

Giao diện tạo đơn chia ra làm các thành phần chính là: Thông tin sản phẩm, Thông tin thanh toán, Thông tin khách hàng, Địa chỉ khách hàng, Thông tin đơn vị vận chuyển và hàng nút thông tin cuối cùng để lưu, in đơn hàng.

Quản lý sản phẩm



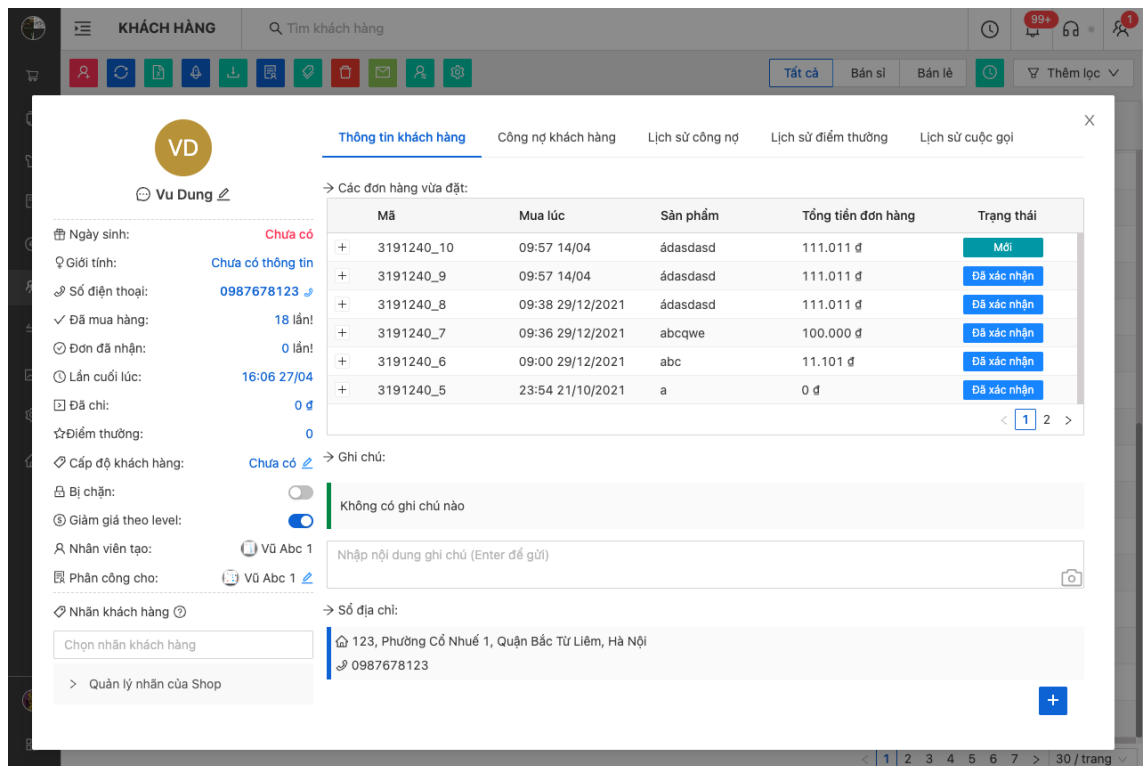
Mã SP	Tên sản phẩm	Hình ảnh	Ấu mã	Video	Tổng nhập	Có thể bán	Chờ vận chuyển	Tổng tồn kho
0989098	sp 1 tesst		1		15	3	1	5
1	sp 1		5		135	78	3	132
010	dgdg		2		3.360	347	4	354
1014	test giá sỉ nhé		1		0	0	0	0
1015	áo chống nắng		11		0	0	0	0
1018	áo len		1		0	0	0	0
1019	mũ 123		2		300.000	300.000	0	300.000
1022 Z	áo phông nữ z		2		10	10	0	10
1023	test đối		1		4	1	2	3
1024	test đối 2		1		16	10	5	15
1026	123456789		1		2	2	0	2
1050	test xuất hàng		1		12	3	5	8
Tổng					19.288.626	3.072.493	10.605	3.290.365

Tổng hàng có thể bán: 3.072.493 Tổng tiền hàng có thể bán: 1.452.169.009.974 đ Tổng tiền còn lại: 490.467.853.947 đ

Hình 4.13: Giao diện quản lý sản phẩm

Quản lý sản phẩm gồm các sản phẩm có trong cửa hàng, có thể lựa chọn các thông tin quan trọng để hiển thị ra bảng. Phía trên cùng là thanh tìm kiếm, lọc sản phẩm và các nút thao tác với sản phẩm. Quản lý danh mục, thẻ sản phẩm, nhóm đơn vị tính. Hiển thị tổng giá trị sản phẩm, tồn kho.

Chi tiết thông tin khách hàng



VD

Vu Dung

Ngày sinh:

Chưa có

Giới tính:

Chưa có thông tin

Số điện thoại:

0987678123

Đã mua hàng:

18 lần!

Đơn đã nhận:

0 lần!

Lần cuối lúc:

16:06 27/04

Đã chi:

0 đ

Điểm thưởng:

0

Cấp độ khách hàng:

Chưa có

Bị chặn:

Không có ghi chú nào

Giảm giá theo level:

Vũ ABC 1

Vũ ABC 1

Nhân viên tạo:

Vũ ABC 1

Phân công cho:

Vũ ABC 1

Nhân khách hàng

Chọn nhân khách hàng

Quản lý nhân của Shop

Thông tin khách hàng

Công nợ khách hàng

Lịch sử công nợ

Lịch sử điểm thưởng

Lịch sử cuộc gọi

Các đơn hàng vừa đặt:

Mã	Mua lúc	Sản phẩm	Tổng tiền đơn hàng	Trạng thái
3191240_10	09:57 14/04	adasdasd	111.011 đ	Mới
3191240_9	09:57 14/04	adasdasd	111.011 đ	Đã xác nhận
3191240_8	09:38 29/12/2021	adasdasd	111.011 đ	Đã xác nhận
3191240_7	09:36 29/12/2021	abcqwe	100.000 đ	Đã xác nhận
3191240_6	09:00 29/12/2021	abc	11.101 đ	Đã xác nhận
3191240_5	23:54 21/10/2021	a	0 đ	Đã xác nhận

Ghi chú:

Không có ghi chú nào

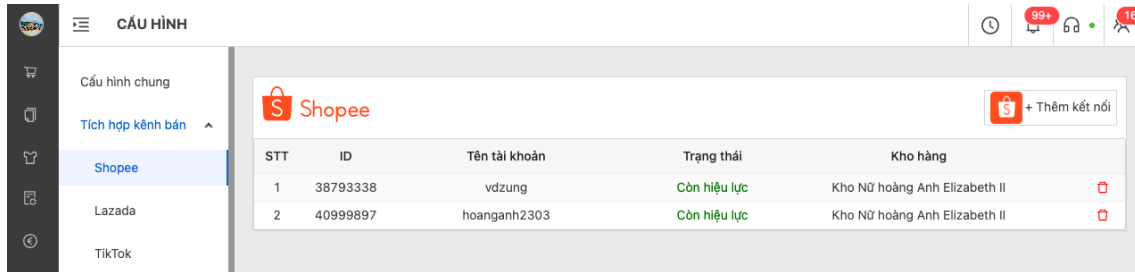
Số địa chỉ:

123, Phường Cổ Nhuế 1, Quận Bắc Từ Liêm, Hà Nội

Hình 4.14: Chi tiết thông tin khách hàng

Giao diện này là chi tiết thông tin của khách mua hàng trong phần quản lý danh sách khách hàng. Gồm các thông tin cơ bản của khách hàng và thông tin quản lý đơn hàng đã mua cũng như công nợ, lịch sử cuộc gọi của khách hàng.

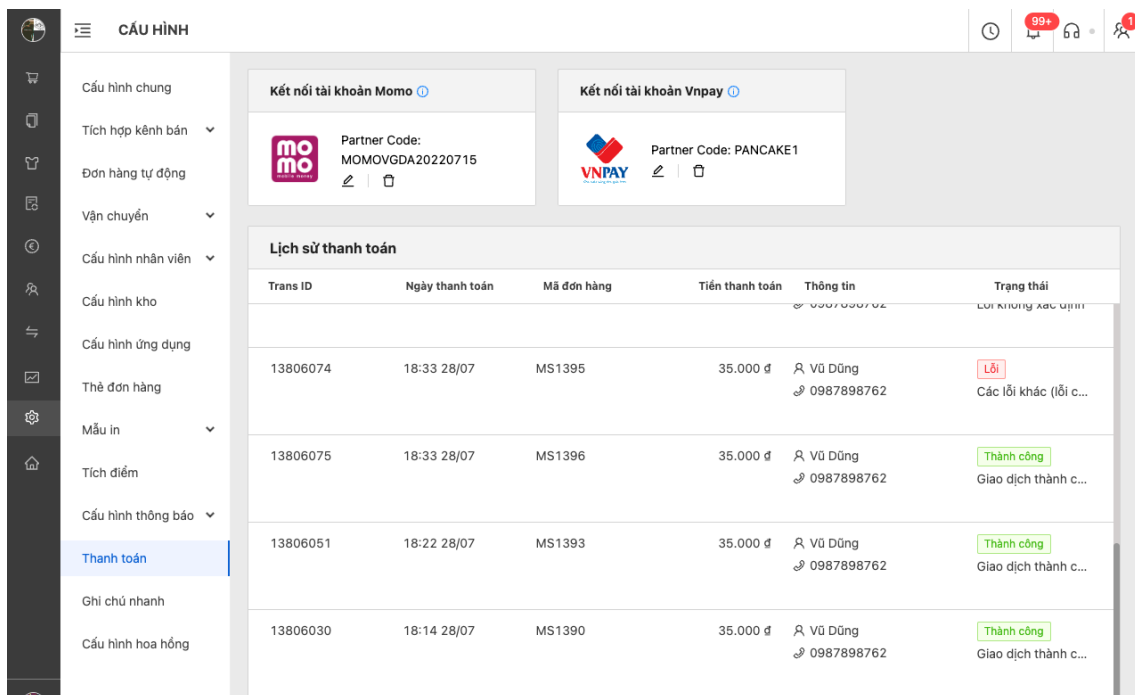
Giao diện tích hợp kênh bán



Hình 4.15: Giao diện tích hợp kênh bán

Người dùng có thể cấu hình các tài khoản từ các sản thương mại điện tử để đồng bộ các thông tin về sản phẩm, đơn hàng về cửa hàng Pancake POS.

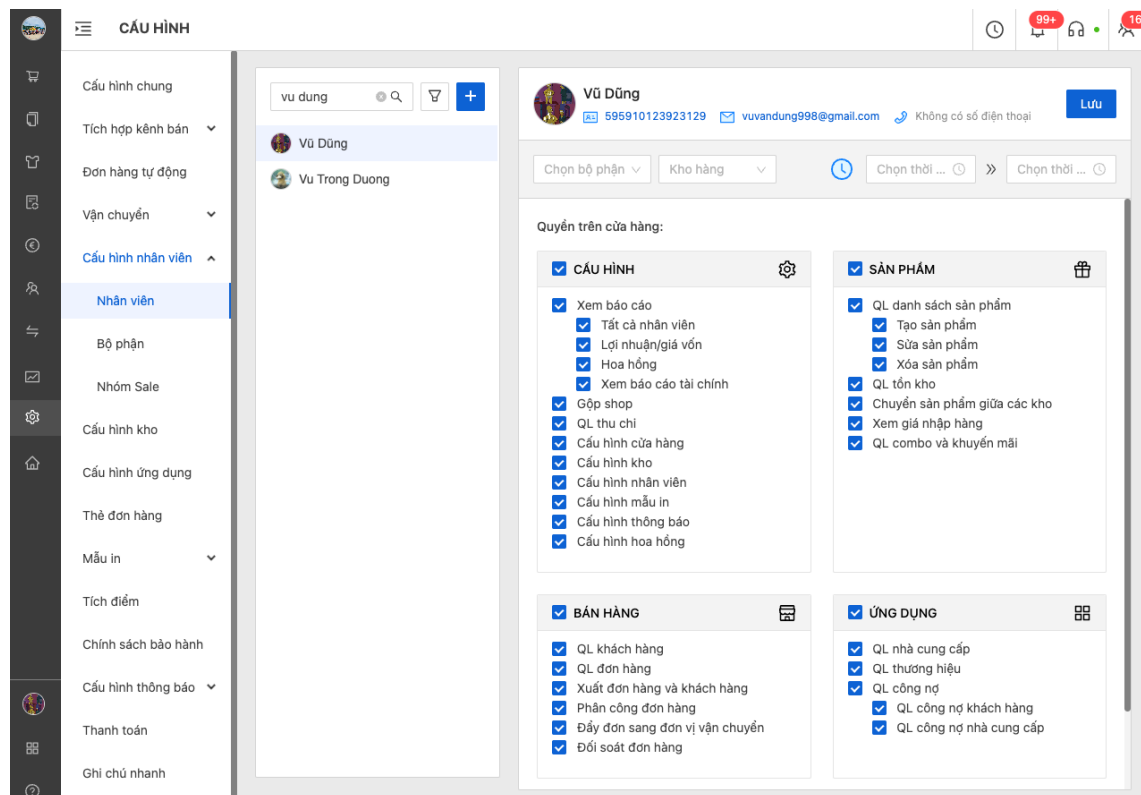
Giao diện tích hợp thanh toán



Hình 4.16: Giao diện tích hợp thanh toán

Ở đây người quản lý cấu hình thông tin tài khoản thanh toán doanh nghiệp của MoMo và VNpay để thực hiện các giao dịch thanh toán trực tuyến. Lịch sử các giao dịch của cửa hàng cũng được thống kê ở đây.

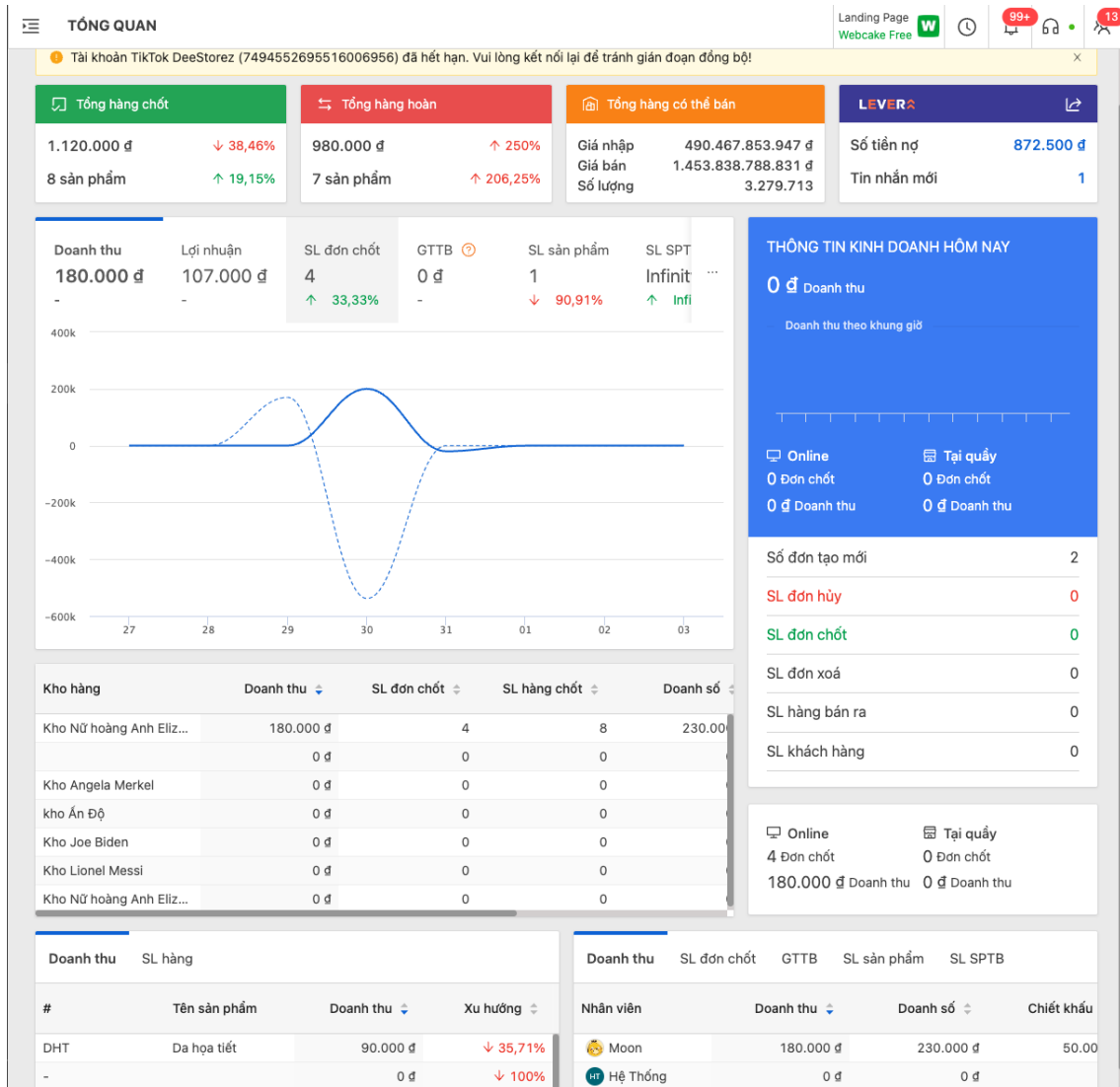
Cấu hình nhân viên



Hình 4.17: Cấu hình nhân viên

Giao diện quản lý nhân viên của cửa hàng, có thể tìm kiếm nhân viên, thêm nhân viên vào cửa hàng. Ngoài ra cho phép phân quyền nhân viên, phân nhân viên vào các bộ phận, kho hàng có trong cửa hàng.

Giao diện thống kê



Hình 4.18: Giao diện thống kê

Giao diện thể hiện tổng quan thống kê của ứng dụng từ doanh thu, lợi nhuận được so sánh theo thời gian có thể lựa chọn, đến thông tin lượng hàng tồn kho, hàng chốt.

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

5.1 Tìm kiếm bằng Elasticsearch

5.1.1 Bài toán

Trong quá trình xây dựng và mở rộng phần mềm quản lý bán hàng, thì đơn hàng sẽ là nội dung chính mà doanh nghiệp chú ý đến. Vì vậy khối lượng công việc và dữ liệu để xử lý về các vấn đề liên quan đến đơn hàng vô cùng lớn. Cụ thể số lượng bản ghi được lưu trong cơ sở dữ liệu hiện tại là hơn 925 triệu bản ghi. Câu hỏi đặt ra là làm sao để cải thiện tốc độ đọc các bản ghi từ cơ sở dữ liệu để đem lại tốc độ tối ưu nhất cho việc quản lý đơn hàng của doanh nghiệp.

Vấn đề gặp phải khi query từ bảng đơn hàng là tốc độ query với những trường được đánh index theo PostgreSQL thì tương đối nhanh nhưng những trường không đánh index thì rất lâu (có thể mất vài chục phút). Mà không thể tạo index với tất cả các trường vì số lượng trường trong bảng đơn hàng ngày càng mở rộng và những vấn đề liên quan khi lấy dữ liệu từ các trường ảo (ví dụ: tính tổng giữa các trường hoặc join vào các bảng liên quan để lấy dữ liệu).

5.1.2 Giải pháp

Giải pháp mà đội ngũ của chúng tôi lựa chọn là Elasticsearch để giải quyết bài toán đề trên. Sau đây là tổng quan về Elasticsearch, cách thức hoạt động mà chúng tôi đã áp dụng vào phần mềm

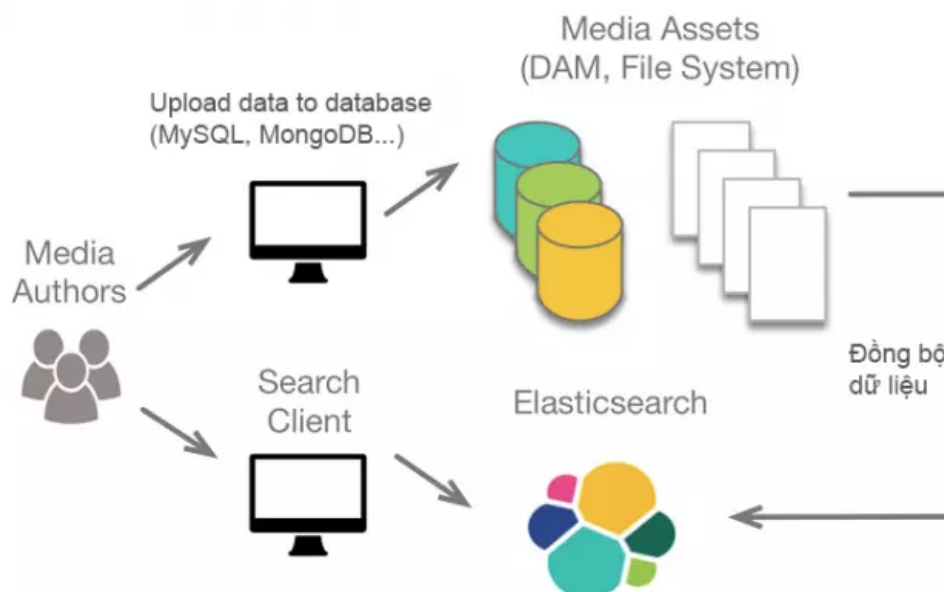
Elasticsearch là gì

Elasticsearch là một công cụ phân tích và tìm kiếm mã nguồn mở, phân tán được xây dựng trên Apache Lucene và được phát triển bằng Java. Nó bắt đầu như một phiên bản có thể mở rộng của khung tìm kiếm nguồn mở Lucene, sau đó thêm khả năng mở rộng các chỉ số Lucene theo chiều ngang.

Elasticsearch cho phép chúng ta lưu trữ, tìm kiếm và phân tích khối lượng dữ liệu khổng lồ một cách nhanh chóng, gần thời gian thực và đưa ra câu trả lời chỉ trong mili giây. Nó có thể đạt được phản hồi tìm kiếm nhanh chóng vì thay vì tìm kiếm văn bản trực tiếp, nó sẽ tìm kiếm một chỉ mục.

Nó sử dụng cấu trúc dựa trên tài liệu thay vì bảng và lược đồ và đi kèm với các API REST mở rộng để lưu trữ và tìm kiếm dữ liệu. Về cốt lõi, bạn có thể coi Elasticsearch như một máy chủ có thể xử lý các yêu cầu JSON và trả lại cho bạn dữ liệu JSON.

Elasticsearch hoạt động như thế nào



Hình 5.1: Cơ chế hoạt động của Elasticsearch

Elasticsearch được xây dựng để hoạt động như một Cloud Server và có tìm kiếm thông qua cơ chế RESTfull. Từ đây lập trình viên tạo ra các HTTP Request gửi đến server Elasticsearch theo dữ liệu dạng JSON. Các dữ liệu này được lưu và đánh theo Index (chỉ mục) giúp tìm kiếm đem lại hiệu quả cao.

Trong đồ án, dữ liệu trong bảng đơn hàng (khoảng 100 trường dữ liệu cơ sở) được phân tích, xử lý và đánh chỉ mục và gọi đến HTTP request để đẩy lên Elasticsearch

Ưu điểm

Tìm kiếm dữ liệu nhanh chóng, mạnh mẽ vì xây dựng trên nền tảng Apache Lucene (near-realtime searching)

Có khả năng mở rộng theo chiều ngang

Tối ưu tìm kiếm. Cho phép tìm kiếm sai chính tả theo sai số nhất định, hỗ trợ nhiều kiểu dữ liệu.

Hỗ trợ việc tìm kiếm bằng việc đặc tả các câu truy vấn phức tạp một cách cụ thể thông qua JSON.

Nhược điểm

Vì elasticsearch được thiết kế chỉ phù hợp cho mục đích tìm kiếm nên về những nhiệm vụ CRUD sẽ kém so với các loại database như MySQL, PostgreSQL,...

Không đảm bảo toàn vẹn dữ liệu. Tức là khi thực hiện các hành động cập nhật

bản ghi hàng loạt mà bị sai logic sẽ dẫn tới mất mát dữ liệu.

Không phù hợp với các hệ thống cập nhật dữ liệu một cách thường xuyên vì việc đánh lại index sẽ rất tốn kém.

5.1.3 Kết quả đạt được

Thống kê dữ liệu đơn hàng một cách nhanh chóng từ việc tính tổng giá trị tất cả đơn hàng, phân loại đơn hàng theo trạng thái của đơn và lọc tất cả các trường có trong đơn hàng nhanh chóng (realtime).

Elasticsearch cho phép tìm kiếm dữ liệu sai chính tả theo sai số nhất định bằng hàm "fuzzy" có khả năng tìm gần đúng theo giá trị đầu vào. Chúng tôi đã áp dụng vào để tìm kiếm các nội dung như ghi chú, thông tin sản phẩm có trong đơn hàng.

Có thể lấy data từ các trường ảo nhanh chóng (đã nêu ở vấn đề bài toán). Thao tác lấy dữ liệu đồng thời nhiều thành phần trong đơn hàng mà ko ảnh hưởng tới hiệu năng của việc tìm kiếm.

5.2 Hiện thị danh sách dữ liệu lớn với React Virtualized

5.2.1 Bài toán

Một trong những vấn đề tiếp theo khi làm việc với phần đơn hàng của phần mềm quản lý là làm thế nào để hiển thị nhiều nhất số lượng đơn hàng trên một trang. Điều này giúp cho khách hàng dễ dàng thao tác mà không cần phải chọn lần lượt các đơn hàng ở các trang khác nhau để thao tác cập nhật hàng loạt.

Bài toán đặt ra ở đây làm sao để hiển thị hàng nghìn đơn hàng, mỗi đơn hàng lại có đến trăm trường dữ liệu mà tối ưu nhất về hiệu năng hiển thị. Như đã nói trong phần công nghệ React sử dụng DOM ảo, mỗi lần component thay đổi trạng thái thì tất cả dữ liệu đơn hàng sẽ đẩy lên để so sánh với DOM xem phần DOM nào cần cập nhật. Vì vậy mỗi lần thao tác làm phần đơn hàng render lại sẽ mất rất nhiều thời gian để hiển thị list đơn hàng này.

5.2.2 Giải pháp

Để giải quyết bài toán trên thì chúng ta cần hiển thị danh sách đơn một cách nhanh nhất những đơn đã được nạp vào DOM, trong khi đó chờ các đơn hàng khác tải lên và render tiếp trên DOM. Cụ thể thay vì render 1000 đơn hàng cùng lúc thì chúng ta sẽ render 30 đơn hàng một lần và các đơn hàng sau sẽ hiển thị lần lượt khi chúng ta thực hiện scroll. Tức là khi scroll đến đâu thì hiển thị đến đó

Một trong những thư viện hiệu quả cho giải pháp nêu trên là React Virtualized.

Virtualized là một kỹ thuật mà chỉ hiển thị những gì có thể nhìn thấy được (thường trong một màn hình). Được sử dụng trong một danh sách có nhiều phần

tử mà có thể thao tác scroll trong danh sách đó và chỉ hiển thị những phần tử được nhìn thấy trong cửa sổ scrollable.

Virtualized là một giải pháp hữu ích để hiển thị danh sách dữ liệu lớn mà đem lại hiệu suất hiển thị cao.

5.3 Quản lý hàng đợi với RabbitMQ

5.3.1 Bài toán

Khi hàng nghìn khách hàng đồng loạt thực hiện thao tác từ ứng dụng thì lượng request tới server rất lớn gây ra tình trạng chậm, quá tải thậm chí mất request.

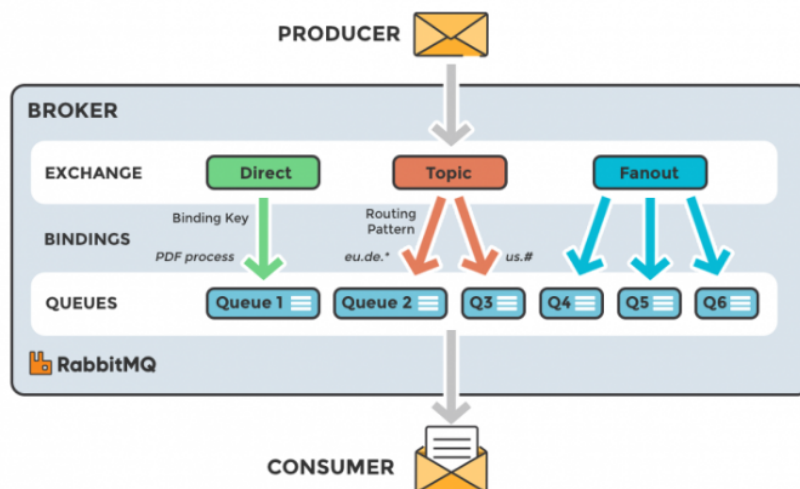
Xử lý các vấn đề về bất đồng bộ cũng như thực hiện song song các tác vụ dưới back-end cũng gặp một số vấn đề gây ra khoảng thời gian đợi lớn cho người dùng. Ví dụ như việc đơn hàng đang được liên kết với sàn thương mại điện tử TikTok shop, dữ liệu từ sàn đổ về, đồng thời người dùng thao tác cập nhật từ phần mềm. Khi này xảy ra vấn đề tổng hợp lại dữ liệu để cập nhật vào cơ sở dữ liệu.

5.3.2 Giải pháp

RabbitMQ là một giải pháp hữu hiệu cho việc thực hiện các thao tác xử lý hàng đợi, thực hiện các công việc đa luồng nhằm đồng bộ hóa dữ liệu một cách song song.

Tìm hiểu về RabbitMQ

RabbitMQ là một AMQP message broker. Là một chương trình trung gian được xây dựng với mục đích liên kết các ứng dụng với nhau, giao tiếp một cách dễ dàng. Sử dụng giao thức AMQP (Advanced Message Queue Protocol) để có thể gửi dữ liệu thông qua cơ chế hàng đợi.



Hình 5.2: Cơ chế hoạt động của RabbitMQ

Tin nhắn không được xuất bản trực tiếp vào hàng đợi, thay vào đó, Producer gửi tin nhắn đến exchange. Exchange chịu trách nhiệm định tuyến các thông điệp đến các hàng đợi khác nhau với sự trợ giúp của các ràng buộc (binding) và khóa định tuyến (routing key). Một ràng buộc là một liên kết giữa một hàng đợi và một trao đổi.

Đầu tiên, Producer xuất bản một tin nhắn cho exchange. Khi tạo exchange, kiểu exchange được chỉ định (Direct, Topic, Fanout) nhằm phân loại tin nhắn.

Tiếp theo, exchange được tin nhắn và bây giờ chịu trách nhiệm định tuyến tin nhắn. Exchange phân loại các thuộc tính tin nhắn khác nhau, ví dụ như khóa định tuyến, tùy thuộc vào loại trao đổi.

Các ràng buộc phải được tạo từ exchange đến hàng đợi. Trao đổi định tuyến thông điệp vào hàng đợi sẽ tùy thuộc vào thuộc tính của tin nhắn.

Cuối cùng, các tin nhắn ở trong hàng đợi cho đến khi chúng được xử lý bởi Consumer.

Trên đây là cơ chế hoạt động của RabbitMQ.

Một số tính năng nổi bật của RabbitMQ

Cho phép các ứng dụng phần mềm kết nối và mở rộng quy mô. Các ứng dụng có thể kết nối với nhau, như các thành phần của một ứng dụng lớn hơn hoặc với các thiết bị và dữ liệu của người dùng. Nhắn tin không đồng bộ, tách ứng dụng bằng cách tách dữ liệu gửi và nhận.

Cung cấp nhiều tính năng cho hiệu suất bằng độ tin cậy, bao gồm tính bền bỉ, xác nhận giao hàng, xác nhận của người phát hành và tính khả dụng cao.

Hỗ trợ các hàng đợi và luồng được nhân rộng. các máy trong một cụm, đảm bảo rằng ngay cả trong trường hợp lỗi phần cứng, các thông báo vẫn an toàn miễn là có phần lớn các nút cụm trực tuyến.

Đối với các máy chủ cần được kết nối lỏng lẻo và không đáng tin cậy hơn mức cho phép phân cụm, RabbitMQ sẽ cung cấp mô hình liên kết.

RabbitMQ đi kèm với giao diện người dùng quản lý để sử dụng cho phép chúng ta theo dõi và kiểm soát mọi khía cạnh của nhà môi giới tin nhắn.

Nếu hệ thống nhắn tin của chúng ta hoạt động sai, RabbitMQ cung cấp hỗ trợ theo dõi để cho phép tìm hiểu điều gì đang xảy ra.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Đóng góp của đề án

Đề án đã trình bày tương đối đầy đủ về quá trình xây dựng và hoàn thiện phần mềm quản lý bán hàng, tác giả đã trình bày những khảo sát về thực trạng của phần mềm quản lý bán hàng, từ đó đưa ra tổng quan các chức năng và thiết kế kiến trúc mà phần mềm cần có, bao gồm:

Các Use case đặc tả chức năng chính của phần mềm

Công nghệ nền tảng được sử dụng

Cách xây dựng và lựa chọn kiến trúc hệ thống phần mềm

6.2 Hướng phát triển

Mặc dù hệ thống đã đạt được mục tiêu ban đầu nhưng hệ thống còn tồn đọng nhiều vấn đề về hiệu năng cũng như các tính năng mới bổ sung để hoàn thiện phần mềm. Một số tính năng bổ sung thêm trong tương lai

Tích hợp thêm các phương thức thanh toán hấp dẫn để kéo thêm khách mua hàng như phương thức trả góp.

Đồng bộ thêm các sàn thương mại điện tử lớn để mở rộng tệp khách hàng như Woocommerce, Tokopedia.

Cải thiện lại giao diện người dùng để phù hợp, tối ưu cho các người dùng mới.

6.3 Kết luận

Việc triển khai xây dựng ứng dụng quản lý bán hàng bước đầu đã đạt được những thành quả nhất định. Ứng dụng đã được xây dựng thành công, đáp ứng được nhu cầu của bài toán (hỗ trợ các nhà quản lý có thể quản lý thông tin của doanh nghiệp một cách dễ dàng).

Việc ứng dụng công nghệ vào trong lĩnh vực quản lý giúp cho công việc trở nên đơn giản và trực quan hơn rất nhiều so với cách làm truyền thống trước đây từ đó giúp công ty tiết kiệm chi phí, nâng cao năng suất.

Ứng dụng cơ bản đã đáp ứng các yêu cầu của khách hàng và có những cải tiến so với ứng dụng tiền nhiệm là khác về mặt hiệu năng và giao diện.

TÀI LIỆU THAM KHẢO

[1] Chinmayee Deshpand, *The Best Guide to Know What Is React*. [Online]. Available: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs> (visited on 06/30/2022).

[2] Gints Dreimanis, *Introduction to Elixir*. [Online]. Available: <https://serokell.io/blog/introduction-to-elixir> (visited on 06/30/2022).

[3] NAM VU, *RabbitMQ – Một message queue phổ biến mà dev nào cũng nên biết*. [Online]. Available: <https://blog.ntechdevelopers.com/rabbitmq-mot-message-queue-pho-bien-ma-dev-nao-cung-nen-biet/> (visited on 07/05/2022).

[4] Matthew Martin, *MVC Framework Tutorial for Beginners What is, Architecture & Example*. [Online]. Available: <https://www.guru99.com/mvc-tutorial.html> (visited on 06/25/2022).

[5] Ralf Abueg, *Elasticsearch: What It Is, How It Works, And What It's Used For*. [Online]. Available: <https://www.knowi.com/blog/what-is-elastic-search/> (visited on 07/05/2022).