

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐỒ ÁN TỐT NGHIỆP

PHÁT TRIỂN ỨNG DỤNG CHO CHUỖI CỦA HÀNG BÁN HOA QUẢ

SINH VIÊN THỰC HIỆN	: ĐỖ HOÀNG LINH
MÃ SINH VIÊN	: 14510201035
KHOA	: CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐỖ HOÀNG LINH

**PHÁT TRIỂN ỨNG DỤNG CHO CHUỖI CỦA
HÀNG BÁN HOA QUẢ**

CHUYÊN NGÀNH	: CÔNG NGHỆ THÔNG TIN
MÃ SỐ	: 74.80.201
NGƯỜI HƯỚNG DẪN	: TS. TRẦN QUÝ NAM

HÀ NỘI - 2024

LỜI CAM ĐOAN

Em là Đỗ Hoàng Linh, sinh viên của trường Đại Học Đại Nam, cam kết rằng Báo cáo Đồ án Tốt nghiệp có đề tài là “PHÁT TRIỂN ỨNG DỤNG CHO CHUỖI CỬA HÀNG BÁN HOA QUẢ” được viết bởi chính em dựa trên kinh nghiệm và kiến thức thực tế trong quá trình học tập.

Em xin cam kết báo cáo đồ án này được hoàn thành dựa trên các kết quả đồ án của em và các kết quả nghiên cứu này chưa được dùng cho bất cứ Báo cáo (báo cáo, khóa luận tốt nghiệp) cùng cấp nào khác.

Nếu có bất kỳ phát hiện về vi phạm hoặc không trung thực, em sẽ chịu trách nhiệm pháp lý và chấp nhận các hình thức xử lý mà **Đại Học Đại Nam** trường quy định.

Em cam đoan rằng các thông tin và kết quả được trình bày trong báo cáo là trung thực và xác thực.

Hà nội, ngày...tháng... năm 2024

Sinh viên

(ký và ghi rõ họ và tên)

LỜI MỞ ĐẦU

Trong bối cảnh phát triển mạnh mẽ của ngành công nghiệp thực phẩm và nhu cầu tiêu dùng các sản phẩm tươi sống ngày càng tăng cao, việc quản lý hiệu quả chuỗi cửa hàng hoa quả trở thành một thách thức lớn và cũng là cơ hội để nâng cao năng suất, hiệu quả kinh doanh. Các chuỗi cửa hàng hoa quả không chỉ cần đảm bảo chất lượng sản phẩm mà còn phải tối ưu hóa quy trình vận hành từ quản lý hàng tồn kho, theo dõi đơn hàng, đến dịch vụ khách hàng và chăm sóc khách hàng. Điều này đòi hỏi sự hỗ trợ từ các giải pháp công nghệ hiện đại và phù hợp.

Với sự phát triển không ngừng của công nghệ thông tin, các giải pháp phần mềm quản lý đã trở thành công cụ đắc lực, giúp các doanh nghiệp tiết kiệm thời gian, giảm thiểu chi phí và nâng cao hiệu quả kinh doanh. Các hệ thống quản lý thông minh không chỉ cung cấp cái nhìn tổng quan về hoạt động kinh doanh mà còn hỗ trợ ra quyết định dựa trên dữ liệu chính xác và kịp thời. Trong bối cảnh này, việc xây dựng một ứng dụng quản lý chuỗi cửa hàng hoa quả không chỉ là một nhiệm vụ cấp thiết mà còn có ý nghĩa thực tiễn cao, đáp ứng nhu cầu quản lý hiệu quả và chuyên nghiệp của các doanh nghiệp.

Đồ án tốt nghiệp của em tập trung vào việc phát triển một ứng dụng quản lý chuỗi cửa hàng hoa quả với mục tiêu giúp các doanh nghiệp trong lĩnh vực này tối ưu hóa hoạt động kinh doanh, nâng cao chất lượng dịch vụ và tăng cường khả năng cạnh tranh. Ứng dụng này sẽ cung cấp các chức năng quan trọng như quản lý hàng hóa, theo dõi tình trạng đơn hàng, phân tích dữ liệu bán hàng, quản lý khách hàng và báo cáo tổng hợp. Bên cạnh đó, ứng dụng còn tích hợp các tính năng tự động hoá nhằm giảm thiểu công việc thủ công, giúp nhân viên tập trung vào các nhiệm vụ có giá trị cao hơn.

Trong quá trình thực hiện đồ án, em đã áp dụng các phương pháp nghiên cứu khoa học và kỹ thuật hiện đại, từ việc khảo sát nhu cầu thực tế của các cửa hàng hoa quả, phân tích yêu cầu người dùng, thiết kế giao diện người dùng thân thiện, cho đến triển khai và kiểm thử hệ thống. Chúng em cũng đã tham khảo các mô hình quản lý tiên tiến, học hỏi từ những kinh nghiệm thực tế của các doanh nghiệp trong và ngoài nước để đảm bảo rằng sản phẩm của mình đáp ứng được các tiêu chuẩn chất lượng cao nhất.

Em tin rằng, với những tính năng hữu ích và giao diện thân thiện, ứng dụng này sẽ mang lại nhiều giá trị cho người sử dụng, giúp họ nắm bắt và kiểm soát tốt hơn mọi khía cạnh của hoạt động kinh doanh. Qua đó, các cửa hàng hoa quả có thể nâng cao chất lượng dịch vụ, đáp ứng nhu cầu ngày càng cao của thị trường và duy trì sự cạnh tranh trong ngành. Em cũng kỳ vọng rằng, ứng dụng này sẽ mở ra những hướng đi mới, những cơ hội mới trong việc ứng dụng công nghệ vào quản lý chuỗi cung ứng thực phẩm tươi sống.

LỜI CẢM ƠN

Được sự phân công của khoa **Công Nghệ Thông Tin Trường Đại Học Đại Nam**, và dưới sự hướng dẫn của thầy **TS. Trần Quý Nam**, em đã hoàn thành đề tài là **“PHÁT TRIỂN ỨNG DỤNG CHO CHUỖI CỬA HÀNG BÁN HOA QUẢ”**. Để hoàn thành đồ án này em xin chân thành cảm ơn thầy đã tận tình hướng dẫn em trong suốt quá trình làm đồ án tốt nghiệp. Với sự chỉ bảo của thầy, em đã có những định hướng trong công việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án tốt nghiệp.

Em nhận ra rằng bài viết có thể còn thiếu sót và mong nhận được sự góp ý chân thành từ các thầy cô để em có thể khắc phục.

Em xin chân thành cảm ơn!

NHẬN XÉT

This image shows a full page of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for handwriting practice or general writing. There are no margins, text, or other markings on the page.

MỤC LỤC

LỜI CAM ĐOAN	1
LỜI MỞ ĐẦU	1
LỜI CẢM ƠN	1
NHẬN XÉT.....	1
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	1
1.1. Lí do chọn đề tài.	1
1.2. Mục tiêu nghiên cứu.....	2
1.3. Đối tượng nghiên cứu.....	3
1.4. Phương pháp nghiên cứu.....	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	7
2.1. Lý do chọn Vue.js và Asp.Net	7
2.2. VueJS.....	8
2.2.1. VueJS là gì?	8
2.2.2. Ưu và nhược điểm	11
2.2.3. Ứng dụng của VueJS	12
2.3. Typescript	13
2.3.1. Interfaces	15
2.3.2. Classes	16
2.3.3. Modules.....	17
2.3.4. Ưu điểm của Typescripts	18
2.4. API (Application Programming Interface)	19
2.4.1. API là gì?	19
2.4.2. Ứng dụng API	22

2.4.3. Web API là gì?	22
2.4.4. Web API hoạt động như nào?	23
2.4.5. Ưu và nhược điểm của Web API.....	23
2.5. Asp.Net framework.....	24
2.6. Thanh toán vn pay	24
2.7. Google map	27
2.7.1 Google cloud	27
2.7.2. Google map.....	30
2.7.3. Mối liên hệ giữa Google map và Google Cloud.....	32
CHƯƠNG 3: YÊU CẦU VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG	34
3.1. Phân tích hệ thống.....	34
3.2. Các chức năng chính.....	34
3.3.Chức năng dành cho khách hàng.	34
3.4.Các chức năng cho admin	35
3.5.Mô hình phân cấp chức năng.....	35
3.6.2. Biểu đồ tuần tự.....	42
3.6.3. Biểu đồ hoạt động	46
3.7.Thiết kế cơ sở dữ liệu	49
3.7.1.Thiết kế bảng DATABASE.....	49
3.7.2 Sơ đồ DIAGRAMS	55
CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC	56
4.1. Trang đăng nhập.....	56
4.2. Trang đăng ký	57
4.3. Trang chủ.....	58
4.4. Trang sản phẩm	60
4.5. Khu vực chi nhánh cửa hàng	62

4.6. Danh mục sản phẩm	64
4.7. Trang Dashboard	66
4.8. Trang addmin quản lý cửa hàng	67
4.9. Trang addmin quản lý hoa quả	68
4.10. Trang quản lý các loại	70
KẾT LUẬN	72
TÀI LIỆU THAM KHẢO	74

DANH MỤC VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	AI	Artificial Intelligence
2	CRUD	Create, Read, Update, Delete
3	HTML	Hypertext Markup Language
4	CSS	Cascading Style Sheets
5	API	Application Programming Interface
6	DOM	Document Object Model
7	UI/UX	User Interface/User Experience
8	MVC	Model-View-Controller
9	JSON/XML	JavaScript Object Notation / eXtensible Markup Language
10	HTTP/HTTPS	Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure

DANH MỤC BẢNG BIỂU

Bảng 3. 1	Bill_Details.....	49
Bảng 3. 2	EFMigrationsHistory.....	49
Bảng 3. 3	Bills.....	50
Bảng 3. 4	Carts.....	51
Bảng 3. 5	Categories.....	51
Bảng 3. 6	Fruits.....	52
Bảng 3. 7	RefreshTokens.....	53
Bảng 3. 8	Stores.....	53
Bảng 3. 9	User.....	54

DANH MỤC HÌNH ẢNH

Hình 2. 1 Web API.....	21
Hình 3. 1Usecase tổng quát.....	37
Hình 3. 2 Usecase Login	38
Hình 3. 3 Usecase quản lý cửa hàng.....	38
Hình 3. 4 Usecase Quản lý hoá đơn.....	39
Hình 3. 5 Usecase quản lý giỏ hàng.....	39
Hình 3. 6 Usecase quản lý loại sản phẩm	40
Hình 3. 7 Usecase quản lý người dùng.....	40
Hình 3. 8 Usecase quản lý sản phẩm.....	41
Hình 3. 9 Usecase tìm kiếm.....	41
Hình 3. 10 Usecase đăng ký	42
Hình 3. 11 Biểu đồ tuần tự đăng nhập.....	42
Hình 3. 12 Biểu đồ tuần tự đăng ký	43
Hình 3. 13 Biểu đồ tuần tự tìm kiếm.....	43
Hình 3. 14 Biểu đồ tuần tự thêm sản phẩm.....	44
Hình 3. 15 Biểu đồ tuần tự sửa sản phẩm	44
Hình 3. 16 Biểu đồ tuần tự xoá sản phẩm	45
Hình 3. 17 Biểu đồ tuần tự giỏ hàng	45
Hình 3. 18 Biểu đồ tuần tự giỏ hàng	46
Hình 3. 19 Biểu đồ hoạt động Login	46
Hình 3. 20 Biểu đồ hoạt động thêm dữ liệu.....	47
Hình 3. 21 Biểu đồ hoạt động sửa dữ liệu.....	47
Hình 3. 22 Biểu đồ hoạt động xoá dữ liệu.....	48

Hình 3. 23 Biểu đồ hoạt động đặt hàng	48
Hình 3. 24 Sơ đồ Diagram.....	55
Hình 4. 1 Trang đăng nhập.....	56
Hình 4. 2 Trang đăng ký	57
Hình 4. 3 Trang chủ.....	58
Hình 4. 4 Trang sản phẩm	60
Hình 4. 5 Mục chi nhánh cửa hàng	62
Hình 4. 6 Trang danh mục sản phẩm	64
Hình 4. 7 Trang Dashboard	66
Hình 4. 8 Trang addmin quản lý cửa hàng	68
Hình 4. 9 Addmin quản lý hoa quả	69
Hình 4. 10 Trang quản lý các loại	70

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1. Lí do chọn đề tài

Nhu cầu tiêu thụ hoa quả sạch và an toàn đang ngày càng gia tăng trong cộng đồng. Với các vấn đề về an toàn thực phẩm ngày càng phức tạp và các vụ bê bối liên quan đến thực phẩm bẩn xuất hiện thường xuyên, người tiêu dùng trở nên cẩn trọng hơn trong việc chọn lựa sản phẩm cho gia đình. Một ứng dụng hiện đại, tiện lợi sẽ giúp khách hàng dễ dàng tiếp cận, đặt hàng và nhận hàng tại nhà mà không cần tốn nhiều thời gian và công sức, đồng thời đảm bảo tính an toàn và chất lượng của sản phẩm.

Ngoài ra, từ góc độ doanh nghiệp, việc xây dựng và quản lý chuỗi cửa hàng bán hoa quả thông qua ứng dụng số hóa mang lại nhiều lợi ích vượt trội. Ứng dụng này sẽ giúp các doanh nghiệp tối ưu hóa quy trình vận hành, từ việc quản lý kho hàng, theo dõi lượng hàng tồn kho, cho đến việc quản lý đơn hàng và giao hàng. Điều này không chỉ giúp giảm thiểu chi phí vận hành mà còn nâng cao hiệu suất làm việc và tăng cường sự hài lòng của khách hàng nhờ dịch vụ chăm sóc khách hàng được cải thiện.

Hơn nữa, phát triển ứng dụng chuỗi cửa hàng bán hoa quả còn thể hiện sự kết hợp hoàn hảo giữa công nghệ thông tin và quản lý kinh doanh, hai lĩnh vực đang phát triển mạnh mẽ và có tính ứng dụng cao trong cuộc sống hiện đại. Đây là một lĩnh vực có nhiều tiềm năng phát triển và đang được nhiều doanh nghiệp quan tâm đầu tư. Việc nghiên cứu và triển khai đề tài này sẽ không chỉ giúp sinh viên nắm vững kiến thức chuyên môn mà còn có cơ hội áp dụng vào thực tiễn, giải quyết các vấn đề cụ thể của ngành bán lẻ.

Cuối cùng, sự bùng nổ của công nghệ di động và xu hướng mua sắm trực tuyến đã thay đổi cách thức tiêu dùng của người dân. Một ứng dụng di động cho chuỗi cửa hàng bán hoa quả không chỉ đáp ứng nhu cầu mua sắm tiện lợi của khách hàng mà còn mở ra nhiều cơ hội kinh doanh mới cho doanh nghiệp, giúp họ tiếp cận được với một lượng lớn khách hàng tiềm năng mà không bị giới hạn bởi khoảng cách địa lý. Chính vì những lý do trên, việc lựa chọn đề tài phát triển ứng dụng chuỗi cửa hàng bán hoa quả là một sự lựa chọn đúng đắn và có ý nghĩa thực tiễn cao cho đồ án tốt nghiệp.

1.2. Mục tiêu nghiên cứu

Nâng cao trải nghiệm mua sắm của khách hàng: Phát triển một ứng dụng thân thiện, dễ sử dụng, giúp khách hàng có thể dễ dàng tìm kiếm, chọn lựa và đặt mua hoa quả từ chuỗi cửa hàng. Ứng dụng cần tích hợp các tính năng như gợi ý sản phẩm, đánh giá và nhận xét từ người dùng, và chức năng thanh toán trực tuyến an toàn.

Tối ưu hóa quy trình quản lý của doanh nghiệp: Xây dựng hệ thống quản lý thông minh, giúp doanh nghiệp theo dõi và quản lý hàng tồn kho, đơn hàng, và chuỗi cung ứng một cách hiệu quả. Điều này bao gồm việc phát triển các công cụ để phân tích dữ liệu bán hàng, dự báo nhu cầu và tối ưu hóa quy trình vận hành.

Cải thiện dịch vụ chăm sóc khách hàng: Tạo ra các tính năng hỗ trợ khách hàng như dịch vụ tư vấn trực tuyến, theo dõi đơn hàng, và chính sách đổi trả hàng linh hoạt. Mục tiêu là tạo ra một môi trường mua sắm an tâm và thuận tiện cho khách hàng.

Tăng cường khả năng cạnh tranh của chuỗi cửa hàng: Sử dụng công nghệ để nâng cao khả năng cạnh tranh của doanh nghiệp trên thị trường. Điều này bao gồm việc tích hợp các chương trình khuyến mãi, thẻ thành viên, và các chiến dịch marketing thông qua ứng dụng để thu hút và giữ chân khách hàng.

Nghiên cứu và áp dụng các công nghệ tiên tiến: Khám phá và ứng dụng các công nghệ mới như trí tuệ nhân tạo (AI), học máy (Machine Learning), và phân tích dữ liệu lớn (Big Data) vào việc quản lý và phát triển chuỗi cửa hàng. Mục tiêu là tạo ra một hệ thống thông minh và linh hoạt, có khả năng tự động hóa nhiều quy trình và cải thiện hiệu quả hoạt động.

Đánh giá hiệu quả và tác động của ứng dụng: Thiết lập các tiêu chí để đánh giá hiệu quả của ứng dụng sau khi triển khai, bao gồm các chỉ số về mức độ hài lòng của khách hàng, tăng trưởng doanh thu, và hiệu quả quản lý. Mục tiêu là không chỉ phát triển một ứng dụng hiệu quả mà còn đảm bảo rằng ứng dụng này có thể mang lại lợi ích lâu dài cho doanh nghiệp..

1.3. Đối tượng nghiên cứu

Khách hàng là đối tượng chính, bao gồm cả người tiêu dùng hiện tại và khách hàng tiềm năng. Việc nghiên cứu sâu về nhu cầu, sở thích, thói quen mua sắm và các yếu tố ảnh hưởng đến quyết định mua hoa quả của họ là điều cần thiết. Điều này không chỉ giúp định hình các tính năng và dịch vụ của ứng dụng mà còn đảm bảo rằng ứng dụng đáp ứng đúng mong đợi của khách hàng, từ đó nâng cao trải nghiệm người dùng và tăng cường mức độ hài lòng của khách hàng.

Tiếp theo, chuỗi cửa hàng bán hoa quả và nhân viên của họ cũng là đối tượng quan trọng. Phân tích cấu trúc tổ chức, quy trình kinh doanh và chiến lược quản lý hiện tại của các cửa hàng sẽ giúp xác định những điểm mạnh, điểm yếu, cơ hội và thách thức mà doanh nghiệp đang đối mặt. Từ đó, có thể đề xuất các giải pháp công nghệ nhằm cải thiện hiệu quả hoạt động, tối ưu hóa quy trình vận hành và nâng cao năng suất lao động. Nhân viên cửa hàng cũng được xem xét kỹ lưỡng, từ vai trò, nhiệm vụ đến các thách thức trong công việc, để phát triển các công cụ hỗ trợ công việc và nâng cao hiệu suất làm việc của họ. Công nghệ và hệ thống thông tin là yếu tố then chốt trong việc phát triển ứng dụng. Điều này bao gồm việc đánh giá các hệ thống quản lý hiện tại, như quản lý kho hàng, đơn hàng và chuỗi cung ứng, để xác định các hạn chế và đề xuất giải pháp công nghệ nhằm cải thiện hiệu quả quản lý. Đồng thời, nghiên cứu và áp dụng các công nghệ mới như trí tuệ nhân tạo, học máy và phân tích dữ liệu lớn vào việc phát triển ứng dụng sẽ mang lại nhiều lợi ích vượt trội cho doanh nghiệp. Các công nghệ tiên tiến này có thể giúp tự động hóa nhiều quy trình, tối ưu hóa quản lý dữ liệu và cung cấp các dự đoán chính xác về nhu cầu thị trường.

Thị trường và đối thủ cạnh tranh cũng là những đối tượng nghiên cứu không thể thiếu. Phân tích thị trường bán hoa quả hiện nay, bao gồm xu hướng tiêu dùng, nhu cầu thị trường và các yếu tố ảnh hưởng, sẽ giúp xác định vị trí và chiến lược cạnh tranh của chuỗi cửa hàng. Đồng thời, phân tích đối thủ cạnh tranh trực tiếp và gián tiếp, bao gồm các ứng dụng bán lẻ khác, sẽ giúp rút ra những bài học quý giá và đề xuất các cải tiến cho ứng dụng. Việc nghiên cứu các điểm mạnh, điểm yếu và chiến lược kinh doanh của đối thủ sẽ cung cấp những thông tin quan trọng để xây dựng một ứng dụng cạnh tranh và độc đáo.

Cuối cùng, khía cạnh kinh tế và xã hội cũng được xem xét kỹ lưỡng. Đánh giá tác động kinh tế của việc triển khai ứng dụng đối với chuỗi cửa hàng và thị trường bán hoa quả bao gồm các yếu tố như chi phí triển khai, lợi nhuận kỳ vọng và khả năng mở rộng kinh doanh. Đồng thời, nghiên cứu tác động xã hội của ứng dụng, như việc tạo ra việc làm, nâng cao chất lượng cuộc sống của người tiêu dùng và các tác động tích cực đến cộng đồng, sẽ đảm bảo rằng ứng dụng không chỉ mang lại lợi ích kinh tế mà còn đóng góp vào sự phát triển bền vững của xã hội. Bằng cách nghiên cứu toàn diện các đối tượng này, đồ án tốt nghiệp sẽ cung cấp một cái nhìn sâu sắc và đầy đủ, từ đó đề xuất các giải pháp hiệu quả và khả thi cho việc phát triển ứng dụng chuỗi cửa hàng bán hoa quả.

1.4. Phương pháp nghiên cứu

Khảo sát và phỏng vấn:

Khảo sát khách hàng: Thiết kế bảng khảo sát trực tuyến để thu thập ý kiến từ khách hàng về nhu cầu, sở thích và thói quen mua sắm. Các câu hỏi nên bao gồm cả định lượng và định tính để có cái nhìn toàn diện.

Phỏng vấn sâu: Thực hiện các buổi phỏng vấn sâu với một nhóm khách hàng đại diện, nhà quản lý và nhân viên cửa hàng để hiểu rõ hơn về yêu cầu và kỳ vọng cụ thể.

- Nghiên cứu tài liệu và phân tích thị trường:

Nghiên cứu tài liệu: Thu thập và phân tích các báo cáo, tài liệu khoa học và bài viết liên quan đến thị trường bán hoa quả, xu hướng tiêu dùng và công nghệ ứng dụng trong ngành bán lẻ.

Phân tích thị trường: Sử dụng các công cụ phân tích thị trường để đánh giá tình hình hiện tại và dự báo xu hướng phát triển của thị trường bán hoa quả.

- Phân tích đối thủ cạnh tranh:

Nghiên cứu đối thủ cạnh tranh: Thu thập thông tin về các đối thủ cạnh tranh trực tiếp và gián tiếp, phân tích các điểm mạnh, điểm yếu và chiến lược kinh doanh của họ.

- Phân tích và thiết kế hệ thống:

Phân tích yêu cầu hệ thống: Xác định các yêu cầu chức năng và phi chức năng của ứng dụng. Xây dựng tài liệu yêu cầu hệ thống chi tiết dựa trên thông tin từ khảo sát, phỏng vấn và nghiên cứu tài liệu.

Thiết kế hệ thống: Phát triển các mô hình thiết kế hệ thống như sơ đồ chức năng, sơ đồ ERD và mô hình UML. Thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) bằng các công cụ như Figma hoặc Adobe XD.

- Phát triển ứng dụng với Vue.js và ASP.NET:

Cấu trúc dự án: Thiết lập cấu trúc dự án với Vue.js cho phần frontend và ASP.NET cho phần backend. Sử dụng Vue CLI để tạo dự án Vue.js và ASP.NET Core để phát triển backend.

Phát triển frontend: Sử dụng Vue.js để xây dựng giao diện người dùng tương tác, bao gồm các thành phần như trang chủ, trang sản phẩm, giỏ hàng và trang thanh toán. Áp dụng Vue Router cho điều hướng và Vuex để quản lý trạng thái ứng dụng.

Phát triển backend: Sử dụng ASP.NET Core để xây dựng API RESTful, xử lý yêu cầu từ frontend và tương tác với cơ sở dữ liệu. Thiết kế các dịch vụ để quản lý dữ liệu như sản phẩm, đơn hàng và khách hàng. Sử dụng Entity Framework Core để làm việc với cơ sở dữ liệu.

Tích hợp frontend và backend: Sử dụng Axios hoặc Fetch API để gọi API từ Vue.js, đảm bảo giao tiếp hiệu quả giữa frontend và backend. Thiết lập cơ chế xác thực và phân quyền người dùng để bảo vệ các tài nguyên.

- Kiểm thử ứng dụng:

Kiểm thử đơn vị (Unit Testing): Viết các test case cho từng thành phần của ứng dụng, sử dụng các công cụ như Jest cho Vue.js và xUnit hoặc NUnit cho ASP.NET.

Kiểm thử tích hợp (Integration Testing): Đảm bảo các thành phần của hệ thống hoạt động chính xác khi kết hợp với nhau. Sử dụng Postman hoặc Swagger để kiểm thử API.

Kiểm thử hệ thống (System Testing): Đảm bảo toàn bộ hệ thống hoạt động đúng chức năng và đáp ứng yêu cầu. Mô phỏng các kịch bản sử dụng thực tế để kiểm tra tính năng và hiệu suất của ứng dụng.

Kiểm thử chấp nhận người dùng (User Acceptance Testing): Mời người dùng thực tế trải nghiệm ứng dụng và đưa ra phản hồi. Dựa trên phản hồi để thực hiện các cải tiến cần thiết.

- Triển khai và đánh giá:

Triển khai ứng dụng: Đưa ứng dụng lên các nền tảng di động và web. Sử dụng các dịch vụ như Azure, AWS hoặc Heroku để triển khai backend ASP.NET và frontend Vue.js.

Đánh giá hiệu quả và cải tiến: Thu thập phản hồi từ người dùng sau khi triển khai ứng dụng. Đánh giá hiệu quả và tác động của ứng dụng thông qua các chỉ số như mức độ hài lòng của khách hàng, tăng trưởng doanh thu và hiệu quả quản lý. Thực hiện các cải tiến dựa trên phản hồi để nâng cao chất lượng ứng dụng.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Lý do chọn Vue.js và Asp.Net

Việc chọn Vue.js và ASP.NET cho phát triển ứng dụng chuỗi cửa hàng bán hoa quả trong dự án tốt nghiệp là một quyết định chiến lược, đảm bảo sự hiệu quả và thành công của dự án. Vue.js, là một framework JavaScript nổi bật với tính nhẹ nhàng và tốc độ nhanh chóng, giúp tối ưu hóa hiệu suất và cải thiện trải nghiệm người dùng nhờ vào virtual DOM. Điều này giúp trang web phản hồi nhanh và mượt mà, giảm thiểu thời gian chờ đợi của người dùng khi truy cập vào ứng dụng. Với cú pháp đơn giản, dễ hiểu và dễ học, Vue.js không chỉ phù hợp cho những người mới bắt đầu mà còn giúp các lập trình viên có kinh nghiệm tăng tốc quá trình phát triển. Kiến trúc component-based của Vue.js cho phép xây dựng và quản lý các thành phần giao diện một cách độc lập, giúp mã nguồn trở nên rõ ràng, dễ bảo trì và tái sử dụng. Hơn nữa, Vue.js dễ dàng tích hợp với các thư viện và framework khác, cũng như các dự án hiện có, tạo điều kiện thuận lợi cho việc mở rộng và nâng cấp ứng dụng. Cộng đồng phát triển lớn và năng động của Vue.js, cùng với nguồn tài liệu phong phú, cung cấp sự hỗ trợ mạnh mẽ và nhiều plugin hữu ích để giải quyết các vấn đề phát sinh.

ASP.NET, ngược lại, là một nền tảng back-end mạnh mẽ của Microsoft, nổi bật với hiệu suất cao và các tính năng bảo mật vượt trội. ASP.NET cung cấp khả năng xử lý nhiều yêu cầu đồng thời một cách hiệu quả, đảm bảo sự mượt mà và ổn định cho ứng dụng. Các tính năng bảo mật mạnh mẽ của ASP.NET, bao gồm xác thực, ủy quyền, mã hóa dữ liệu và bảo vệ khỏi các cuộc tấn công XSS, CSRF, SQL Injection,... rất quan trọng cho một ứng dụng thương mại điện tử như chuỗi cửa hàng bán hoa quả, đảm bảo an toàn cho dữ liệu và giao dịch của người dùng. Công cụ phát triển như Visual Studio và Visual Studio Code cung cấp nhiều tính năng hỗ trợ lập trình viên, từ IntelliSense, debugging, đến profiling, giúp tăng hiệu quả và năng suất làm việc. Thư viện phong phú và các framework hỗ trợ của ASP.NET, như Entity Framework, ASP.NET Core Identity, SignalR,... giúp giảm thiểu thời gian và công sức phát triển các tính năng phức tạp.

ASP.NET cũng được thiết kế để dễ dàng mở rộng và tích hợp với các hệ thống và dịch vụ khác, như cơ sở dữ liệu SQL Server, các dịch vụ đám mây Azure, và các API bên thứ ba, giúp xây dựng một hệ thống linh hoạt và mạnh mẽ, có khả năng xử lý tăng trưởng

lưu lượng truy cập và dữ liệu khi quy mô của chuỗi cửa hàng tăng lên. Ngoài ra, ASP.NET có một cộng đồng phát triển lớn và lâu đời, với nhiều tài liệu, ví dụ và hướng dẫn chi tiết, giúp bạn dễ dàng tìm kiếm sự hỗ trợ và giải pháp cho các vấn đề gặp phải trong quá trình phát triển. Được hỗ trợ liên tục từ Microsoft, ASP.NET luôn cập nhật và bảo mật tốt nhất, đảm bảo bạn luôn sử dụng những công nghệ mới nhất và an toàn nhất.

Tổng hợp lại, sự kết hợp giữa Vue.js và ASP.NET mang lại một giải pháp mạnh mẽ và linh hoạt cho dự án phát triển ứng dụng chuỗi cửa hàng bán hoa quả. Vue.js giúp tạo ra giao diện người dùng nhanh chóng, mượt mà và thân thiện, dễ học và dễ sử dụng, có khả năng tái sử dụng và mở rộng tốt. ASP.NET cung cấp một back-end mạnh mẽ, bảo mật và hiệu quả, hỗ trợ phát triển nhanh chóng và khả năng mở rộng cao, phù hợp với các yêu cầu phức tạp của ứng dụng thương mại điện tử. Sự kết hợp này sẽ giúp bạn hoàn thành dự án tốt nghiệp một cách hiệu quả, đạt được kết quả tốt và đáp ứng được các yêu cầu khắt khe của một ứng dụng quản lý chuỗi cửa hàng bán hoa quả.

2.2. VueJS

2.2.1. VueJS là gì?

Vue.js là một framework (khung ứng dụng) JavaScript mã nguồn mở dùng để xây dựng các giao diện người dùng (UI) động cho các ứng dụng web. Vue.js tập trung vào việc tạo các thành phần (component) tái sử dụng và dễ sử dụng, và cung cấp các tính năng để quản lý trạng thái ứng dụng (application state) một cách hiệu quả.

Vue.js có thể tích hợp vào các ứng dụng web hiện có hoặc được sử dụng để xây dựng các ứng dụng web độc lập. Vue.js cũng được đánh giá cao về hiệu suất và khả năng tương thích với các thư viện và framework khác.

Vue.js được tạo ra vào năm 2014 bởi Evan You - một nhà phát triển web tại Google. Trước đó, Evan đã có kinh nghiệm làm việc với các framework khác như AngularJS và BackboneJS, nhưng anh cảm thấy các framework này quá nặng và khó sử dụng.

Với ý tưởng tạo ra một framework nhẹ và dễ sử dụng hơn, Evan đã phát triển Vue.js bằng cách kết hợp các tính năng tốt nhất từ các framework khác. Vue.js lấy cảm hứng từ AngularJS về cách quản lý trạng thái ứng dụng, và từ React về cách tạo ra các thành phần (component) và tái sử dụng chúng.

Giờ đây, Vue.js đã nhanh chóng được cộng đồng phát triển web đón nhận và trở thành một trong những framework JavaScript phổ biến nhất hiện nay. Hiện nay, Vue.js được sử dụng để phát triển các ứng dụng web độc lập, hoặc tích hợp vào các dự án web hiện có để tạo ra các giao diện người dùng động.

Vue.js có nhiều tính năng hữu ích để xây dựng các ứng dụng web động và tạo các giao diện người dùng (UI) đẹp và dễ sử dụng. Sau đây là một số tính năng chính của Vue.js:

Tạo các thành phần (component) tái sử dụng và dễ sử dụng: Vue.js cho phép tạo các thành phần (component) tái sử dụng để giúp tách biệt các phần khác nhau của ứng dụng web và làm cho mã nguồn dễ đọc và dễ bảo trì hơn.

Quản lý trạng thái ứng dụng (application state) một cách hiệu quả: Vue.js cung cấp Vuex, một thư viện quản lý trạng thái ứng dụng giúp đơn giản hóa quá trình quản lý và theo dõi các trạng thái khác nhau của ứng dụng.

Tích hợp vào các ứng dụng web hiện có hoặc độc lập: Vue.js có thể tích hợp vào các ứng dụng web hiện có bằng cách sử dụng các thư viện hỗ trợ hoặc độc lập với các ứng dụng web mới được xây dựng.

Hiệu suất và khả năng tương thích với các thư viện và framework khác: Vue.js có hiệu suất tốt và tương thích tốt với các thư viện và framework khác, như React hoặc Angular.

Hỗ trợ điều hướng (routing): Vue.js có thể hỗ trợ điều hướng (routing) cho các ứng dụng đơn trang (Single-Page Applications) bằng cách sử dụng Vue Router, một thư viện routing chuyên dụng.

Cú pháp đơn giản và dễ hiểu: Vue.js sử dụng cú pháp đơn giản và dễ hiểu giúp cho các nhà phát triển web dễ dàng học và sử dụng nó.

Nói chung, các tính năng của Vue.js như tạo các thành phần (component) tái sử dụng, quản lý trạng thái ứng dụng (application state) hiệu quả, tích hợp vào các ứng dụng web hiện có hoặc độc lập, hiệu suất tốt và khả năng tương thích với các thư viện và framework khác, hỗ trợ điều hướng (routing), và cú pháp đơn giản và dễ hiểu đã giúp Vue.js trở thành một trong những framework JavaScript phổ biến nhất hiện nay.

Dễ dàng học và sử dụng: Vue.js có cú pháp đơn giản và dễ hiểu, cho phép người dùng nhanh chóng học và sử dụng nó. Nó cũng cung cấp cho người dùng nhiều công cụ và tài liệu hữu ích để giúp họ dễ dàng bắt đầu với Vue.js.

Tốc độ và hiệu suất: Vue.js có tốc độ và hiệu suất rất cao, giúp ứng dụng của bạn chạy mượt mà và nhanh hơn so với các framework khác.

Tính linh hoạt: Vue.js là một framework linh hoạt, cho phép bạn sử dụng các thư viện hoặc plug-in khác nhau để mở rộng chức năng của nó. Nó cũng cho phép bạn tích hợp Vue.js với các thư viện và framework khác để tạo ra các ứng dụng phức tạp hơn.

Hỗ trợ cộng đồng tốt: Vue.js có một cộng đồng lớn và đầy nhiệt huyết, cung cấp nhiều tài nguyên và hỗ trợ cho người dùng. Nó cũng có nhiều trang web và forum để trao đổi kinh nghiệm và giúp đỡ nhau khi gặp vấn đề.

Khả năng tái sử dụng: Vue.js cho phép bạn tái sử dụng các thành phần của ứng dụng, giúp bạn tiết kiệm thời gian và nỗ lực trong việc phát triển ứng dụng của mình.

Tích hợp tốt với các công cụ khác: Vue.js cung cấp tích hợp tốt với các công cụ phổ biến khác như Webpack, Babel và ESLint, giúp người dùng tối ưu hóa quá trình phát triển ứng dụng của mình.

Vue.js là một framework JavaScript rất linh hoạt và có nhiều tính năng mạnh mẽ, cho phép bạn xây dựng nhiều loại ứng dụng web khác nhau, từ các ứng dụng đơn giản đến các ứng dụng phức tạp. Một số ứng dụng của Vue.js có thể kể đến như:

Single-page applications (SPA): Vue.js được sử dụng rộng rãi để xây dựng các ứng dụng đơn trang (SPA). Với Vue Router, bạn có thể dễ dàng tạo các route cho ứng dụng của mình và giúp người dùng điều hướng giữa các trang một cách dễ dàng.

Progressive web applications (PWA): Vue.js cũng có thể được sử dụng để xây dựng các ứng dụng web tiên tiến (PWA). Nhờ vào tính năng service worker và webpack, Vue.js cho phép các ứng dụng web của bạn có thể hoạt động ngoại tuyến và tăng trải nghiệm người dùng.

Interactive user interfaces: Vue.js được thiết kế để xây dựng các giao diện người dùng tương tác và phong phú. Với v-model directive, Vue.js cho phép các thành phần của giao diện người dùng được đồng bộ hóa với các thành phần khác của ứng dụng.

Mobile applications: Vue.js cũng có thể được sử dụng để xây dựng các ứng dụng di động. Với các thư viện như Vue Native hoặc Weex, bạn có thể sử dụng Vue.js để tạo ứng dụng di động cross-platform.

Large-scale applications: Vue.js cũng được sử dụng để xây dựng các ứng dụng quy mô lớn. Với Vuex, bạn có thể quản lý trạng thái của ứng dụng của mình và tạo ra các ứng dụng phức tạp với nhiều thành phần khác nhau.

2.2.2. Ưu và nhược điểm

Giống như mọi loại công nghệ, Vue.js gây ra nhiều tranh cãi trong cộng đồng. Vậy ưu và nhược điểm của VueJS là gì? Trước tiên, hãy cùng glints tìm hiểu ưu điểm của framework này.

a. Ưu điểm

Dễ học và sử dụng: Vue.js có cú pháp đơn giản và gần gũi với HTML, CSS và JavaScript, làm cho việc học và sử dụng nó trở nên dễ dàng đối với cả người mới bắt đầu và những nhà phát triển có kinh nghiệm. Vue.js cung cấp một cách tiếp cận tự nhiên cho việc xây dựng các ứng dụng web tương tác một cách nhanh chóng.

Hiệu suất cao: Vue.js sử dụng Virtual DOM để quản lý và cập nhật giao diện người dùng một cách hiệu quả, giúp giảm thiểu tải lại trang và cải thiện trải nghiệm người dùng. Điều này làm cho ứng dụng Vue.js có hiệu suất tốt và mượt mà, đặc biệt là trong các ứng dụng web đòi hỏi sự tương tác cao.

Cộng đồng lớn và tích cực: Vue.js có một cộng đồng lớn và tích cực của các nhà phát triển trên toàn thế giới. Điều này đảm bảo rằng bạn có thể dễ dàng tìm kiếm các tài liệu, hướng dẫn, và giải pháp cho các vấn đề cụ thể mà bạn có thể gặp phải trong quá trình phát triển ứng dụng của mình.

Tích hợp dễ dàng: Vue.js có khả năng tích hợp dễ dàng với các thư viện và công nghệ khác như Vuex (quản lý trạng thái ứng dụng), Vue Router (điều hướng ứng dụng), và Vue CLI (công cụ phát triển dự án). Điều này giúp cho việc xây dựng ứng dụng phức tạp trở nên thuận tiện và linh hoạt.

Đa dạng về tính năng: Vue.js cung cấp một loạt các tính năng hữu ích cho việc phát triển ứng dụng web, bao gồm các directive, computed properties, watchers, v.v. Điều này cho phép bạn xây dựng các ứng dụng có chức năng phong phú và linh hoạt.

b. Nhược điểm

Quá dễ dàng: Mặc dù Vue.js dễ học và sử dụng, nhưng đôi khi điều này cũng có thể dẫn đến việc sử dụng không cẩn thận hoặc không nhất quán trong mã nguồn. Việc có quá nhiều cách tiếp cận để giải quyết một vấn đề có thể làm cho mã nguồn trở nên khó hiểu và khó bảo trì.

Tích hợp với các dự án lớn: Trong các dự án lớn, việc quản lý trạng thái ứng dụng và đồng bộ hóa dữ liệu có thể trở nên phức tạp. Mặc dù có Vuex để giải quyết vấn đề này, nhưng việc triển khai và quản lý Vuex trong các ứng dụng lớn có thể đòi hỏi một quá trình học và triển khai phức tạp.

Cung cấp ít tính năng so với các framework khác: So với các framework như Angular hoặc React, Vue.js có ít tính năng tích hợp sẵn. Điều này có thể là một lợi ích cho những dự án nhỏ, nhưng có thể là một hạn chế đối với các dự án lớn và phức tạp.

Hỗ trợ kém cho dự án lớn: Mặc dù Vue.js có thể được sử dụng cho các dự án lớn, nhưng so với Angular hoặc React, nó có ít công cụ và hỗ trợ ít chuyên sâu cho việc quản lý và mở rộng các dự án lớn. Điều này có thể gây ra một số khó khăn trong việc xử lý các vấn đề phức tạp và quản lý mã nguồn trong các dự án lớn.

2.2.3. Ứng dụng của VueJS

Single Page Applications (SPAs): Vue.js được sử dụng phổ biến trong việc phát triển các ứng dụng web đơn trang (Single Page Applications). Nhờ vào Virtual DOM và khả năng tương tác mượt mà, Vue.js cho phép xây dựng các ứng dụng web mạnh mẽ mà không cần tải lại trang khi người dùng tương tác.

Ứng dụng di động (Mobile Applications): Vue.js có thể được sử dụng để phát triển các ứng dụng di động thông qua các framework như NativeScript hoặc Quasar Framework. Điều này cho phép phát triển ứng dụng di động đa nền tảng với một mã nguồn duy nhất sử dụng Vue.js. Ứng dụng quản trị (Admin Panels): Vue.js thường được sử dụng để xây dựng các giao diện quản trị cho các ứng dụng web hoặc hệ thống quản lý nội dung. Các tính

năng như linh hoạt, dễ dàng tích hợp và hiệu suất cao làm cho Vue.js trở thành lựa chọn phổ biến cho việc phát triển các bảng điều khiển quản trị.

E-commerce Websites: Vue.js có thể được sử dụng để phát triển các trang web thương mại điện tử với các tính năng như xem trước sản phẩm, giỏ hàng, thanh toán và quản lý tài khoản người dùng. Vue.js có thể kết hợp tốt với các thư viện và API khác nhau để tạo ra các trải nghiệm mua sắm trực tuyến tốt nhất.

Ứng dụng tương tác thời gian thực (Real-time Applications): Vue.js có thể được sử dụng để phát triển các ứng dụng tương tác thời gian thực như chat trực tiếp, bảng xếp hạng trò chơi, hoặc các ứng dụng cộng tác nhóm. Với khả năng xử lý sự kiện và cập nhật DOM một cách mượt mà, Vue.js là một lựa chọn lý tưởng cho các ứng dụng này.

Ứng dụng đa trang (Multi-page Applications): Mặc dù Vue.js thường được sử dụng cho các ứng dụng đơn trang, nhưng nó cũng có thể được áp dụng cho các ứng dụng đa trang. Vue.js có khả năng linh hoạt để tích hợp với các trang web tồn tại, giúp cải thiện trải nghiệm người dùng mà không cần phải tạo lại toàn bộ ứng dụng.

2.3. Typescript

TypeScript là một ngôn ngữ lập trình được phát triển và duy trì bởi Microsoft. Nó là một siêu tập hợp cú pháp nghiêm ngặt của JavaScript và thêm tính năng kiểu tĩnh tùy chọn vào ngôn ngữ. TypeScript được thiết kế để phát triển các ứng dụng lớn và biên dịch sang JavaScript. Vì TypeScript là một siêu tập hợp của JavaScript, các chương trình JavaScript hiện có cũng là các chương trình TypeScript hợp lệ về mặt cú pháp, tuy nhiên, chúng có thể bị lỗi type-check.

TypeScript có thể được sử dụng để phát triển các ứng dụng JavaScript cho cả thực thi phía máy khách và phía máy chủ (như với Node.js hoặc Deno). Có nhiều tùy chọn có sẵn để biên dịch. Có thể sử dụng Trình biên dịch TypeScript mặc định, hoặc có thể sử dụng trình biên dịch Babel để chuyển đổi TypeScript sang JavaScript.

TypeScript hỗ trợ các tệp định nghĩa có thể chứa thông tin kiểu của các thư viện JavaScript hiện có, giống như tệp tiêu đề C++ có thể mô tả cấu trúc của tệp đối tượng hiện có. Điều này cho phép các chương trình khác sử dụng các giá trị được xác định trong tệp

như thế chúng là các thực thể TypeScript được nhập tĩnh. Có các tệp tiêu đề của bên thứ ba cho các thư viện phổ biến như jQuery, MongoDB và D3.js. Tiêu đề TypeScript cho các mô-đun cơ bản của Node.js cũng có sẵn, cho phép phát triển các chương trình Node.js trong TypeScript.

Bản thân trình biên dịch TypeScript được viết bằng TypeScript và được biên dịch sang JavaScript. Nó được cấp phép theo Giấy phép Apache 2.0. TypeScript được bao gồm như một ngôn ngữ lập trình hạng nhất trong Microsoft Visual Studio 2013 Update 2 trở lên, bên cạnh C# và các ngôn ngữ Microsoft khác. Một phần mở rộng chính thức cho phép Visual Studio 2012 cũng hỗ trợ TypeScript. Anders Hejlsberg, kiến trúc sư chính của C# và người tạo ra Delphi và Turbo Pascal, đã tham gia vào quá trình phát triển TypeScript.

TypeScript được công bố lần đầu tiên vào tháng 10 năm 2012 (ở phiên bản 0.8), sau hai năm phát triển nội bộ tại Microsoft. Ngay sau khi thông báo, Miguel de Icaza đã ca ngợi bản thân ngôn ngữ này, nhưng chỉ trích việc thiếu hỗ trợ IDE trưởng thành ngoài Microsoft Visual Studio, vốn chưa có trên Linux và OS X vào thời điểm đó. Ngày nay, có hỗ trợ trong các IDE khác, đặc biệt là trong Eclipse, thông qua một trình cắm thêm do Palantir Technologies đóng góp. Các trình soạn thảo văn bản khác nhau, bao gồm Emacs, Vim, Webstorm, Atom và Visual Studio Code của riêng Microsoft cũng hỗ trợ TypeScript.

TypeScript 0.9, được phát hành vào năm 2013, đã thêm hỗ trợ cho generic.

TypeScript 1.0 được phát hành tại hội nghị nhà phát triển Build của Microsoft vào năm 2014. Visual Studio 2013 Update 2 cung cấp hỗ trợ tích hợp cho TypeScript. Vào tháng 7 năm 2014, nhóm phát triển đã công bố một trình biên dịch TypeScript mới, đạt hiệu suất gấp 5 lần. Đồng thời, mã nguồn, ban đầu được lưu trữ trên CodePlex, đã được chuyển sang GitHub.

Vào ngày 22 tháng 9 năm 2016, TypeScript 2.0 đã được phát hành; nó đã giới thiệu một số tính năng, bao gồm khả năng cho các lập trình viên tùy ý ngăn các biến được gán giá trị null, đôi khi được gọi là sai lầm hàng tỷ đô la.

TypeScript 3.0 được phát hành vào ngày 30 tháng 7 năm 2018, mang đến nhiều bổ sung ngôn ngữ như tuple (bộ giá trị) trong rest parameter (tham số phần còn lại) và spread expression (biểu thức trải rộng), rest parameter với các kiểu tuple, generic rest parameter...

TypeScript 4.0 được phát hành vào ngày 20 tháng 8 năm 2020. Mặc dù 4.0 không giới thiệu bất kỳ thay đổi đột phá nào, nhưng nó đã bổ sung các tính năng ngôn ngữ như Hệ thống JSX tùy chỉnh và các kiểu Variadic Tuple (bộ giá trị đa dạng).

TypeScript bắt nguồn từ những thiếu sót của JavaScript đối với việc phát triển các ứng dụng quy mô lớn ở cả Microsoft và các khách hàng bên ngoài của công ty này. Những thách thức đối với việc xử lý mã JavaScript phức tạp đã dẫn đến nhu cầu về công cụ tùy chỉnh để dễ dàng phát triển các thành phần trong ngôn ngữ này.

Các nhà phát triển TypeScript đã tìm kiếm một giải pháp không phá vỡ tính tương thích với tiêu chuẩn và khả năng hỗ trợ đa nền tảng của nó. Biết rằng đề xuất tiêu chuẩn ECMAScript hiện tại hứa hẹn hỗ trợ cho lập trình dựa trên lớp trong tương lai, TypeScript đã được xây dựng dựa trên đề xuất đó. Điều này dẫn đến một trình biên dịch JavaScript với một tập hợp các phần mở rộng ngôn ngữ cú pháp, một siêu tập hợp dựa trên đề xuất ECMAScript, chuyển đổi các phần mở rộng đó thành JavaScript thông thường. Theo nghĩa này, tính năng lớp của TypeScript là bản xem trước những gì mong đợi của ECMAScript 2015. Một khía cạnh duy nhất không có trong đề xuất, nhưng được thêm vào TypeScript, là tính năng kiểu tĩnh tùy chọn, cho phép phân tích ngôn ngữ tĩnh, tạo điều kiện hỗ trợ công cụ và IDE.

2.3.1. Interfaces

Trong TypeScript, sử dụng interface là cách khai báo trước cấu trúc cần phải có cho các class, các function, các biến...

TypeScript không biên dịch interface sang mã JavaScript. Typescript chỉ dùng interface cho mục đích hỗ trợ kiểm tra kiểu và cấu trúc của các biến, function, class ... sử dụng theo interface được chính xác.

Mỗi interface được định nghĩa với từ khóa interface. Khi định nghĩa, bạn khai báo các property và method. Method thì có thể dùng kiểu function thông thường hoặc arrow function đều được. Trong TypeScript, interface có thể dùng để định nghĩa 1 kiểu dữ liệu (type) mới và cũng có thể implement nó trong 1 class.

Sử dụng interface trong TypeScript khá đa dạng. Bạn có thể dùng interface như là 1 kiểu hàm. Có thể sử dụng interface để tạo kiểu array. Tức dùng interface để khai báo kiểu của các giá trị trong mảng và kiểu cho index của mảng.

Trong interface có thể chỉ định một thuộc tính nào đó là tùy chọn, bằng cách dùng dấu ? Khi đó các đối tượng theo interface có thể định nghĩa thuộc tính đó hoặc không. Trong interface có thể đánh dấu 1 thuộc tính nào đó là read only. Nghĩa là chỉ có thể đọc giá trị của thuộc tính mà không thể gán giá trị cho nó. Mỗi interface có thể extend 1 hoặc nhiều interface khác.

Interface trong TypeScript cũng có thể implement cho class. Class nào được implement với interface thì phải khai báo đủ các hàm và thuộc tính trong interface. Ngoài ra có thể định nghĩa thêm các thuộc tính khác tùy ý.

Sử dụng interface trong typescript rất linh động phải không. Đây là công cụ mạnh giúp bạn code javascript ngon lành, nhất là trong các dự án nghiệp vụ phức tạp.

2.3.2. Classes

Class trong typescript là một cách thức mô tả các biến có cấu trúc phức tạp, Class được tạo ra để khai báo các cấu trúc dành cho việc tạo các biến sau này. Khi tạo class, bạn sẽ khai báo nhiều thuộc tính (còn gọi là property) và nhiều hàm (method, còn gọi là các hành động) làm mẫu cho các biến sẽ tạo theo class.

Trong class, constructor là một hàm đặc biệt. Nếu bạn có khai báo hàm có tên constructor trong class thì hàm này sẽ chạy tự động khi 1 biến được tạo ra, Tức khi bạn tạo biến bằng từ khóa new.

Trong các hàm của class, bạn có thể dùng từ khóa this – đây là từ khóa diễn tả “đối tượng hiện hành”. Mỗi khi cần truy xuất đến thuộc tính của class thì dùng từ khóa this.

Class là hình mẫu cho các biến sẽ tạo. Biến được tạo theo class sẽ có cấu trúc đúng như khai báo trong class. Do đó class sanpham sẽ có các thuộc tính id, ten, usd, status, tienvn(), trangthai(). Biến này còn gọi là đối tượng, là 1 instance của class.

Các thành phần trong class là property và method có thể được chỉ định truy xuất ở các cấp độ khác nhau. Sau đây là một các access modifier (cấp độ truy xuất) trong TypeScript: public, private , protected, readonly.

Public là cấp độ truy xuất mặc định cho các property và method trong class TypeScript. Public cho phép truy xuất ở trong class cũng như ngoài class, không có giới hạn.

Trong class, thuộc tính và hàm nào được chỉ định cấp độ truy xuất là `private` thì nó chỉ có thể được sử dụng trong class mà thôi, không thể truy xuất được bên ngoài class chứa nó.

`Protected` là cấp độ truy xuất chỉ có thể truy xuất từ bên trong class hoặc từ các class con. Bên ngoài class truy xuất không được.

`Readonly` để đánh dấu 1 property trong class là chỉ đọc được từ ngoài class nhưng không gán được. Vì lý do này mà bạn cần khởi tạo giá trị cho property trong hàm constructor của class.

Mỗi class trong TypeScript có thể kế thừa từ 1 class cha. Thực hiện bằng cách dùng từ khóa `extends` khi tạo . Class con kế thừa tất cả thuộc tính và hàm của class cha ngoại trừ những cái `private` và constructor. Trong class con nếu có hàm constructor thì phải gọi hàm constructor của cha cho bằng cách sử dụng hàm `super()` trước khi chạy code trong class con.

Trong class, thành viên nào (property, method) được chỉ định `static` , tức có từ khóa `static` ở trước thì sẽ được truy xuất bằng cách dùng cú pháp `classname.ten` mà không cần đến việc tạo đối tượng. Static member khác với public member. Do đó chúng có thể giống tên nhau trong class. Static không thể truy xuất thông qua đối tượng là instance của class còn public member thì được.

Lớp trừu tượng được tạo bằng từ khóa `abstract`. Mục đích của việc tạo lớp trừu tượng là để làm mẫu cho các lớp con kế thừa mà không cần code cụ thể. Với lớp trừu tượng thì không tạo được biến instance

Một lớp trừu tượng thường là nơi khai báo tên và tham số cho các method trừu tượng hoặc các property. Các lớp con bắt buộc phải định nghĩa các methods trừu tượng

2.3.3. Modules

Kể từ ES6, JavaScript bắt đầu hỗ trợ các modules như một phần của ngôn ngữ. TypeScript chia sẻ các khái niệm module giống với JavaScript.

Một module trong TypeScript có chứa cả khai báo và code. Một module thực thi trong phạm vi riêng của nó, không phải trong phạm vi toàn cục. Điều đó có nghĩa là khi bạn khai báo các biến, hàm, lớp, giao diện, v.v., trong một module, chúng không hiển thị

bên ngoài module, trừ khi bạn xuất chúng một cách rõ ràng bằng cách sử dụng câu lệnh `export`.

Trong module này, chúng ta đặt từ khóa `export` trước từ khóa `interface` để các modules có thể sử dụng nó. Nếu bạn không sử dụng từ khóa `export`, thì `interface Validator` sẽ là kiểu `private` trong `Validator.ts`, do đó, bạn không thể sử dụng từ các modules khác.

Có thể đưa các thành phần được `export` trong module vào file khác để dùng. Có thể `import` 1 thành phần cụ thể nào đó trong module khác hoặc `import` toàn bộ những gì được module khác `export` cũng được.

Trong 1 module, có thể `export` tiếp tục 1 vài thành phần đã `import` hoặc `export` toàn bộ từ module đã `import`, khi `export` có thể đặt tên mới cho 1 thành phần.

Trong JavaScript ES6, từ khóa `module` được sử dụng để định nghĩa một module, điều này cho phép bạn định nghĩa nhiều module trong một file. TypeScript sử dụng từ khóa `namespace` với cú pháp và mục đích tương tự.

Chú ý: Cả hai từ khóa `"module"` và `"namespace"` đều có thể sử dụng trong TypeScript với cú pháp và mục đích sử dụng tương tự nhau. Tuy nhiên các tài liệu TypeScript không khuyến khích sử dụng từ khóa `"module"` trong mã TypeScript, nó chỉ nên dùng trong mã JavaScript.

Trong TypeScript từ khóa `"namespace"` mạnh hơn một chút so với từ khóa `"module"`. Bạn có thể định nghĩa một module trên nhiều file với từ khóa `"namespace"` và sử dụng cờ `"--outFile"` nếu muốn chúng ghép lại với nhau thành một file duy nhất tại thời điểm biên dịch.

2.3.4. Ưu điểm của Typescripts

a. Ưu điểm

Kiểu dữ liệu tĩnh: TypeScript cho phép bạn xác định kiểu dữ liệu của biến, tham số và giá trị trả về của hàm, giúp phát hiện lỗi và làm tăng tính ổn định của mã.

Hỗ trợ ECMAScript mới: TypeScript thường hỗ trợ các tính năng của các phiên bản ECMAScript mới, giúp bạn sử dụng các tính năng mới nhất của JavaScript mà không cần chờ đợi sự hỗ trợ của các trình duyệt.

Cải thiện tính bảo trì: Kiểu dữ liệu tĩnh và tính năng hướng đối tượng của TypeScript làm cho mã dễ bảo trì hơn, giúp ít lỗi hơn và dễ dàng hiểu hơn.

Cộng đồng lớn: TypeScript có một cộng đồng lớn và sôi động, với nhiều tài liệu, thư viện và công cụ hỗ trợ.

Tích hợp với các công cụ phổ biến: TypeScript có thể tích hợp dễ dàng với nhiều công cụ phát triển phổ biến như Visual Studio Code, Webpack và Angular.

b. Nhược điểm

Học phần mở rộng: Việc học TypeScript có thể đòi hỏi một thời gian để hiểu và làm quen với các khái niệm như kiểu dữ liệu tĩnh và hướng đối tượng nếu bạn không quen với chúng.

Tăng thời gian phát triển ban đầu: Việc định nghĩa kiểu dữ liệu có thể làm tăng thời gian phát triển ban đầu so với việc viết mã JavaScript thuần túy.

Khó khăn trong tích hợp với mã JavaScript cũ: Nếu bạn có một dự án sử dụng mã JavaScript đã tồn tại, việc tích hợp TypeScript có thể gặp khó khăn do cú pháp và cách thức làm việc của TypeScript khác với JavaScript thuần túy.

2.4. API (Application Programming Interface)

2.4.1. API là gì?

Một giao diện lập trình ứng dụng (tiếng Anh: application programming interface, viết tắt: API) là một giao diện mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng. Chẳng hạn, một chương trình máy tính có thể (và thường là phải) dùng các hàm API của hệ điều hành để xin cấp phát bộ nhớ và truy xuất tập tin. Nhiều loại hệ thống và ứng dụng thực hiện API, như các hệ thống đồ họa, cơ sở dữ liệu, mạng, dịch vụ web, và ngay cả một số trò chơi máy tính. Đây là phần mềm hệ thống cung cấp đầy đủ các chức năng và các tài nguyên mà các lập trình viên có thể rút ra từ đó để tạo nên các tính năng giao tiếp người - máy như: các trình đơn kéo xuống, tên lệnh, hộp hội thoại, lệnh bàn phím và các cửa sổ. Một trình ứng dụng có thể sử dụng nó để yêu cầu và thi hành các dịch vụ cấp thấp do hệ điều hành của máy tính thực hiện. Hệ

giao tiếp lập trình ứng dụng giúp ích rất nhiều cho người sử dụng vì nó cho phép tiết kiệm được nhiều thời gian tìm hiểu các chương trình mới, do đó khích lệ mọi người dùng nhiều ứng dụng hơn.

Một trong các mục đích chính của một API là cung cấp khả năng truy xuất đến một tập các hàm hay dùng — ví dụ, hàm để vẽ các cửa sổ hay các icon trên màn hình. Các API, cũng như hầu hết các interfaces, là trừu tượng (abstract). Phần mềm mà muốn cung cấp truy xuất đến chính nó thông qua các API cho sẵn, phải hiện thực API đó. Trong nhiều tình huống, một API thường là một phần của bộ SDK, hay software development kit. Một bộ SDK có thể bao gồm một API cũng như các công cụ/phần cứng, vì thế hai thuật ngữ này không thay thế cho nhau được.

Có nhiều mô hình thiết kế khác nhau cho các APIs. Interfaces nhằm là cách thực thi nhanh nhất thường gồm các tập các hàm, thủ tục, biến và các cấu trúc dữ liệu. Tuy nhiên, các mô hình khác vẫn tồn tại, như bộ thông dịch dùng để ước giá biểu thức trong ECMAScript/JavaScript. Một API tốt thường cung cấp một "hộp đen" hay là một lớp trừu tượng (abstraction layer) bao bọc nó, nhằm đảm bảo là nhà lập trình không thể biết cách hiện thực cụ thể bên trong của mỗi hàm trong API. Điều này làm cho việc thiết kế lại hay cải tiến hàm của API đó trở nên dễ dàng hơn vì nó không làm đổ ố các đoạn mã khác mà có sử dụng các hàm đó.

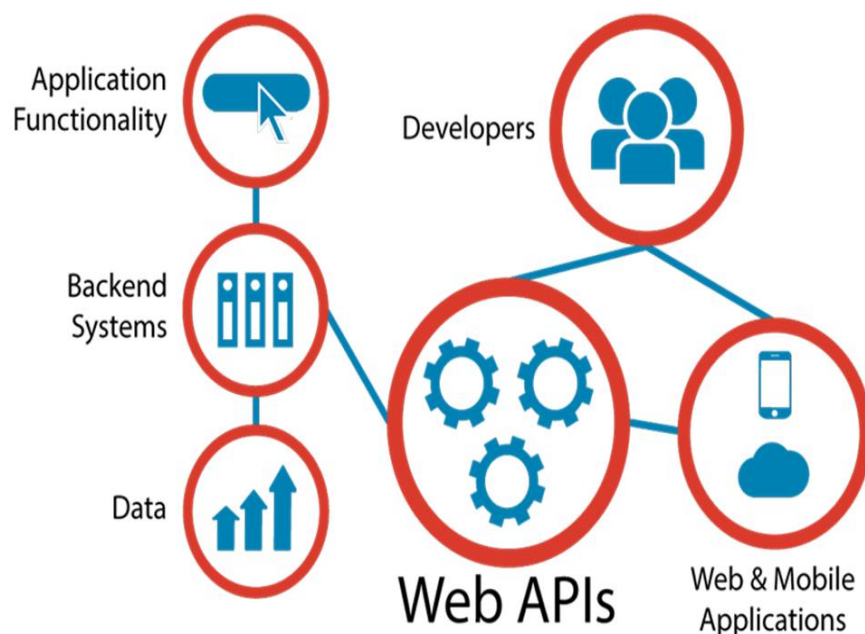
Có hai dòng chính sách đối với việc công bố các APIs:

Một số công ty bảo vệ APIs của họ một cách mạnh mẽ. Ví dụ, Sony thường chỉ cung cấp API chính thức của PlayStation 2 cho các nhà phát triển PlayStation có đăng ký. Điều này là vì Sony muốn giới hạn những người có thể viết trò chơi trên PlayStation 2, và muốn thu lợi nhuận từ những người này càng nhiều càng tốt. Đây thường là chính sách đối với các công ty mà họ không thu lợi từ việc bán các hiện thực API của họ. Tuy nhiên, PlayStation 3 là công bố hoàn toàn APIs.

Một số công ty thì cung cấp miễn phí APIs. Ví dụ, Microsoft công bố hầu như hoàn toàn thông tin về các API, để cho các phần mềm có thể được viết chạy trên nền Windows. Việc bán của các phần mềm hãng thứ 3 đồng thời với việc phải mua Hệ điều hành Microsoft Windows. Đây thường là các công ty thu lợi nhuận từ việc bán các hiện thực API.

Một số APIs, chẳng hạn các API là chuẩn cho một hệ điều hành, được hiện thực dưới dạng các thư viện mã độc lập được phân phối kèm theo hệ điều hành. Một số khác thì đòi hỏi nhà sản xuất phần mềm phải tích hợp API trực tiếp vào trong chương trình. Microsoft Windows APIs đi kèm theo hệ điều hành cho phép mọi người có thể sử dụng chúng. Phần mềm cho các hệ thống nhúng như thiết bị chơi trò chơi thường thuộc vào loại tích hợp vào trong ứng dụng. Trong khi các tài liệu API chính thức của PlayStation là nên đọc, nhưng nó chẳng giúp ích gì nếu ta chẳng có các hiện thực của nó, dưới dạng một thư viện độc lập hay bộ phát triển phần mềm.

Một API mà cho phép truy xuất và sử dụng tự do được gọi là "mở." Các APIs được cung cấp bởi phần mềm mở (như mọi phần mềm được phân phối theo giấy phép đăng ký GNU), là mở theo đúng nghĩa, vì mọi người có thể xem mã nguồn của phần mềm và tìm ra API. Mặc dù việc tham khảo hiện thực vẫn tồn tại cho một API (như với Microsoft Windows cho Win32 API), thì việc tạo thêm các hiện thực bổ sung vẫn có thể diễn ra. Ví dụ, hầu hết các Win32 API có thể được cung cấp từ hệ thống UNIX dùng phần mềm tên là Wine.



Hình 2. 1 Web API

2.4.2. Ứng dụng API

Web API: là hệ thống API được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến Web API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Ví dụ: Bạn thiết kế chức năng login thông Google, Facebook, Twitter, Github... Điều này có nghĩa là bạn đang gọi đến API của. Hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API.

API trên hệ điều hành: Windows hay Linux có rất nhiều API, họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Nó giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.

API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau và nó cũng giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác. Ví dụ bạn có thể dùng Php để yêu cầu một thư viện tạo file PDF được viết bằng C++.

2.4.3. Web API là gì?

Web API là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS.

Những điểm nổi bật của Web API:

Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu. Nó giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

Tự động hóa sản phẩm: Với web API, chúng ta sẽ tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng năng suất và tạo hiệu quả công việc cao hơn.

Khả năng tích hợp linh động: API cho phép lấy nội dung từ bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng. API hoạt động như

một chiếc cổng, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.

Cập nhật thông tin thời gian thực: API có chức năng thay đổi và cập nhật thay đổi theo thời gian thực. Với công nghệ này, dữ liệu sẽ được truyền đi tốt hơn, thông tin chính xác hơn, dịch vụ cung cấp linh hoạt hơn.

Có tiêu chuẩn chung để sử dụng: Bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ sử dụng. Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

2.4.4. Web API hoạt động như nào?

Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.

Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.

Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu...

2.4.5. Ưu và nhược điểm của Web API

a. Ưu điểm

- Web API được sử dụng hầu hết trên các ứng dụng desktop, ứng dụng mobile và ứng dụng website.
- Linh hoạt với các định dạng dữ liệu khi trả về client: Json, XML hay định dạng khác.
- Nhanh chóng xây dựng HTTP service: URI, request/response headers, caching, versioning, content formats và có thể host trong ứng dụng hoặc trên IIS.

- Mã nguồn mở, hỗ trợ chức năng RESTful đầy đủ, sử dụng bởi bất kì client nào hỗ trợ XML, Json.

- Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

- Giao tiếp hai chiều được xác nhận trong các giao dịch, đảm bảo độ tin cậy cao.

b. Nhược điểm

- Do web API còn khá mới nên chưa thể đánh giá nhiều về nhược điểm của mô hình nay. Tuy nhiên, có hai nhược điểm dễ dàng nhận thấy:

- Web API chưa hoàn toàn phải là RESTful service, mới chỉ hỗ trợ mặc định GET, POST

- Để sử dụng hiệu quả cần có kiến thức chuyên sâu, có kinh nghiệm backend tốt

- Tốn thời gian và chi phí cho việc phát triển, nâng cấp và vận hành

- Có thể gặp vấn đề về bảo mật khi hệ thống bị tấn công nếu không giới hạn điều kiện kỹ.

2.5. Asp .Net framework

ASP.NET (viết tắt của Active Server Pages .NET) là một khung phần mềm do Microsoft phát triển để xây dựng các ứng dụng web động. ASP.NET được xây dựng trên nền tảng .NET Framework và cung cấp nhiều tính năng mạnh mẽ để phát triển các ứng dụng web hiện đại, bảo mật và hiệu quả.

Ưu điểm của ASP.NET:

- Miễn phí và mã nguồn mở: ASP.NET là mã nguồn mở và được cung cấp miễn phí bởi Microsoft, giúp cho mọi người có thể dễ dàng truy cập và sử dụng.

- Lập trình hướng đối tượng: ASP.NET hỗ trợ lập trình hướng đối tượng, giúp cho việc phát triển các ứng dụng web dễ dàng, hiệu quả và bảo trì.

- Mạnh mẽ và linh hoạt: ASP.NET cung cấp nhiều tính năng mạnh mẽ để phát triển các ứng dụng web đa dạng, bao gồm các ứng dụng web Forms, ứng dụng web MVC, dịch vụ web, REST API, v.v.

- Hỗ trợ cộng đồng lớn: ASP.NET có cộng đồng người dùng và nhà phát triển lớn, giúp cho việc tìm kiếm hỗ trợ và giải đáp thắc mắc trở nên dễ dàng hơn.

Ứng dụng của ASP.NET:

- Xây dựng các ứng dụng web: ASP.NET được sử dụng để xây dựng các ứng dụng web đa dạng, bao gồm các trang web thương mại điện tử, trang web tin tức, trang web mạng xã hội, v.v.
- Phát triển các dịch vụ web: ASP.NET hỗ trợ phát triển các dịch vụ web SOAP và RESTful, giúp cho việc trao đổi dữ liệu giữa các ứng dụng trở nên dễ dàng hơn.
- Tạo các ứng dụng di động: ASP.NET có thể được sử dụng để xây dựng các ứng dụng di động lai bằng Xamarin.
- Phát triển các ứng dụng IoT: ASP.NET có thể được sử dụng để xây dựng các ứng dụng Internet of Things (IoT) để kết nối và quản lý các thiết bị thông minh.

Các thành phần chính của ASP.NET:

- Common Language Runtime (CLR): CLR là môi trường thực thi cho các ứng dụng .NET, giúp cho việc quản lý bộ nhớ, bảo mật và hiệu suất của ứng dụng.
- .NET Framework Class Library (FCL): FCL cung cấp một tập hợp các lớp và thành phần sẵn có để phát triển các ứng dụng .NET, bao gồm các lớp để truy cập dữ liệu, xử lý chuỗi, quản lý mạng, v.v.
- ASP.NET Web Forms: Web Forms là một mô hình lập trình web truyền thống trong ASP.NET, giúp cho việc xây dựng các ứng dụng web với giao diện người dùng kéo thả.
- ASP.NET MVC (Model-View-Controller): MVC là một mô hình lập trình web hiện đại trong ASP.NET, giúp cho việc xây dựng các ứng dụng web với kiến trúc rõ ràng, dễ bảo trì và dễ kiểm thử.
- ASP.NET Web API: Web API là một khung phần mềm để phát triển các dịch vụ web RESTful trong ASP.NET.

2.6. Thanh toán vn pay

VNPay là một trong những công ty cung cấp dịch vụ thanh toán điện tử hàng đầu tại Việt Nam. Được thành lập bởi Công ty Cổ phần Giải pháp Thanh toán Việt Nam (VNPAY), VNPay cung cấp nhiều giải pháp thanh toán di động và trực tuyến nhằm đáp ứng nhu cầu của người dùng và doanh nghiệp.

Các dịch vụ và sản phẩm chính của VNPay:

VNPay-QR:

Mã QR thanh toán: VNPay-QR là một trong những dịch vụ nổi bật nhất của VNPay, cho phép người dùng thực hiện thanh toán bằng cách quét mã QR. Điều này giúp đơn giản hóa quá trình thanh toán tại các cửa hàng, siêu thị, nhà hàng, và nhiều địa điểm khác.

Hỗ trợ nhiều ngân hàng: VNPay-QR được tích hợp với nhiều ngân hàng lớn tại Việt Nam, giúp người dùng có thể thanh toán trực tiếp từ tài khoản ngân hàng mà không cần sử dụng tiền mặt hay thẻ tín dụng.

Cổng thanh toán VNPay:

Thanh toán trực tuyến: Cổng thanh toán VNPay cho phép các doanh nghiệp tích hợp dịch vụ thanh toán trực tuyến vào website hoặc ứng dụng của họ, hỗ trợ các phương thức thanh toán như thẻ tín dụng, thẻ ghi nợ, và chuyển khoản ngân hàng.

An toàn và bảo mật: VNPay tuân thủ các tiêu chuẩn bảo mật quốc tế để đảm bảo an toàn cho các giao dịch trực tuyến.

Ứng dụng di động:

Ví điện tử: VNPay cung cấp ứng dụng ví điện tử cho phép người dùng nạp tiền, chuyển tiền, và thanh toán các dịch vụ như điện, nước, internet, và nhiều dịch vụ khác.

Tích hợp nhiều dịch vụ: Ứng dụng ví điện tử VNPay thường tích hợp nhiều tiện ích khác nhau như mua vé xem phim, đặt vé máy bay, và thanh toán học phí.

Giải pháp thanh toán cho doanh nghiệp:

Máy POS: VNPay cung cấp các giải pháp thanh toán qua máy POS, giúp các doanh nghiệp dễ dàng chấp nhận thanh toán bằng thẻ và mã QR.

Giải pháp E-commerce: Hỗ trợ các doanh nghiệp thương mại điện tử tích hợp các phương thức thanh toán đa dạng, giúp tối ưu hóa quá trình mua sắm trực tuyến cho khách hàng.

Lợi ích của VNPay:

Tiện lợi: VNPay cung cấp các giải pháp thanh toán đa dạng, giúp người dùng và doanh nghiệp thực hiện các giao dịch một cách nhanh chóng và thuận tiện.

An toàn: Các giải pháp của VNPay được thiết kế với các tiêu chuẩn bảo mật cao, đảm bảo an toàn cho thông tin và giao dịch của người dùng.

Hỗ trợ đa nền tảng: VNPay hỗ trợ nhiều nền tảng và thiết bị, từ ứng dụng di động đến website và máy POS, đáp ứng nhu cầu thanh toán của nhiều đối tượng người dùng khác nhau.

Phổ biến: Với mạng lưới rộng khắp và sự hợp tác với nhiều ngân hàng và đối tác kinh doanh, VNPay trở thành một trong những phương thức thanh toán phổ biến và đáng tin cậy tại Việt Nam.

Cách sử dụng VNPay:

Đăng ký tài khoản: Người dùng cần đăng ký tài khoản VNPay thông qua ứng dụng hoặc website của VNPay.

Liên kết tài khoản ngân hàng: Liên kết tài khoản ngân hàng hoặc thẻ tín dụng vào VNPay để có thể thực hiện các giao dịch.

Thực hiện thanh toán: Sử dụng VNPay để thanh toán hóa đơn, mua sắm trực tuyến, chuyển tiền, hoặc quét mã QR tại các điểm chấp nhận VNPay.

VNPay đã và đang đóng góp tích cực vào sự phát triển của hệ sinh thái thanh toán điện tử tại Việt Nam, mang lại nhiều tiện ích cho người dùng và doanh nghiệp trong việc thực hiện các giao dịch tài chính.

2.7. Google map

2.7.1 Google cloud

Google Cloud là một bộ dịch vụ điện toán đám mây được phát triển và cung cấp bởi Google. Nó cung cấp một loạt các công cụ và dịch vụ để hỗ trợ các doanh nghiệp và nhà phát triển trong việc xây dựng, triển khai, và quản lý các ứng dụng và dịch vụ của họ trên nền tảng đám mây. Dưới đây là một cái nhìn tổng quan về Google Cloud:

Các dịch vụ chính của Google Cloud:

Điện toán (Compute):

Google Compute Engine (GCE): Cung cấp máy ảo (VM) chạy trên cơ sở hạ tầng của Google, cho phép người dùng chạy các ứng dụng và dịch vụ của họ.

Google Kubernetes Engine (GKE): Dịch vụ quản lý Kubernetes, giúp triển khai, quản lý và mở rộng các ứng dụng container.

Cloud Functions: Dịch vụ tính toán serverless cho phép chạy mã mà không cần quản lý các máy chủ.

App Engine: Nền tảng phát triển ứng dụng serverless hoàn toàn, giúp triển khai các ứng dụng web và di động mà không cần quản lý cơ sở hạ tầng.

Lưu trữ (Storage):

Google Cloud Storage: Dịch vụ lưu trữ đối tượng cho phép lưu trữ và truy cập dữ liệu không có cấu trúc ở quy mô lớn.

Cloud SQL: Dịch vụ cơ sở dữ liệu quản lý cho MySQL, PostgreSQL và SQL Server.

Cloud Bigtable: Dịch vụ cơ sở dữ liệu NoSQL hiệu suất cao, được thiết kế cho các khối lượng công việc lớn và phân tích dữ liệu.

Cloud Spanner: Dịch vụ cơ sở dữ liệu quan hệ có khả năng mở rộng toàn cầu, hỗ trợ các giao dịch mạnh mẽ và độ trễ thấp.

Mạng (Networking):

Virtual Private Cloud (VPC): Cung cấp mạng riêng ảo để kết nối và bảo vệ tài nguyên đám mây.

Cloud Load Balancing: Phân phối lưu lượng truy cập trên nhiều tài nguyên để đảm bảo hiệu suất và độ sẵn sàng cao.

Cloud CDN: Dịch vụ mạng phân phối nội dung giúp giảm độ trễ và cải thiện trải nghiệm người dùng bằng cách phân phối nội dung từ các vị trí gần người dùng hơn.

Dữ liệu và phân tích (Data and Analytics):

BigQuery: Dịch vụ phân tích dữ liệu lớn serverless cho phép thực hiện các truy vấn SQL trên các tập dữ liệu lớn.

Dataflow: Dịch vụ xử lý dữ liệu stream và batch, hỗ trợ mô hình lập trình Apache Beam.

Dataproc: Dịch vụ quản lý Hadoop và Spark để xử lý dữ liệu lớn trên đám mây.

AI và Machine Learning:

AI Platform: Dịch vụ xây dựng, triển khai và quản lý các mô hình machine learning.

AutoML: Bộ công cụ cho phép các nhà phát triển không có kinh nghiệm sâu về machine learning có thể tạo ra các mô hình tùy chỉnh.

Cloud Vision API: Dịch vụ nhận diện và phân loại hình ảnh.

Cloud Speech-to-Text và Text-to-Speech: Dịch vụ chuyển đổi giọng nói thành văn bản và ngược lại.

Công cụ phát triển (Developer Tools):

Cloud Source Repositories: Dịch vụ lưu trữ mã nguồn.

Cloud Build: Dịch vụ tích hợp liên tục và triển khai liên tục (CI/CD) để xây dựng và triển khai ứng dụng.

Quản lý và bảo mật (Management and Security):

Cloud IAM (Identity and Access Management): Quản lý quyền truy cập cho các tài nguyên Google Cloud.

Cloud Key Management: Dịch vụ quản lý và bảo vệ các khóa mã hóa.

Cloud Monitoring và Logging: Giám sát hiệu suất và ghi nhật ký hoạt động của các dịch vụ đám mây.

Lợi ích của Google Cloud:

Khả năng mở rộng: Hạ tầng của Google Cloud cho phép mở rộng quy mô ứng dụng và dịch vụ từ nhỏ đến lớn một cách linh hoạt.

Hiệu suất cao: Sử dụng hạ tầng của Google, bao gồm các trung tâm dữ liệu và mạng lưới toàn cầu, giúp đảm bảo hiệu suất và độ sẵn sàng cao.

Bảo mật mạnh mẽ: Google Cloud cung cấp nhiều lớp bảo mật và tuân thủ các tiêu chuẩn quốc tế về bảo mật dữ liệu.

Đổi mới nhanh chóng: Google Cloud cung cấp các công cụ và dịch vụ tiên tiến, hỗ trợ doanh nghiệp nhanh chóng triển khai các giải pháp mới.

Sử dụng Google Cloud:

Để bắt đầu sử dụng Google Cloud, người dùng cần tạo tài khoản Google Cloud và có thể tận dụng các khoản tín dụng miễn phí mà Google cung cấp cho người dùng mới.

Sau đó, họ có thể truy cập Google Cloud Console để quản lý và triển khai các dịch vụ mà họ cần. Google Cloud cũng cung cấp nhiều tài liệu và hướng dẫn chi tiết để hỗ trợ người dùng trong quá trình học tập và sử dụng các dịch vụ của mình.

2.7.2. Google map

Google Maps là một dịch vụ bản đồ và điều hướng trực tuyến được phát triển bởi Google. Nó cung cấp bản đồ, hình ảnh vệ tinh, chế độ xem phố (Street View), thông tin giao thông thời gian thực, và các công cụ điều hướng cho người dùng trên toàn thế giới. Dưới đây là những tính năng chính và ứng dụng của Google Maps:

Các Tính Năng Chính của Google Maps:

Bản đồ và Hình ảnh Vệ tinh:

Bản đồ chi tiết: Cung cấp bản đồ chi tiết của hầu hết các quốc gia và vùng lãnh thổ trên thế giới.

Hình ảnh vệ tinh: Cung cấp hình ảnh vệ tinh cập nhật, cho phép người dùng xem cảnh quan từ trên cao.

Street View (Chế độ xem phố):

Hình ảnh 360 độ: Cung cấp hình ảnh toàn cảnh 360 độ của các con phố và đường đi, giúp người dùng khám phá các địa điểm như thể họ đang đứng ngay tại đó.

Điều hướng và Lộ trình:

Chỉ đường: Cung cấp chỉ đường chi tiết cho việc lái xe, đi bộ, đi xe đạp, và sử dụng phương tiện giao thông công cộng.

Điều hướng thời gian thực: Cung cấp hướng dẫn điều hướng bằng giọng nói và thông tin giao thông thời gian thực, giúp người dùng tránh ùn tắc và tìm lộ trình tốt nhất.

Thông tin Giao thông Thời gian Thực:

Giao thông: Hiện thị thông tin giao thông thời gian thực, bao gồm tắc đường, tai nạn, và các sự cố giao thông khác.

Dự đoán thời gian di chuyển: Dự đoán thời gian di chuyển dựa trên dữ liệu giao thông hiện tại và lịch sử.

Tìm kiếm Địa điểm và Đánh giá:

Tìm kiếm địa điểm: Cho phép tìm kiếm các địa điểm như nhà hàng, khách sạn, cửa hàng, và các điểm quan tâm khác.

Đánh giá và nhận xét: Người dùng có thể xem và để lại đánh giá, nhận xét về các địa điểm.

Lưu và Chia sẻ Địa điểm:

Lưu địa điểm yêu thích: Người dùng có thể lưu các địa điểm yêu thích để truy cập sau.

Chia sẻ vị trí: Cho phép người dùng chia sẻ vị trí hiện tại của họ với bạn bè và gia đình.

Các Ứng Dụng của Google Maps:

Hướng dẫn Du lịch và Khám phá:

Lập kế hoạch du lịch: Google Maps giúp người dùng lập kế hoạch cho các chuyến du lịch bằng cách tìm kiếm và lưu các điểm tham quan, nhà hàng, và khách sạn.

Khám phá địa phương: Người dùng có thể khám phá các địa điểm mới và tìm kiếm các hoạt động giải trí xung quanh họ.

Điều hướng và Di chuyển Hàng Ngày:

Đi làm và đi học: Google Maps cung cấp lộ trình tốt nhất để đi làm, đi học, và các hoạt động hàng ngày khác.

Dịch vụ giao thông công cộng: Cung cấp thông tin về các tuyến xe buýt, tàu điện ngầm, và các phương tiện giao thông công cộng khác.

Hỗ trợ Kinh doanh:

Danh bạ doanh nghiệp: Các doanh nghiệp có thể đăng ký và quản lý thông tin của họ trên Google Maps để khách hàng dễ dàng tìm thấy.

Quảng cáo địa phương: Các doanh nghiệp có thể sử dụng Google Maps để quảng cáo đến khách hàng trong khu vực lân cận.

An ninh và Ứng phó Khẩn cấp:

Dịch vụ khẩn cấp: Google Maps có thể giúp các dịch vụ khẩn cấp xác định và tiếp cận nhanh chóng các địa điểm cần trợ giúp.

Cảnh báo thời tiết và thiên tai: Cung cấp thông tin về các tình huống khẩn cấp như bão, lũ lụt, và cháy rừng.

Tích hợp với Google Cloud:

Google Maps Platform là một phần của Google Cloud, cung cấp các API và SDK cho các nhà phát triển để tích hợp chức năng bản đồ và địa lý vào các ứng dụng và dịch vụ của họ. Các API chính bao gồm:

Maps API: Cho phép hiển thị bản đồ và các tính năng tùy chỉnh trên trang web và ứng dụng di động.

Routes API: Cung cấp tính năng chỉ đường và tính toán lộ trình.

Places API: Cung cấp thông tin chi tiết về các địa điểm, bao gồm tìm kiếm, đánh giá, và ảnh.

2.7.3. Mối liên hệ giữa Google map và Google Cloud

Google Maps và Google Cloud có mối quan hệ chặt chẽ, vì Google Maps Platform (phần dịch vụ dành cho các nhà phát triển của Google Maps) được cung cấp thông qua Google Cloud. Đây là các cách Google Maps liên quan đến Google Cloud:

Google Maps Platform và Google Cloud:

Google Maps Platform:

Google Maps Platform là một bộ các API và SDK (Software Development Kits) cho phép các nhà phát triển tích hợp dịch vụ bản đồ và địa lý của Google vào các ứng dụng, trang web và dịch vụ của họ. Google Maps Platform bao gồm các dịch vụ như Maps (bản đồ), Routes (lộ trình), và Places (địa điểm).

Dịch vụ và API của Google Maps Platform:

Maps API: Cung cấp các dịch vụ hiển thị bản đồ, bao gồm các tính năng như bản đồ tĩnh, bản đồ động, chế độ xem đường phố, và các lớp bản đồ tùy chỉnh.

Routes API: Cho phép tính toán và tối ưu hóa các lộ trình, bao gồm chỉ đường cho ô tô, xe đạp, đi bộ và giao thông công cộng.

Places API: Cung cấp thông tin về địa điểm, bao gồm chi tiết về doanh nghiệp, điểm tham quan và các thông tin vị trí khác.

Tích hợp trên Google Cloud:

Quản lý API: Các API và dịch vụ của Google Maps Platform được quản lý thông qua Google Cloud Console. Người dùng có thể truy cập, quản lý và theo dõi việc sử dụng API, cũng như thiết lập quyền truy cập và bảo mật.

Thanh toán và Billing: Sử dụng Google Cloud để quản lý các chi phí liên quan đến Google Maps Platform, bao gồm theo dõi việc sử dụng và thanh toán cho các dịch vụ đã sử dụng.

Bảo mật và IAM (Identity and Access Management): Quản lý quyền truy cập vào các API của Google Maps Platform thông qua hệ thống IAM của Google Cloud, đảm bảo rằng chỉ những người dùng và dịch vụ được ủy quyền mới có thể truy cập các API và dữ liệu liên quan.

Lợi ích của việc sử dụng Google Maps Platform trên Google Cloud:

Khả năng mở rộng và độ tin cậy:

Hạ tầng của Google Cloud giúp đảm bảo rằng các dịch vụ của Google Maps Platform có thể xử lý lượng lớn yêu cầu và đảm bảo độ tin cậy cao.

Tích hợp với các dịch vụ khác của Google Cloud:

Các API của Google Maps Platform có thể được tích hợp với các dịch vụ khác của Google Cloud như BigQuery (phân tích dữ liệu lớn), Cloud Functions (xử lý sự kiện serverless), và Firebase (phát triển ứng dụng di động).

Bảo mật:

Google Cloud cung cấp các tính năng bảo mật nâng cao để bảo vệ dữ liệu và quản lý quyền truy cập vào các dịch vụ của Google Maps Platform.

Quản lý và phân tích dễ dàng:

Google Cloud Console cung cấp giao diện quản lý dễ sử dụng, giúp theo dõi và phân tích việc sử dụng các dịch vụ Google Maps Platform một cách hiệu quả.

CHƯƠNG 3: YÊU CẦU VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Phân tích hệ thống

Chuỗi cửa hàng bán hoa quả cần một hệ thống quản lý hiệu quả để theo dõi và điều hành các hoạt động kinh doanh hàng ngày. Mục tiêu của dự án là phát triển một ứng dụng web để quản lý sản phẩm, đơn hàng, khách hàng, nhân viên và báo cáo kinh doanh, giúp tăng cường hiệu quả quản lý và cải thiện trải nghiệm khách hàng.

3.2. Các chức năng chính

- Quản lý đơn hàng.
- Thêm, sửa, xóa sản phẩm.
- Đăng ký tài khoản admin.
- Xem sản phẩm.
- Giỏ hàng.
- Chi tiết sản phẩm

3.3. Chức năng dành cho khách hàng

- Chức năng đăng nhập để lưu các trạng thái mà khách hàng đã thao tác.
- Chức năng đăng ký tài khoản cho khách hàng.
- Cung cấp thông tin về sản phẩm:
 - + Hiển thị các loại trái cây có những mục đích khác nhau dành cho mọi người yêu thích.
 - + Hiển thị giá, loại sản phẩm,...
 - + Hiển thị các ảnh của trái cây.
- Chức năng tìm kiếm sản phẩm.
 - + Khách hàng có thể tìm kiếm về các loại trái cây mình yêu thích hoặc mình cần.
- Chức năng giỏ hàng.

+ Khách hàng có thể biết được những sản phẩm mà mình đã chọn, số lượng sản phẩm, loại hàng và cũng có thể thay đổi theo nhu cầu mua hàng của mình.

- Các giao diện thân thiện dễ dàng cho việc mua bán.

- Hỗ trợ trực tuyến.

+ Tư vấn thông tin kịp thời đáp ứng như cầu của khách hàng 1 cách nhanh nhất.

+ Có hotline để liên hệ.

3.4. Các chức năng cho admin

- Quản lý sản phẩm.

+ Tùy chỉnh thêm, sửa, xóa các sản phẩm.

+ Cập nhật các trạng thái sản phẩm.

- Quản lý các tài khoản.
- Tạo các tài khoản dành cho Admin.
- Quản lý loại sản phẩm.
- Quản lý đơn hàng

3.5. Mô hình phân cấp chức năng

Trong mục tiêu xây dựng một trang web chuyên về giới thiệu và bán sản phẩm sữa, em tự hào giới thiệu một mô hình phân cấp chức năng mạnh mẽ, thiết thực và linh hoạt. Mô hình này không chỉ mang lại trải nghiệm mua sắm thuận lợi và thú vị cho khách hàng mà còn giúp quản trị viên hiệu quả trong quản lý và vận hành trang web.

Dành Cho Khách Hàng:

- Đăng Nhập và Đăng Ký:
 - Khách hàng có thể trải nghiệm tốt nhất thông qua việc đăng nhập, lưu trữ trạng thái và theo dõi lịch sử mua hàng của mình. Đồng thời, chức năng đăng ký mới mang lại trải nghiệm cá nhân hóa.
- Thông Tin Sản Phẩm Chi Tiết:

- Chúng em cung cấp thông tin chi tiết và đầy đủ về các loại trái cây, giúp khách hàng dễ dàng chọn lựa.
- Tìm Kiếm Linh Hoạt:
 - Chức năng tìm kiếm nhanh chóng giúp khách hàng dễ dàng tìm thấy sản phẩm mong muốn, từ các loại quả yêu thích đến những loại trái cây cần thiết.
- Quản lý Giỏ Hàng:
 - Khách hàng có thể quản lý và điều chỉnh giỏ hàng một cách linh hoạt, kiểm tra thông tin sản phẩm, số lượng, và tổng giá trị.
- Giao Diện Thân Thiện:
 - Giao diện được thiết kế thân thiện, dễ sử dụng, mang lại trải nghiệm mua sắm trực tuyến gần gũi và thuận tiện.
- Hỗ Trợ Trực Tuyến:
 - Khách hàng luôn được hỗ trợ nhanh chóng qua các kênh trực tuyến, từ tư vấn thông tin đến việc liên hệ qua hotline.

Dành Cho Admin:

- Quản Lý Sản Phẩm:
 - Quản trị viên có thể dễ dàng thêm, sửa, xóa sản phẩm, cũng như cập nhật trạng thái của chúng.
- Quản Lý Tài Khoản:
 - Tạo và quản lý các tài khoản khách hàng và quản trị viên một cách hiệu quả.
- Tạo Tài Khoản Admin:
 - Chức năng này giúp tạo ra các tài khoản quản trị mới để phân chia trách nhiệm và quyền lực.
- Quản Lý Đơn Hàng:
 - Theo dõi và quản lý các đơn hàng một cách dễ dàng, từ xác nhận đến giao hàng.
- Quản Lý Loại Sản Phẩm:

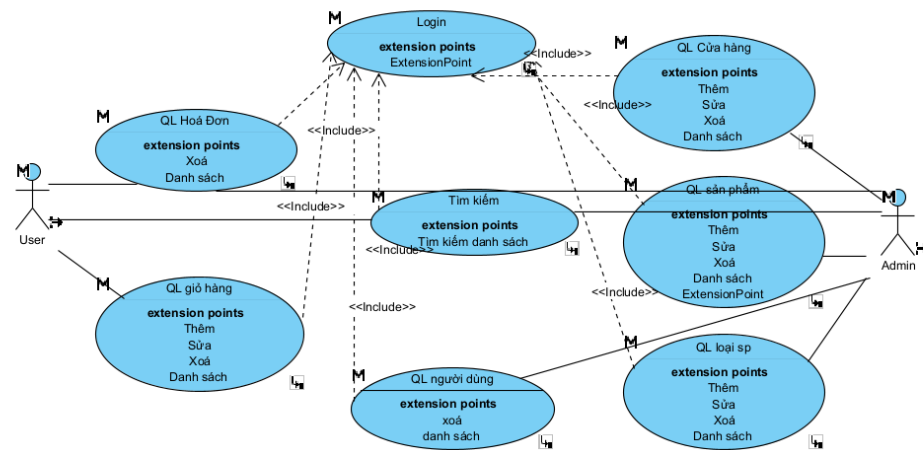
- Theo dõi và quản lý các một cách dễ dàng, Quản trị viên có thể dễ dàng thêm, sửa, xóa sản phẩm, cũng như cập nhật trạng thái của chúng.

Mô hình phân cấp chức năng của chúng em không chỉ đáp ứng nhu cầu đa dạng của khách hàng mà còn tạo điều kiện thuận lợi cho quản trị viên. Với sự linh hoạt và tính hiệu quả, chúng em tin rằng trang web sẽ trở thành điểm đến tin cậy cho những người muốn khám phá và mua sắm các loại trái cây chất lượng cao.

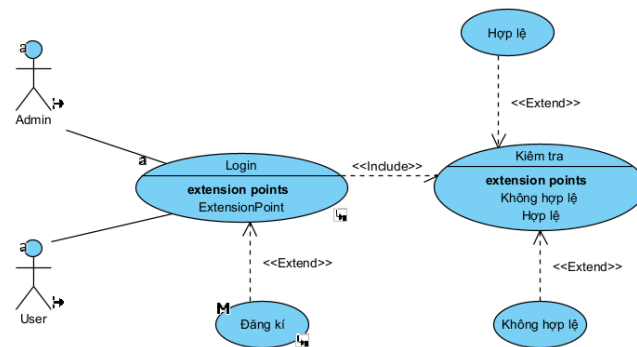
3.6 Phân tích thiết kế UML

3.6.1. Sơ đồ usecase

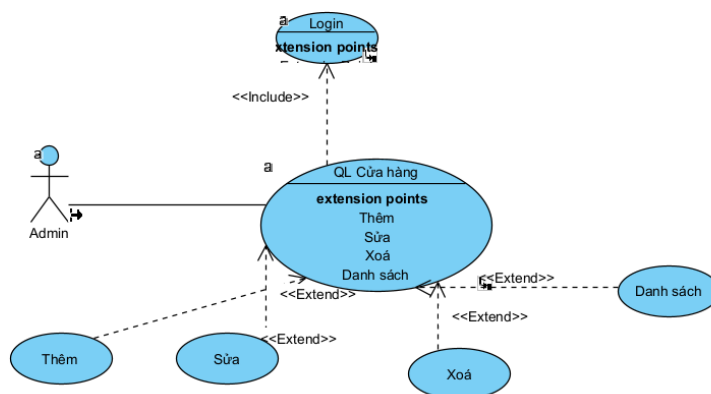
c [Tổng Quát] /



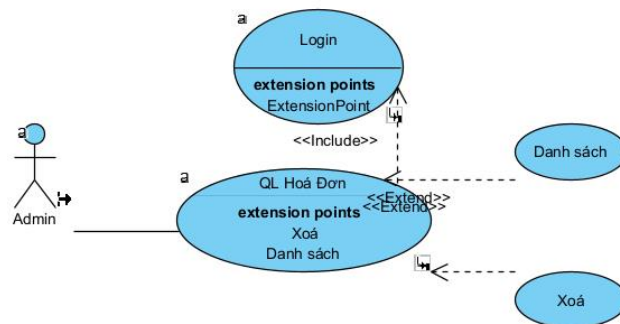
Hình 3. 1Usecase tổng quát



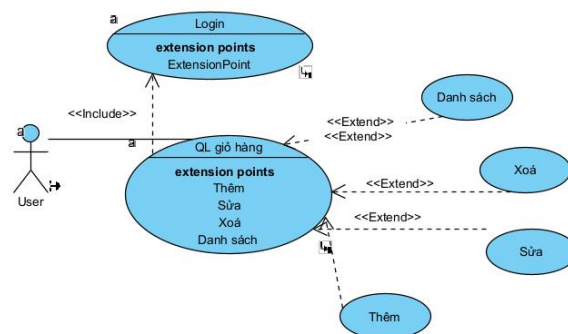
Hình 3. 2 Usecase Login



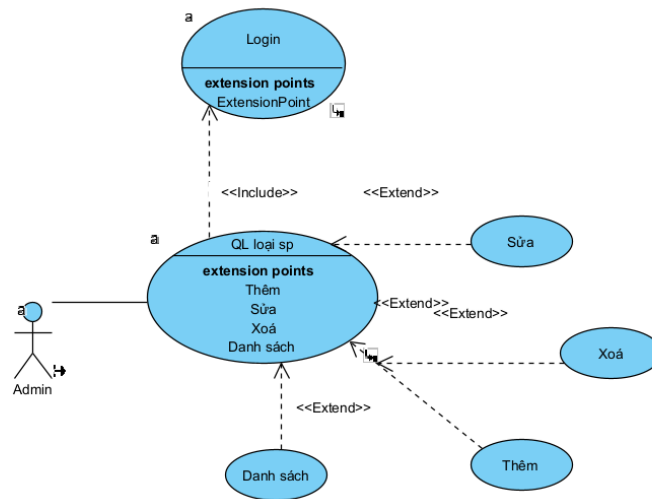
Hình 3. 3 Usecase quản lý cửa hàng



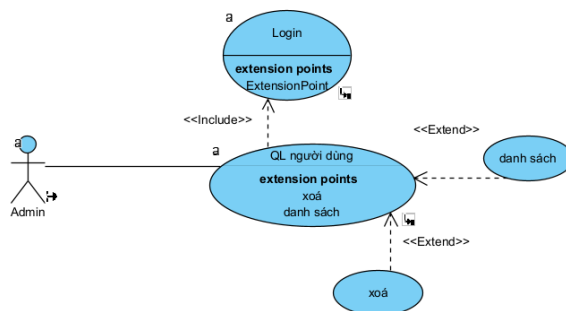
Hình 3. 4 Usecase Quản lý hoá đơn



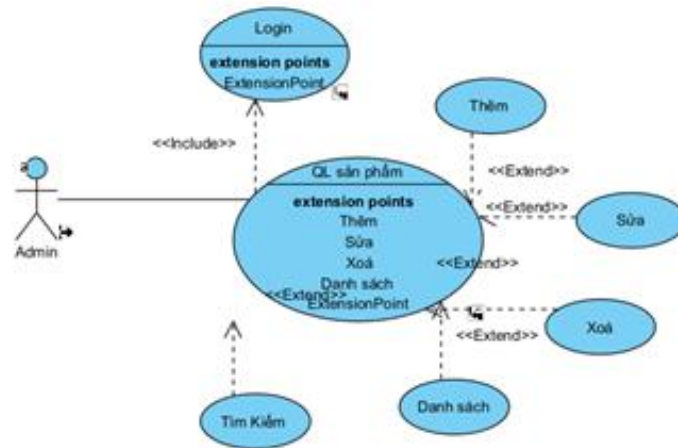
Hình 3. 5 Usecase quản lý giỏ hàng



Hình 3. 6 Usecase quản lý loại sản phẩm

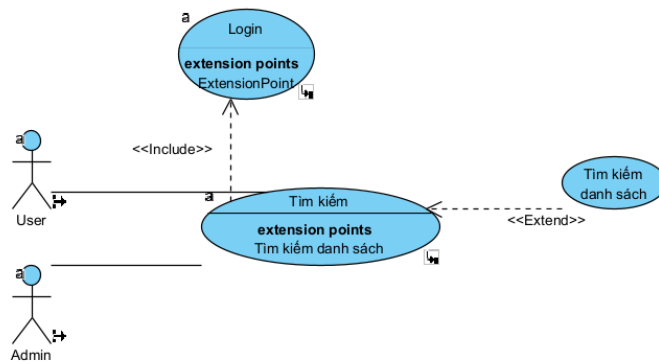


Hình 3. 7 Usecase quản lý người dùng

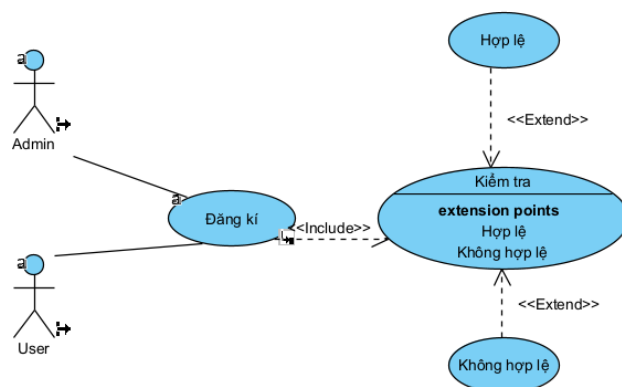


Hình 3. 8 Usecase quản lý sản phẩm

ic [Tìm kiếm]



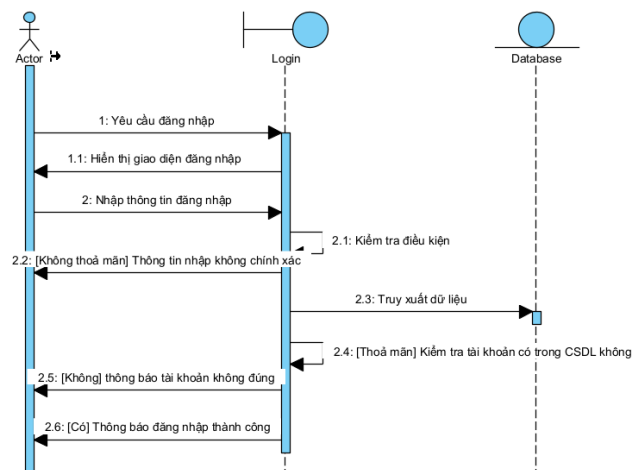
Hình 3. 9 Usecase tìm kiếm



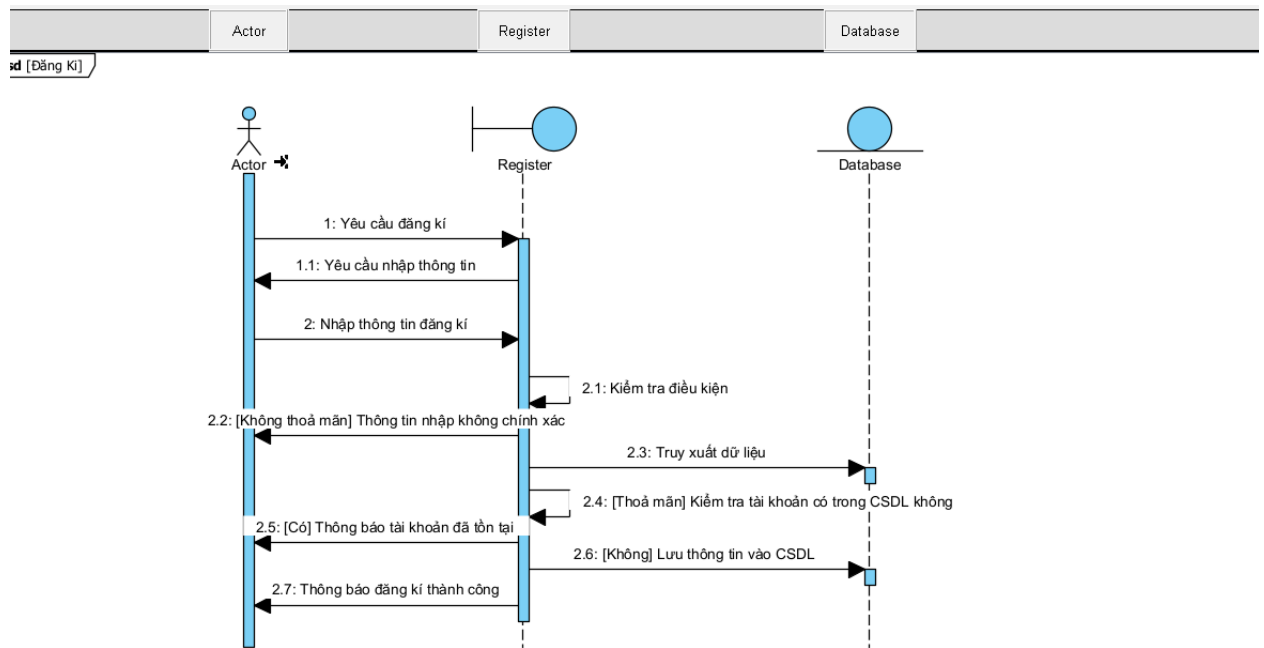
Hình 3. 10 Usecase đăng ký

3.6.2. Biểu đồ tuần tự

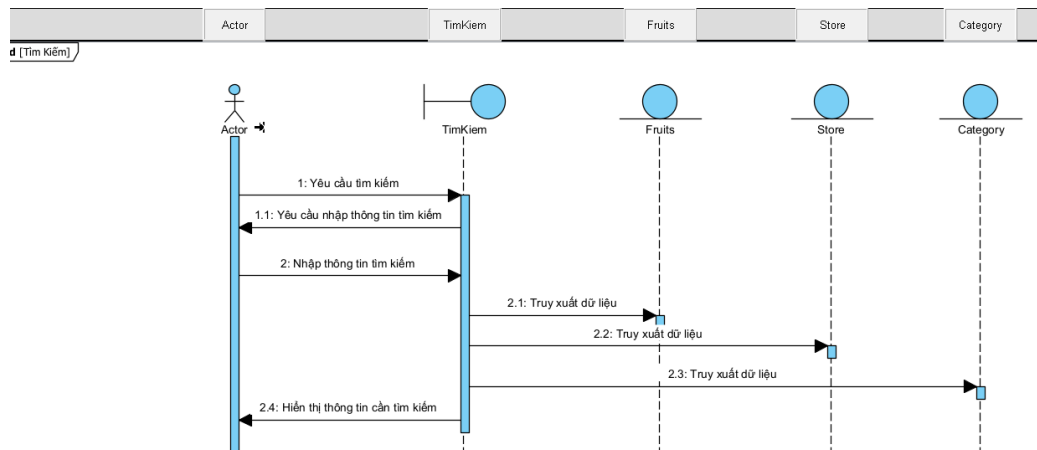
[Đăng nhập]



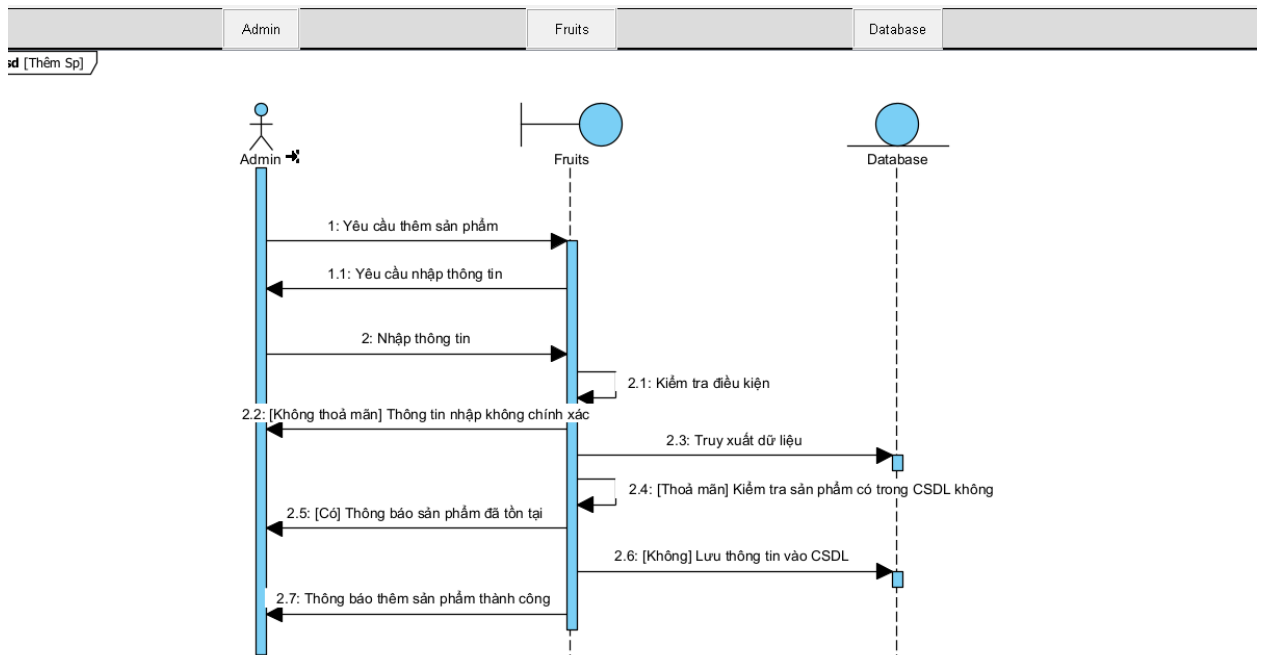
Hình 3. 11 Biểu đồ tuần tự đăng nhập



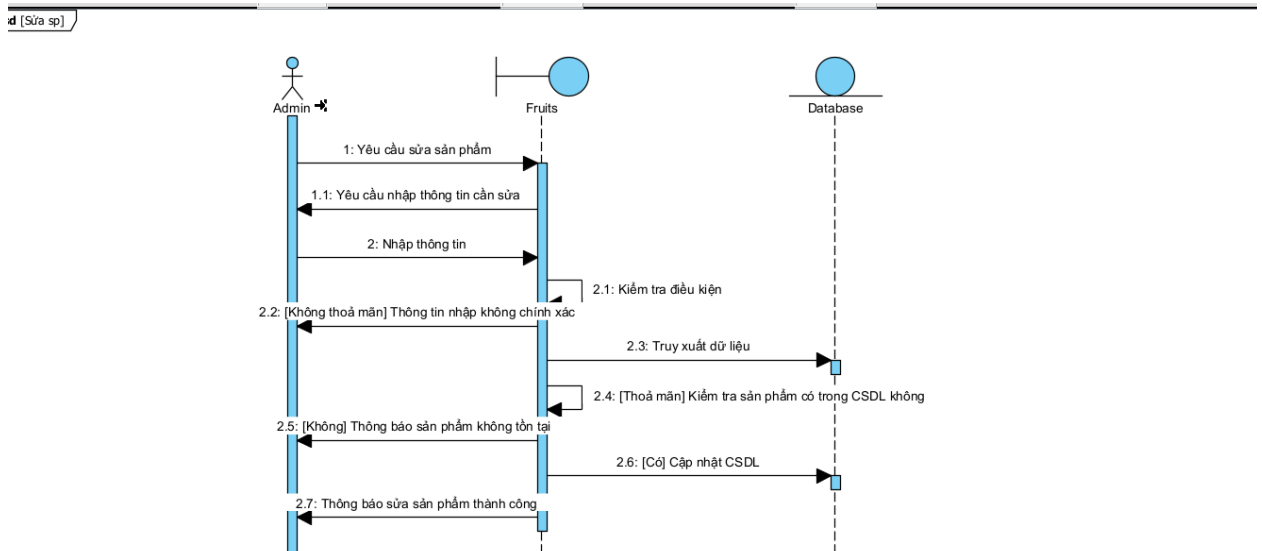
Hình 3. 12 Biểu đồ tuần tự đăng ký



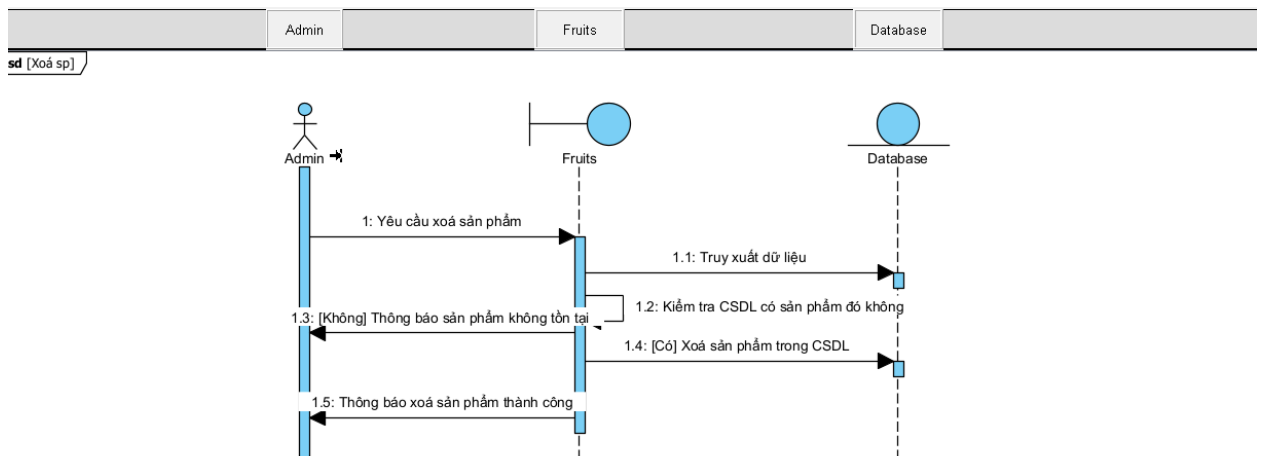
Hình 3. 13 Biểu đồ tuần tự tìm kiếm



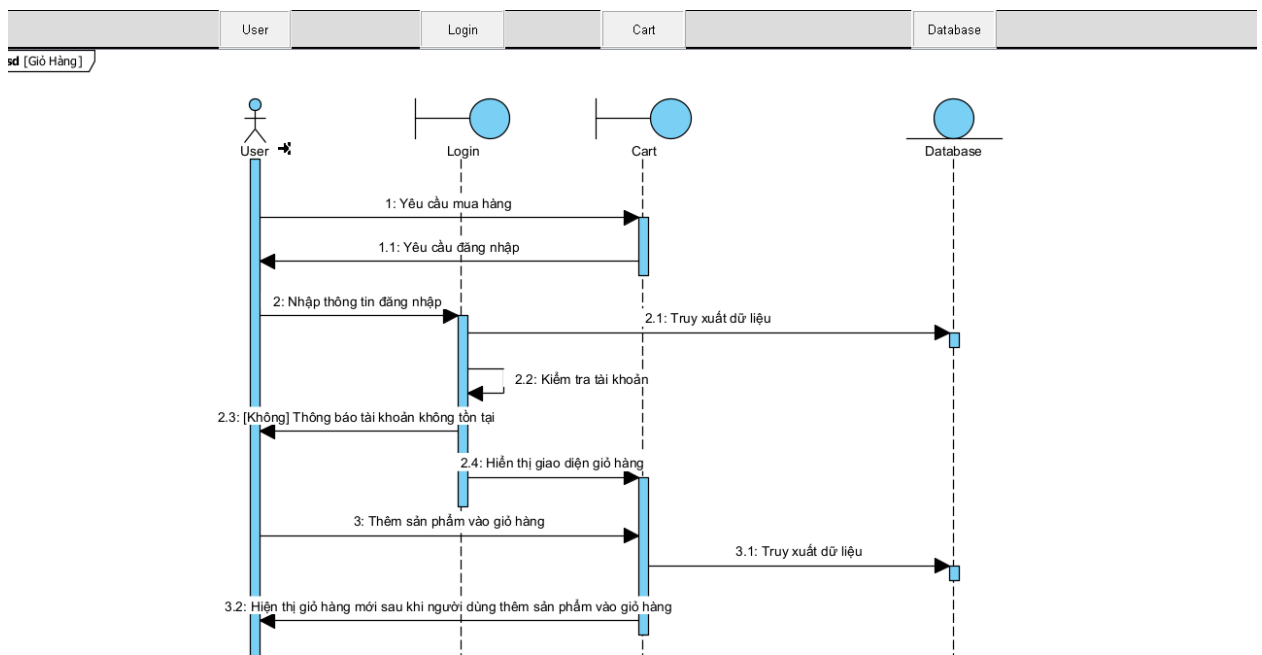
Hình 3. 14 Biểu đồ tuần tự thêm sản phẩm



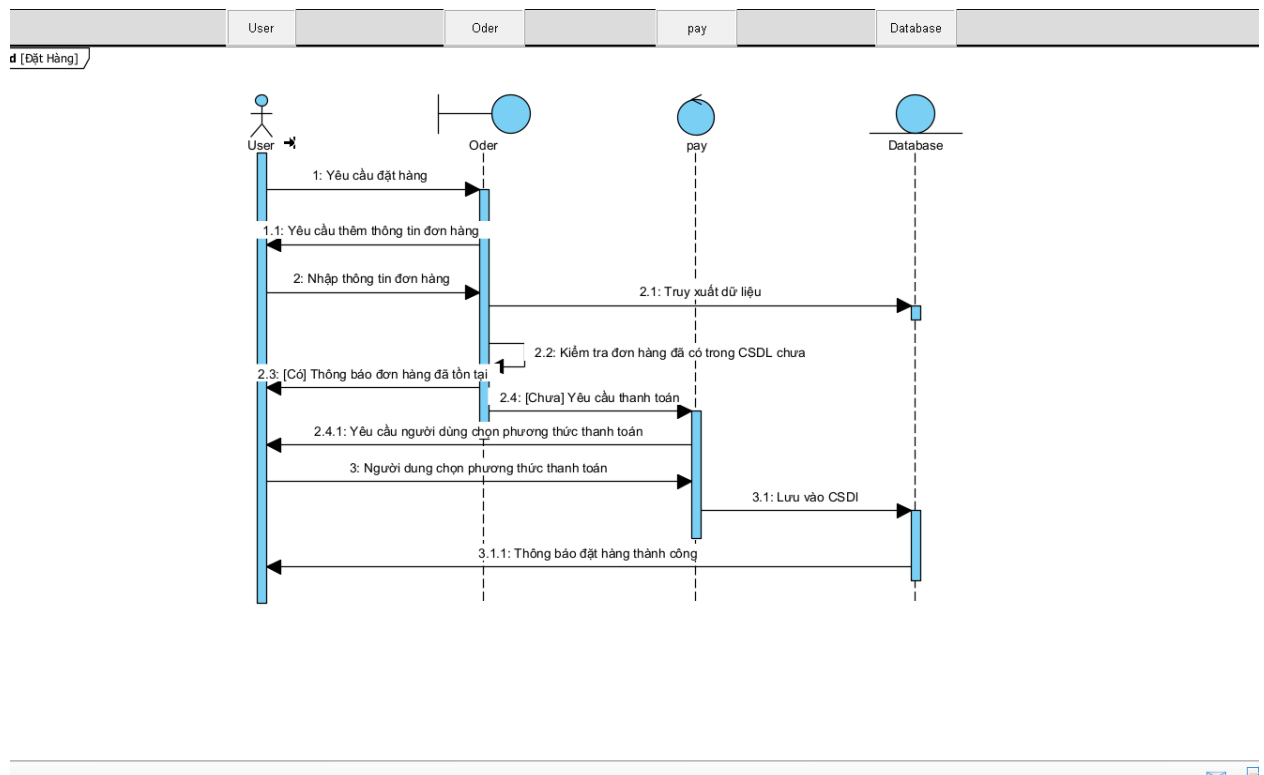
Hình 3. 15 Biểu đồ tuần tự sửa sản phẩm



Hình 3. 16 Biểu đồ tuần tự xoá sản phẩm

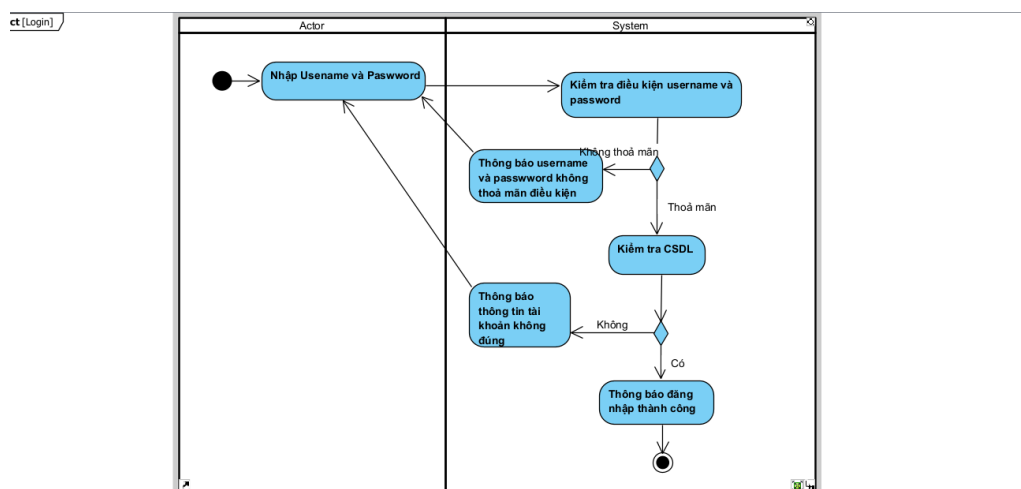


Hình 3. 17 Biểu đồ tuần tự giỏ hàng

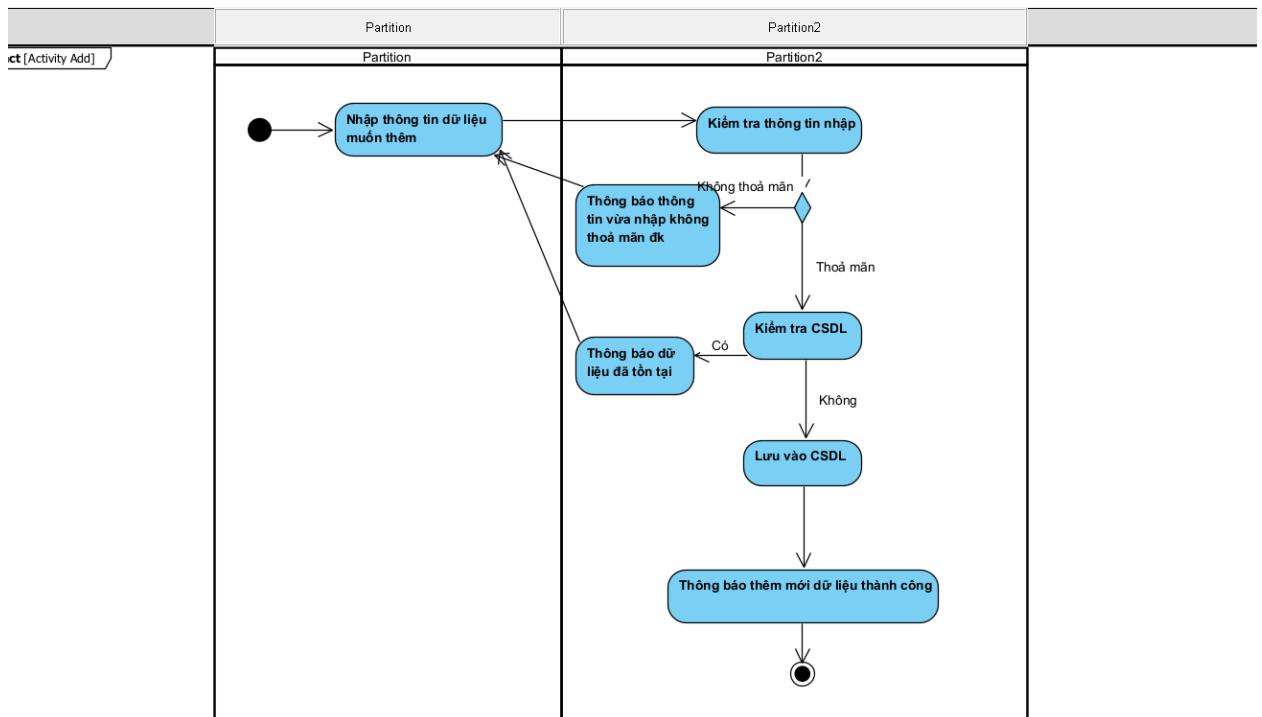


Hình 3. 18 Biểu đồ tuần tự giỏ hàng

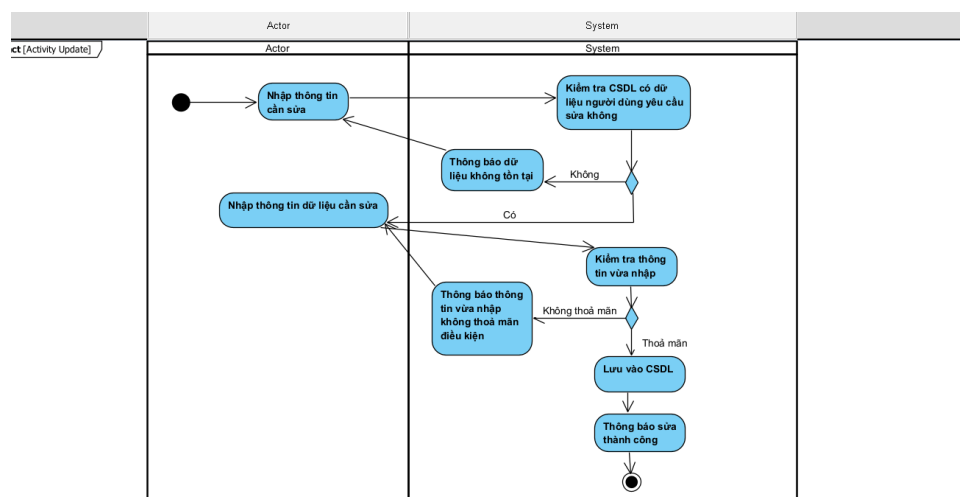
3.6.3. Biểu đồ hoạt động



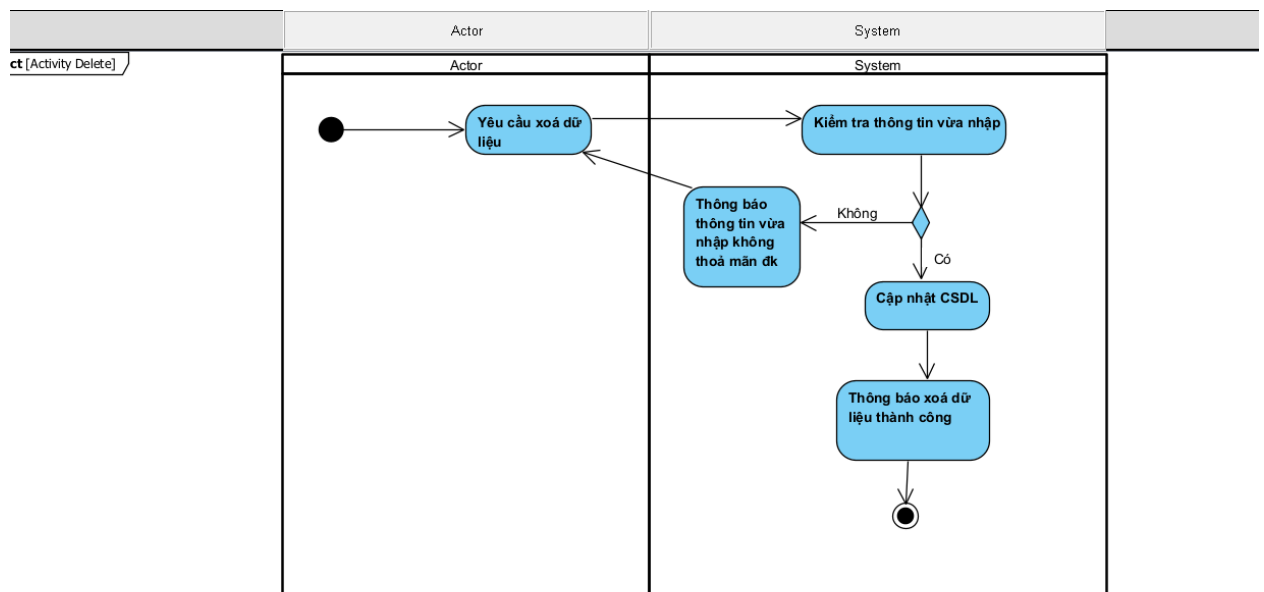
Hình 3. 19 Biểu đồ hoạt động Login



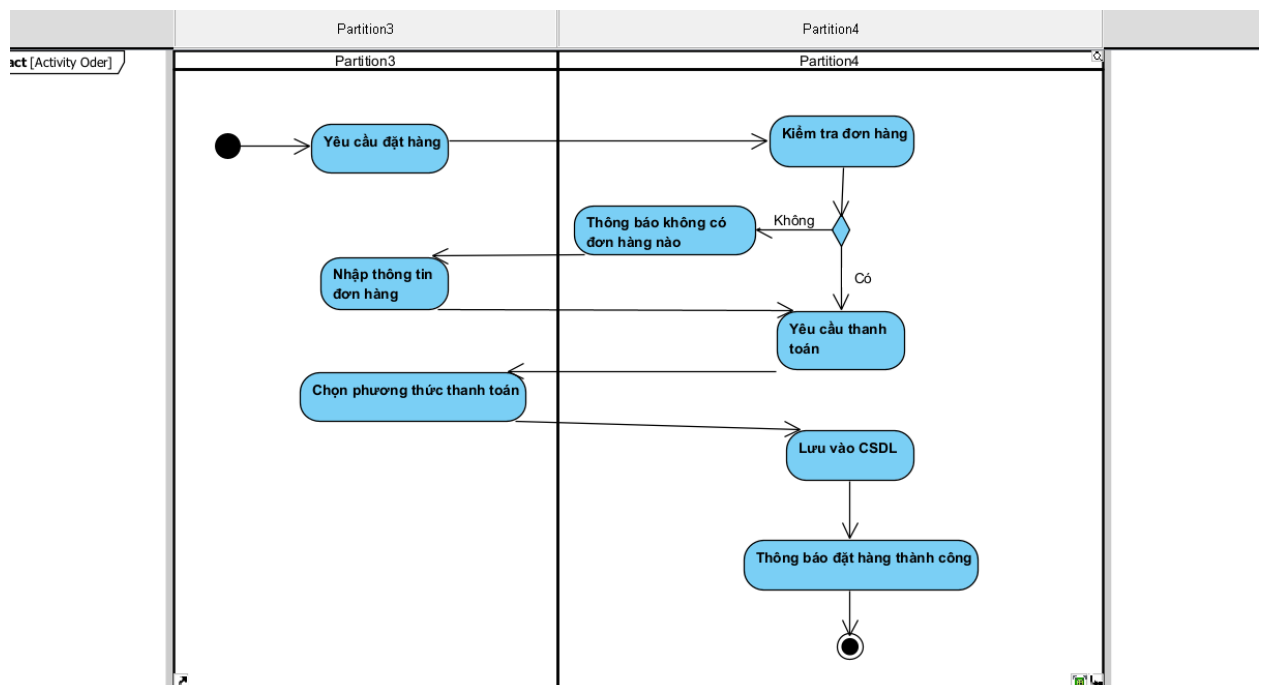
Hình 3. 20 Biểu đồ hoạt động thêm dữ liệu



Hình 3. 21 Biểu đồ hoạt động sửa dữ liệu



Hình 3. 22 Biểu đồ hoạt động xoá dữ liệu



Hình 3. 23 Biểu đồ hoạt động đặt hàng

3.7. Thiết kế cơ sở dữ liệu

3.7.1. Thiết kế bảng DATABASE

Tên cột	Kiểu dữ liệu	Mô tả	Chú thích
Bill_Detail_Id	bigint	ID	Khóa chính
BillId	bigint		
FruitId	bigint		
Quantity	bigint		
StoreId	bigint		
createdBy	bigint		
createdAt	datetime		
updatedBy	bigint		
updatedAt	datetime2(7)		
deletedBy	bigint		
deletedAt	datetime2(7)		
IsDelete	bit		

Bảng 3. 1Bill_Details

Tên cột	Kiểu dữ liệu	Mô tả	Chú thích
MigrationId	nvarchar(150)	ID	Khóa chính
ProductVersion	nvarchar(32)		

Bảng 3. 2 EFMigrationsHistory

Tên cột	Kiểu dữ liệu	Mô tả	Chú thích
BillId	bigint	ID	Khóa chính
FullName	nvarchar(MAX)		
Address	nvarchar(MAX)		
Total_amount	nvarchar(MAX)		
Payments	nvarchar(MAX)		
Phone	nvarchar(MAX)		
UserId	bigint		
Bill_Status	int		
createdBy	bigint		
createdAt	datetime2(7)		
updatedBy	bigint		
updatedAt	datetime2(7)		
deletedBy	bigint		
deletedAt	datetime2(7)		
IsDelete	bit		

Bảng 3. 3 Bills

Tên cột	Kiểu dữ liệu	Mô tả	Chú thích
CartId	bigint	ID	Khóa chính
FruitId	bigint		
Quantity	bigint		
UserId	bigint		
StoreId	bigint		
createdBy	bigint		
createdAt	datetime2(7)		
updatedBy	bigint		
updatedAt	datetime2(7)		
deletedBy	bigint		
deletedAt	datetime2(7)		
IsDelete	bit		

Bảng 3. 4 Cart

Tên cột	Kiểu dữ liệu	Mô tả	Chú thích
CategoriesId	bigint	ID	Khóa chính
createdBy	bigint		
createdAt	datetime2(7)		
updatedBy	bigint		
updatedAt	datetime2(7)		
deletedBy	bigint		
deletedAt	datetime2(7)		
IsDelete	bit		

Bảng 3. 5 Categories

Tên cột	Kiểu dữ liệu	Mô tả	Chú thích
FruitId	bigint	ID	Khóa chính
FruitName	nvarchar(MAX)		
FruitDescription	nvarchar(MAX)		
FruitQuantity	nvarchar(MAX)		
FruitPrice	nvarchar(MAX)		
CategoriesId	bigint		
Discount	nvarchar(MAX)		
PriceDiscount	nvarchar(MAX)		
FruitImg	nvarchar(MAX)		
StoreId	bigint		
createdBy	bigint		
createdAt	datetime2(7)		
updatedBy	bigint		
updatedAt	datetime2(7)		
deletedBy	bigint		

Bảng 3. 6 Fruits

Tên cột	Kiểu dữ liệu	Mô tả	Chú thích
UserId	bigint	ID	Khóa chính
Refresh_TokenExpires	datetime2(7)		
RefreshToken	nvarchar(MAX)		
RefreshTokenExpiration	int		

createdBy	bigint		
createdAt	datetime2(7)		
updatedBy	bigint		
updatedAt	datetime2(7)		
deletedBy	bigint		
deletedAt	datetime2(7)		
IsDelete	bit		

Bảng 3. 7 RefreshTokens

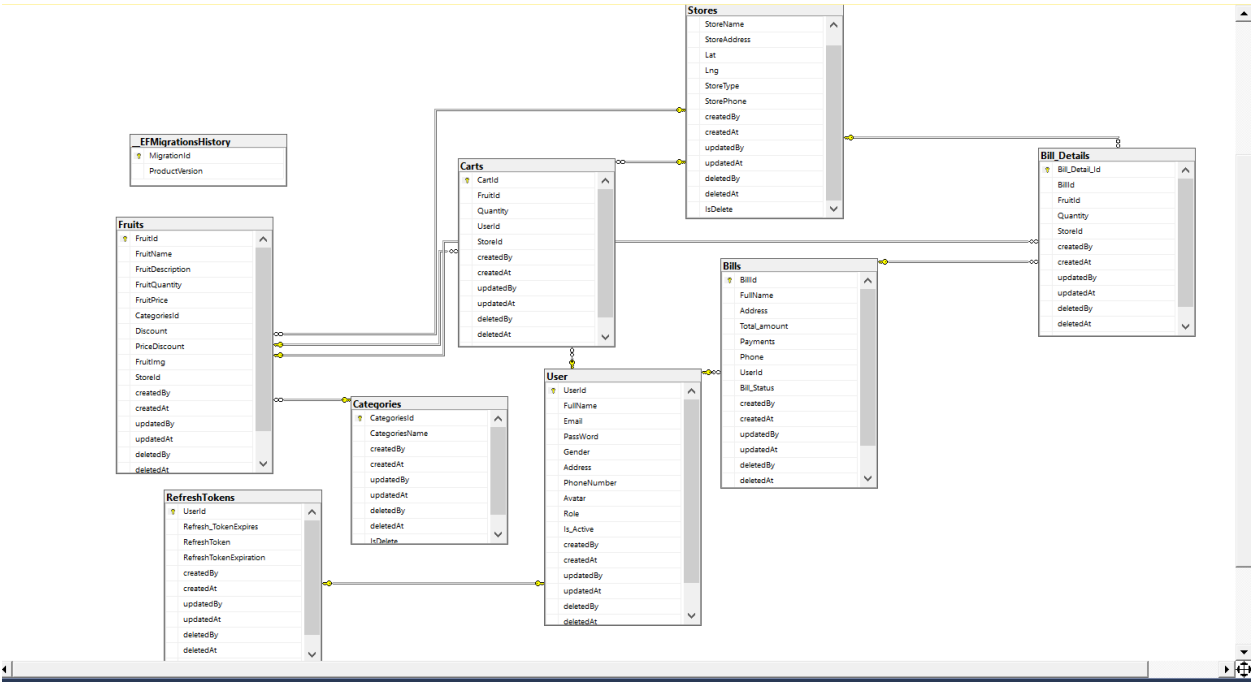
Tên cột	Kiểu dữ liệu	Mô tả	Chú thích
StoreId	bigint	ID	Khóa chính
StoreName	nvarchar(MAX)		
StoreAddress	nvarchar(MAX)		
Lat	nvarchar(MAX)		
Lng	nvarchar(MAX)		
StoreType	nvarchar(MAX)		
StorePhone	nvarchar(MAX)		
createdAt	datetime2(7)		
updatedBy	bigint		
updatedAt	datetime2(7)		
deletedBy	bigint		
deletedAt	datetime2(7)		
IsDelete	bit		

Bảng 3. 8 Stores

Tên cột	Kiểu dữ liệu	Mô tả	Chú thích
UserId	bigint	ID	Khoá chính
FullName	nvarchar(MAX)		
Email	nvarchar(MAX)		
PassWord	nvarchar(MAX)		
Gender	nvarchar(MAX)		
Address	nvarchar(MAX)		
PhoneNumber	nvarchar(MAX)		
Avatar	nvarchar(MAX)		
Role	nvarchar(MAX)		
Is_Active	bit		
createdAt	datetime2(7)		
updatedBy	bigint		
updatedAt	datetime2(7)		
deletedBy	bigint		
deletedAt	datetime2(7)		
IsDelete	bit		

Bảng 3. 9 User

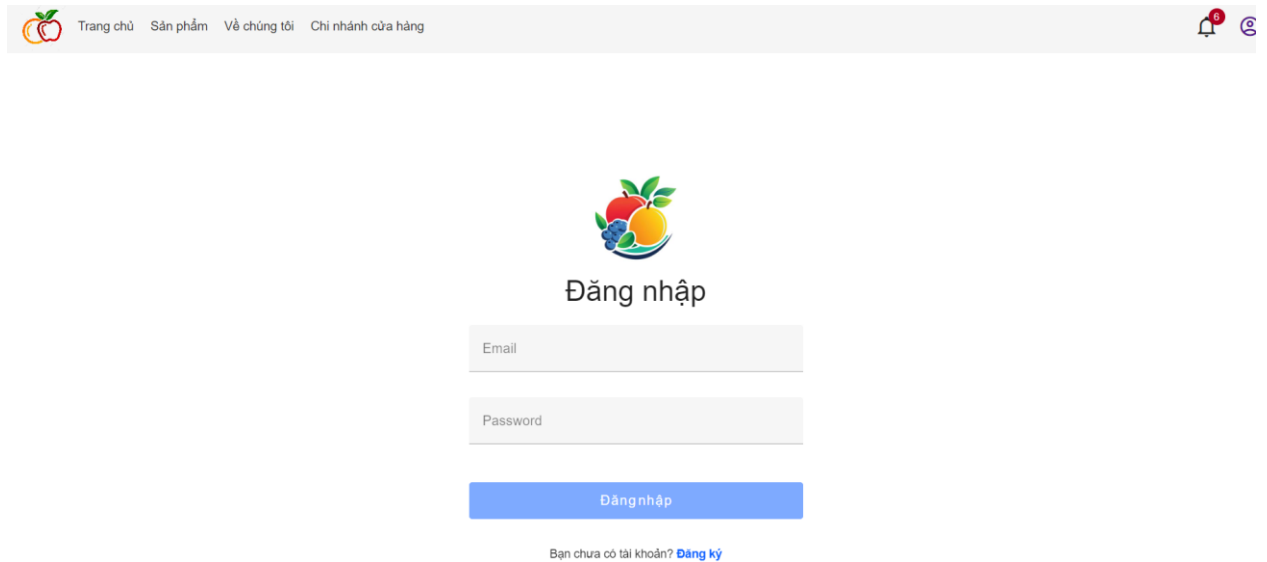
3.7.2. Sơ đồ DIAGRAMS



Hình 3. 24 Sơ đồ Diagram

CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC

4.1. Trang đăng nhập



Trang chủ Sản phẩm Về chúng tôi Chi nhánh cửa hàng

Đăng nhập

Email

Password

Đăng nhập

Bạn chưa có tài khoản? [Đăng ký](#)

Hình 4. 1 Trang đăng nhập

Form Đăng nhập

Tên người dùng:

Label: "Tên người dùng" hoặc "Email"

Input field: Trường nhập liệu để người dùng nhập tên đăng nhập hoặc email. Trường này sẽ có placeholder là "Email".

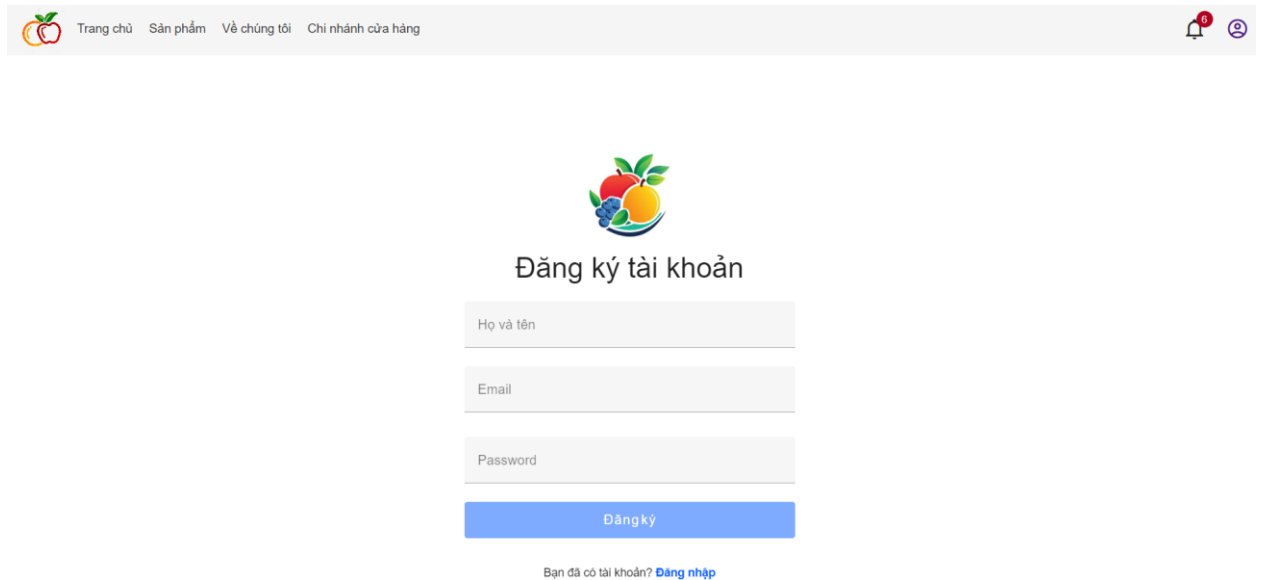
Mật khẩu:

Label: "Mật khẩu"

Nút Đăng nhập:

Nút chính: Nút "Đăng nhập" được đặt dưới các trường nhập liệu, có màu sắc nổi bật để dễ nhận diện và khuyến khích người dùng nhấn vào. Phong chữ trên nút rõ ràng và dễ đọc.

4.2. Trang đăng ký



Trang chủ Sản phẩm Về chúng tôi Chi nhánh cửa hàng

Đăng ký tài khoản

Họ và tên

Email

Password

Đăng ký

[Bạn đã có tài khoản? Đăng nhập](#)

Hình 4. 2 Trang đăng ký

Form Đăng ký

Họ và Tên:

Label: "Họ và Tên"

Input field: Trường nhập liệu cho họ và tên. Placeholder là "Nhập họ và tên".

Email:

Label: "Email"

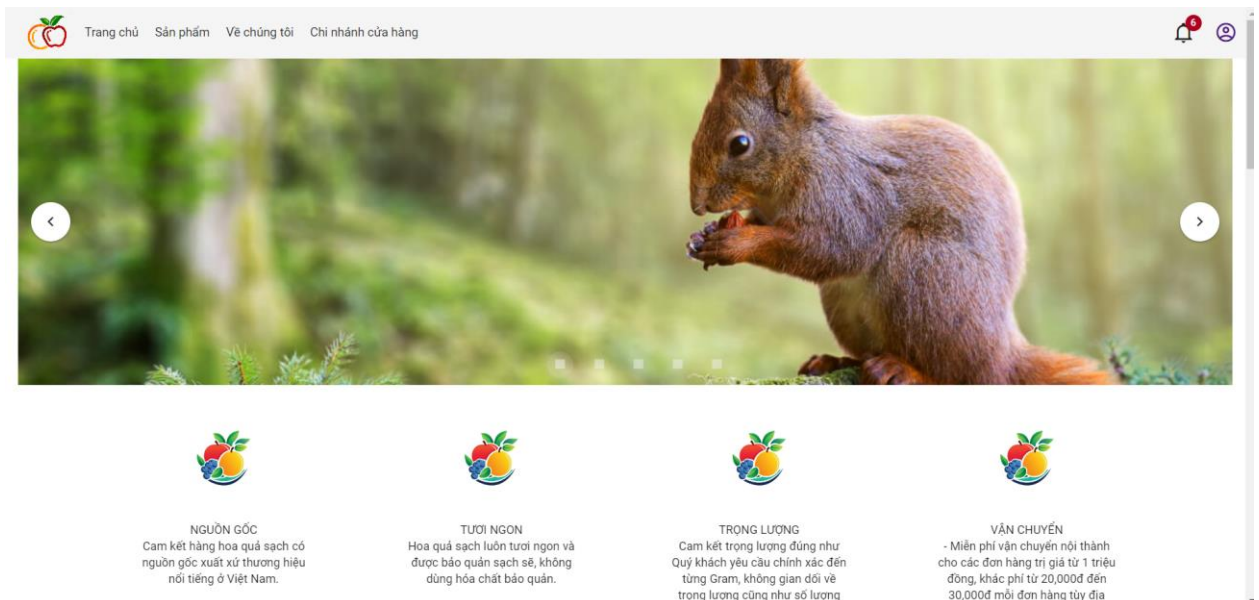
Input field: Trường nhập liệu email. Placeholder là "Nhập email".

Mật khẩu:

Label: "Mật khẩu"

Input field: Trường nhập liệu mật khẩu với placeholder là "Nhập mật khẩu" và biểu tượng mắt để hiện/ẩn mật khẩu.

4.3. Trang chủ



Hình 4. 3 Trang chủ

Trang chủ là bộ mặt của ứng dụng, nơi người dùng có thể tìm thấy thông tin tổng quan về chuỗi cửa hàng, các chi nhánh và các sản phẩm nổi bật. Dưới đây là mô tả chi tiết về giao diện trang chủ cho ứng dụng chuỗi cửa hàng bán hoa quả:

1. Bố cục chung

Header: Phần trên cùng của trang chứa logo, thanh điều hướng (navigation bar), và nút đăng nhập/đăng ký.

Hero Section: Phần giới thiệu nổi bật với hình ảnh hoa quả tươi ngon và một thông điệp chào mừng.

Nội dung chính: Bao gồm các phần như Giới thiệu về chúng em, Các chi nhánh cửa hàng, và Sản phẩm nổi bật.

Footer: Phần dưới cùng chứa thông tin liên hệ, liên kết mạng xã hội, và các liên kết quan trọng khác.

2. Header

Logo: Logo của cửa hàng đặt ở góc trái trên cùng.

Thanh điều hướng: Bao gồm các liên kết đến các trang quan trọng như Trang chủ, Về chúng em, Chi nhánh, Sản phẩm, Liên hệ.

Nút đăng nhập/đăng ký: Đặt ở góc phải trên cùng, giúp người dùng dễ dàng truy cập vào tài khoản của mình.

3. Hero Section

Hình ảnh nền: Hình ảnh hoa quả tươi ngon, sống động.

4. Nội dung chính

a. Giới thiệu về chúng em

Tiêu đề: "Về Chúng Em"

Văn bản giới thiệu: Mô tả ngắn gọn về lịch sử, sứ mệnh và giá trị của chuỗi cửa hàng.

Hình ảnh minh họa: Hình ảnh các cửa hàng, nhân viên, hoặc các hoạt động nổi bật.

b. Các chi nhánh cửa hàng

Tiêu đề: "Chi nhánh của Chúng Em"

Danh sách chi nhánh: Thông tin về các chi nhánh, bao gồm địa chỉ, số điện thoại, giờ mở cửa, và hình ảnh của từng chi nhánh.

Bản đồ: Bản đồ tích hợp hiển thị vị trí các chi nhánh, giúp người dùng dễ dàng tìm kiếm chi nhánh gần nhất.

c. Sản phẩm nổi bật

Tiêu đề: "Sản phẩm Nổi bật"

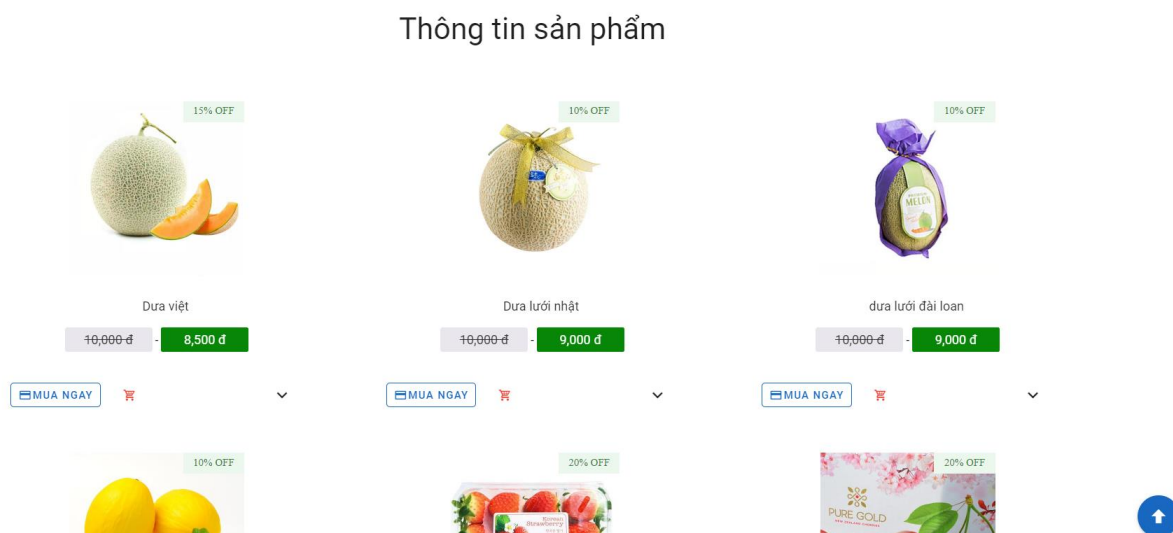
Danh sách sản phẩm: Hình ảnh và mô tả ngắn về các sản phẩm hoa quả được yêu thích nhất.

Nút mua ngay: Nút "Mua ngay" hoặc "Xem chi tiết" dẫn đến trang chi tiết sản phẩm.

5. Footer

Thông tin liên hệ: Địa chỉ, số điện thoại, email liên hệ.

4.4. Trang sản phẩm



Hình 4. 4 Trang sản phẩm

1. Bố cục chung

Header: Phần trên cùng của trang chứa logo, thanh điều hướng (navigation bar), và nút đăng nhập/đăng ký.

Nội dung chính: Bao gồm thông tin chi tiết về sản phẩm, hình ảnh, mô tả, giá, đánh giá của khách hàng và nút mua hàng.

Footer: Phần dưới cùng chứa thông tin liên hệ, liên kết mạng xã hội, và các liên kết quan trọng khác.

2. Header

Logo: Logo của cửa hàng đặt ở góc trái trên cùng.

Thanh điều hướng: Bao gồm các liên kết đến các trang quan trọng như Trang chủ, Về chúng em, Chi nhánh, Sản phẩm, Liên hệ.

Nút đăng nhập/đăng ký: Đặt ở góc phải trên cùng, giúp người dùng dễ dàng truy cập vào tài khoản của mình.

3. Nội dung chính

a. Hình ảnh sản phẩm

Hình ảnh chính: Hình ảnh lớn, chất lượng cao của sản phẩm.

b. Thông tin sản phẩm

Tên sản phẩm: Tiêu đề lớn, rõ ràng.

Mô tả ngắn: Đoạn văn ngắn gọn mô tả về sản phẩm, bao gồm các đặc điểm nổi bật.

Giá: Hiển thị giá của sản phẩm, có thể có giá khuyến mãi nếu có.

c. Nút mua hàng

Nút chính: Nút "Mua ngay" hoặc "Thêm vào giỏ hàng" màu sắc nổi bật, dễ nhìn.

Số lượng: Tùy chọn để người dùng chọn số lượng sản phẩm muốn mua.

d. Chi tiết bổ sung

Mô tả chi tiết: Phần mô tả chi tiết về sản phẩm, công dụng, lợi ích, cách sử dụng.

Đánh giá khách hàng: Khu vực hiển thị đánh giá và bình luận từ khách hàng, giúp người dùng khác tham khảo.

Nút đăng nhập/đăng ký: Đặt ở góc phải trên cùng, giúp người dùng dễ dàng truy cập vào tài khoản của mình.

3. Google Map

Bản đồ Google: Google Map được tích hợp trực tiếp vào trang, hiển thị toàn bộ các chi nhánh của chuỗi cửa hàng dưới dạng các điểm đánh dấu (markers).

Điểm đánh dấu (Markers): Các điểm đánh dấu đại diện cho các chi nhánh, khi người dùng nhấn vào sẽ hiển thị thông tin chi tiết về chi nhánh đó.

Thông tin chi nhánh: Khi nhấn vào điểm đánh dấu, một cửa sổ thông tin (info window) xuất hiện với tên chi nhánh, địa chỉ, số điện thoại, giờ mở cửa và liên kết đến trang chi tiết chi nhánh.

4. Danh sách chi nhánh

Danh sách dạng lưới hoặc danh sách: Danh sách các chi nhánh được hiển thị dưới dạng lưới (grid) hoặc danh sách (list) bên cạnh hoặc dưới bản đồ.

Thông tin chi tiết: Mỗi mục trong danh sách bao gồm tên chi nhánh, địa chỉ, số điện thoại, giờ mở cửa và một nút "Xem trên bản đồ" hoặc "Chi tiết" để người dùng dễ dàng xác định vị trí trên bản đồ hoặc xem thông tin chi tiết hơn.

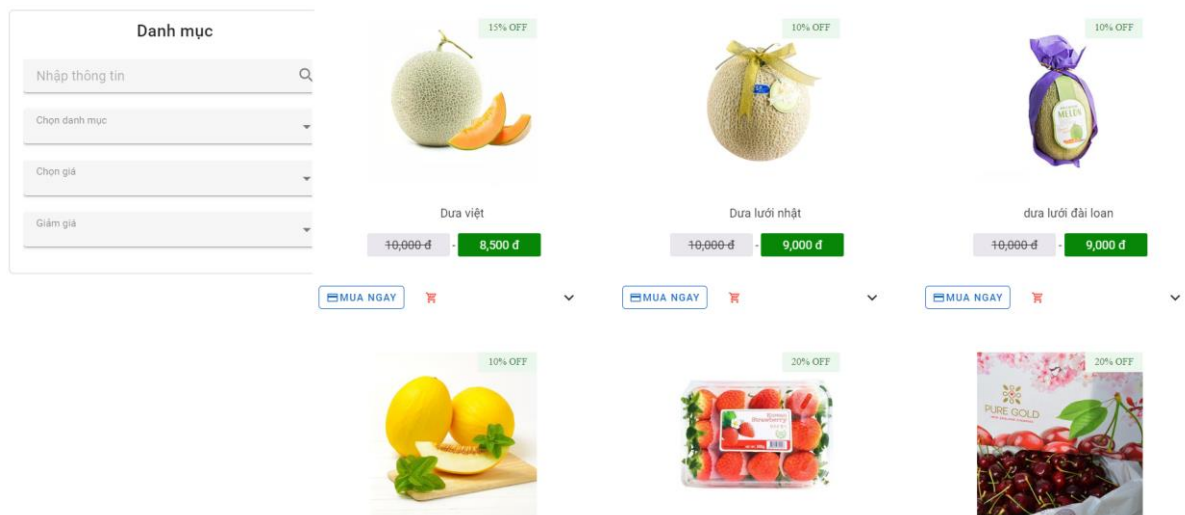
5. Footer

Thông tin liên hệ: Địa chỉ, số điện thoại, email liên hệ.

Liên kết mạng xã hội: Biểu tượng và liên kết đến các trang mạng xã hội như Facebook, Instagram, Twitter.

Các liên kết quan trọng: Liên kết đến các trang như Chính sách bảo mật, Điều khoản sử dụng, Hỗ trợ khách hàng.

4.6. Danh mục sản phẩm



Hình 4. 6 Trang danh mục sản phẩm

Header: Phần trên cùng của trang chứa logo, thanh điều hướng (navigation bar), và nút đăng nhập/đăng ký.

Bộ lọc và tìm kiếm: Bao gồm các bộ lọc để chọn danh mục, nhập thông tin sản phẩm và chọn khoảng giá.

Danh sách sản phẩm: Hiển thị các sản phẩm tương ứng với bộ lọc đã chọn.

Footer: Phần dưới cùng chứa thông tin liên hệ, liên kết mạng xã hội, và các liên kết quan trọng khác.

Logo: Logo của cửa hàng đặt ở góc trái trên cùng.

Thanh điều hướng: Bao gồm các liên kết đến các trang quan trọng như Trang chủ, Về chúng em, Chi nhánh, Sản phẩm, Liên hệ.

Nút đăng nhập/đăng ký: Đặt ở góc phải trên cùng, giúp người dùng dễ dàng truy cập vào tài khoản của mình.

a. Thanh tìm kiếm

Ô tìm kiếm: Ô nhập liệu để người dùng nhập tên sản phẩm hoặc từ khóa liên quan.

Nút tìm kiếm: Nút "Tìm kiếm" để kích hoạt tìm kiếm sản phẩm theo từ khóa đã nhập.

b. Chọn danh mục

Danh sách danh mục: Menu thả xuống hoặc danh sách các danh mục sản phẩm, ví dụ: Hoa quả tươi, Hoa quả sấy, Nước ép trái cây, v.v.

c. Chọn khoảng giá

Thanh trượt giá: Thanh trượt cho phép người dùng chọn khoảng giá từ thấp nhất đến cao nhất.

Ô nhập liệu giá: Hai ô nhập liệu cho phép người dùng nhập trực tiếp khoảng giá mong muốn (giá từ, giá đến).

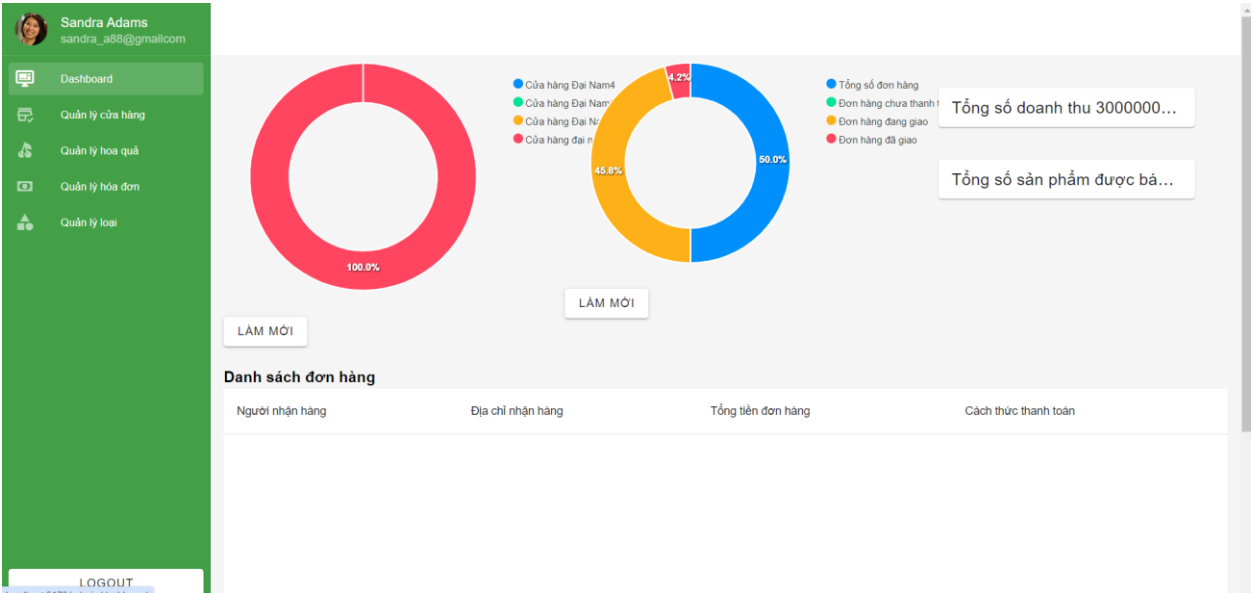
Hiển thị sản phẩm: Các sản phẩm được hiển thị dưới dạng lưới (grid) hoặc danh sách (list) với hình ảnh sản phẩm, tên sản phẩm, giá, và nút "Thêm vào giỏ hàng" hoặc "Chi tiết".

Phân trang: Nếu có nhiều sản phẩm, trang sẽ bao gồm các nút phân trang (pagination) hoặc tính năng cuộn vô hạn (infinite scroll) để tải thêm sản phẩm.

Thông tin liên hệ: Địa chỉ, số điện thoại, email liên hệ.

Liên kết mạng xã hội: Biểu tượng và liên kết đến các trang mạng xã hội như Facebook, Instagram, Twitter.

4.7. Trang Dashboard



Hình 4. 7 Trang Dashboard

Biểu đồ thống kê: Hiển thị biểu đồ thông kê tổng doanh thu trong một khoảng thời gian cụ thể.

Danh sách đơn hàng: Danh sách các đơn hàng bán được, bao gồm thông tin chi tiết về mỗi đơn hàng.

Phần dưới cùng chứa thông tin liên hệ và các liên kết quan trọng khác.

Logo: Logo của cửa hàng đặt ở góc trái trên cùng.

Thanh điều hướng: Bao gồm các liên kết đến các trang quan trọng như Quản lý cửa hàng, Quản lý hoa quả, Quản lý hóa đơn, Quản lý loại.

Biểu đồ thống kê

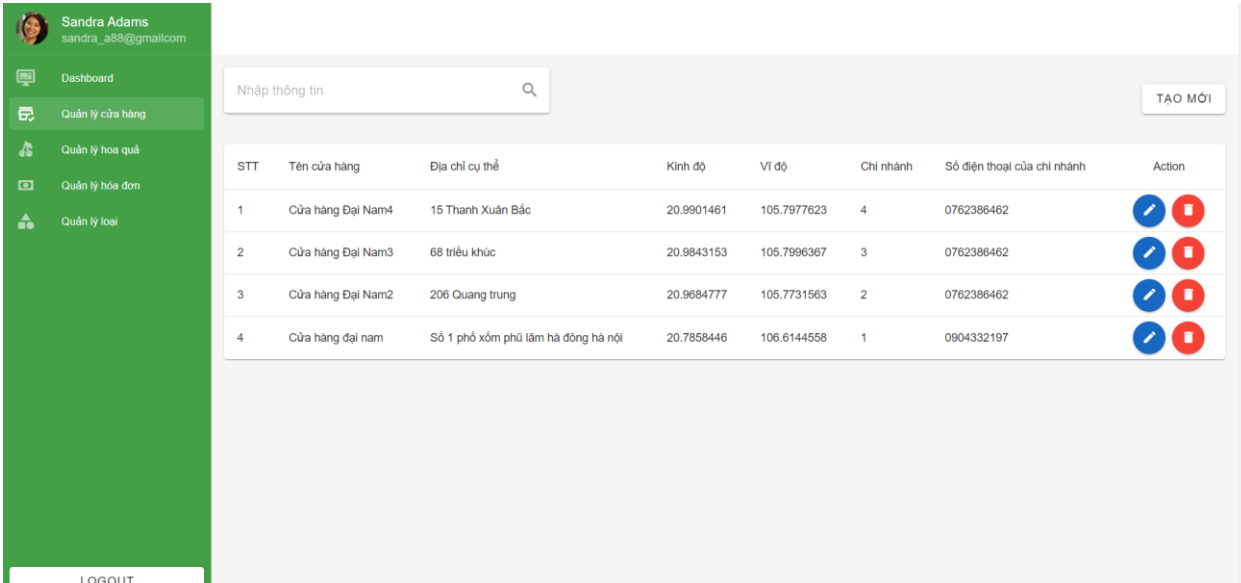
Biểu đồ tổng doanh thu: Biểu đồ dạng đường hoặc cột thể hiện tổng doanh thu của cửa hàng.

Danh sách đơn hàng: Hiện thị các đơn hàng bán được trong một khoảng thời gian cụ thể.









Thông tin đơn hàng: Mỗi mục trong danh sách bao gồm thông tin về đơn hàng như số đơn hàng, người nhận hàng, địa chỉ nhận hàng, tổng tiền đơn hàng và cách thức thanh toán.

Nút xem chi tiết: Nút cho phép người dùng xem chi tiết về đơn hàng.

4.8. Trang admin quản lý cửa hàng



The screenshot displays the admin interface for managing stores. On the left is a green sidebar with the user profile 'Sandra Adams' and email 'sandra_a68@gmail.com'. The sidebar menu includes 'Dashboard', 'Quản lý cửa hàng' (highlighted), 'Quản lý hoa quả', 'Quản lý hóa đơn', and 'Quản lý loại'. At the bottom of the sidebar is a 'LOGOUT' button. The main content area has a search bar labeled 'Nhập thông tin' and a 'TAO MỚI' button. Below this is a table listing four stores with the following data:

STT	Tên cửa hàng	Địa chỉ cụ thể	Kinh độ	Vĩ độ	Chi nhánh	Số điện thoại của chi nhánh	Action
1	Cửa hàng Đại Nam4	15 Thanh Xuân Bắc	20.9901461	105.7977623	4	0762386462	 
2	Cửa hàng Đại Nam3	68 triều khúc	20.9843153	105.7996367	3	0762386462	 
3	Cửa hàng Đại Nam2	206 Quang trung	20.9684777	105.7731563	2	0762386462	 
4	Cửa hàng đại nam	Số 1 phố xóm phù lâm hà đông hà nội	20.7858446	106.6144558	1	0904332197	 

Hình 4. 8 Trang admin quản lý cửa hàng

Form nhập thông tin cửa hàng: Phần chính của trang, chứa form để nhập thông tin về cửa hàng.

Footer: Phần dưới cùng chứa các liên kết quan trọng và thông tin liên hệ.

Logo: Logo của hệ thống quản lý đặt ở góc trái trên cùng.

Thanh điều hướng: Bao gồm các liên kết đến các trang quan trọng trong hệ thống quản lý.

STT: Ô nhập liệu cho số thứ tự của cửa hàng, có thể tự động tăng hoặc nhập thủ công.

Tên cửa hàng: Ô nhập liệu cho tên cửa hàng.

Địa chỉ cụ thể: Ô nhập liệu cho địa chỉ chi tiết của cửa hàng.
































Kinh độ và Vĩ độ: Ô nhập liệu cho tọa độ địa lý của cửa hàng, bao gồm kinh độ và vĩ độ.

Số lượng chi nhánh: Ô nhập liệu cho số lượng chi nhánh của cửa hàng.

Số điện thoại chi nhánh: Ô nhập liệu cho số điện thoại của mỗi chi nhánh, có thể là một hoặc nhiều số.

Nút Lưu: Nút để lưu thông tin của cửa hàng sau khi đã nhập hoặc chỉnh sửa.

4.9. Trang admin quản lý hoa quả

<div> <div>  <div> <div>Sandra Adams</div> <div>sandra_a88@gmail.com</div> </div> </div> <div> <div>Dashboard</div> <div>Quản lý cửa hàng</div> <div>Quản lý hoa quả</div> <div>Quản lý hóa đơn</div> <div>Quản lý loại</div> </div> <div> <div>LOGOUT</div> <div>localhost:5173/admin/fruits</div> </div> </div>										
STT	Hình ảnh	Tên hoa quả	Giá bán	Giá giảm	Giảm giá %	Mô tả chi tiết	Thuộc danh mục	Số lượng	Chi nhánh	Tùy chọn
1		Dưa việt	10000	8500	15%		dưa lươi	100	Cửa hàng Đại Nam3	 
2		Dưa lươi nhật	10000	9000	10%		dưa lươi	100	Cửa hàng Đại Nam3	 
3		dưa lươi dài loan	10000	9000	10%		dưa lươi	2000	Cửa hàng Đại Nam3	 
4		Dưa lươi	30000	27000	10%		dưa lươi	1000	Cửa hàng Đại Nam3	 
5		dầu tây	10000	8000	20%		dầu tây	100	Cửa hàng Đại Nam3	 
6		Cherry_newzealand	20000	16000	20%		chery	20000	Cửa hàng Đại Nam2	 
7		cherry_mi	10000	8000	20%		chery	200000000	Cửa hàng Đại Nam2	 
8		Cherry_chile	60000	54000	10%		chery	100000000	Cửa hàng Đại Nam2	 
9		Cam Mĩ	30000	24000	20%		cam	50	Cửa hàng đại nam	 
10		Cam ục	100000	80000	20%		cam	2	Cửa hàng đại nam	 

Hình 4. 9 Admin quản lý hoa quả

Bảng danh sách hoa quả: Hiện thị thông tin của các loại hoa quả trong danh sách.

Thông tin chi tiết: Mỗi hàng trong bảng chứa thông tin như STT, hình ảnh, tên hoa quả, giá bán, giảm giá (nếu có), mô tả chi tiết, danh mục, số lượng chi nhánh có hoa quả đó.

Nút thêm/sửa/xoá: Cung cấp các nút thêm mới, sửa thông tin hoặc xoá một loại hoa quả.

Chức năng tìm kiếm: Cho phép tìm kiếm hoa quả theo tên.

STT: Ô nhập liệu cho số thứ tự của hoa quả, có thể tự động tăng hoặc nhập thủ công.

Hình ảnh: Cho phép quản trị viên tải lên hoặc chọn hình ảnh đại diện cho hoa quả.

Tên hoa quả: Ô nhập liệu cho tên của hoa quả.

Giá bán: Ô nhập liệu cho giá bán của hoa quả.

Giảm giá: Ô nhập liệu cho mức giảm giá (nếu có) của hoa quả.

Mô tả chi tiết: Ô nhập liệu cho mô tả chi tiết về hoa quả.

Danh mục: Dropdown menu cho phép chọn danh mục mà hoa quả thuộc về.

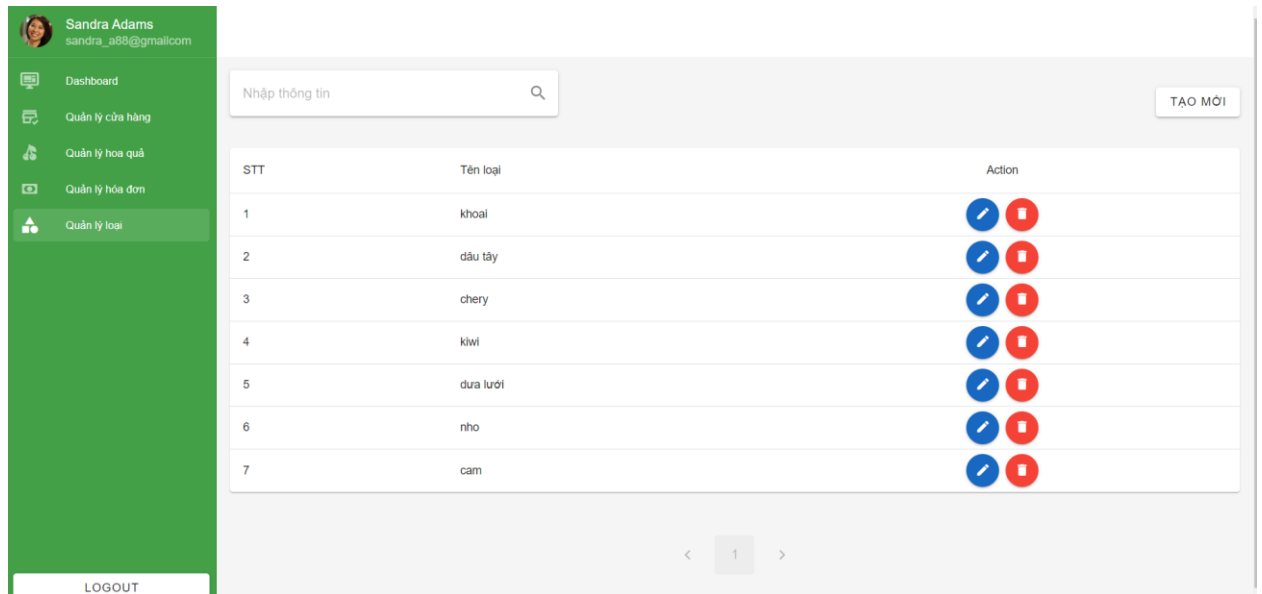
Số lượng chi nhánh: Ô nhập liệu cho số lượng chi nhánh có hoa quả đó.

Nút Lưu: Nút để lưu thông tin của hoa quả sau khi đã nhập hoặc chỉnh sửa.

Nút Reset: Nút để reset form về trạng thái ban đầu.

Thông báo lỗi: Hiện thị thông báo nếu có lỗi xảy ra trong quá trình nhập liệu.

4.10. Trang quản lý các loại



Hình 4. 10 Trang quản lý các loại

Bảng danh sách: Hiện thị danh sách các loại sản phẩm hiện có.

Thông tin chi tiết: Mỗi hàng trong bảng chứa thông tin như STT, tên loại sản phẩm.

Nút thêm/sửa/xoá: Các nút chức năng cho phép thêm mới, sửa đổi hoặc xoá một loại sản phẩm.

Chức năng tìm kiếm: Cho phép tìm kiếm loại sản phẩm theo tên.

STT: Ô nhập liệu cho số thứ tự của loại sản phẩm, có thể tự động tăng hoặc nhập thủ công.

Tên loại sản phẩm: Ô nhập liệu cho tên của loại sản phẩm.

Nút Lưu: Nút để lưu thông tin của loại sản phẩm sau khi đã nhập hoặc chỉnh sửa.

Nút Reset: Nút để reset form về trạng thái ban đầu.

Thông báo lỗi: Hiện thị thông báo nếu có lỗi xảy ra trong quá trình nhập liệu.

KẾT LUẬN

Trong quá trình thực hiện đồ án tốt nghiệp về ứng dụng quản lý chuỗi cửa hàng bán hoa quả, em đã đạt được nhiều thành tựu quan trọng và đáng kể. Ứng dụng được thiết kế và triển khai với các chức năng cơ bản nhưng vô cùng thiết yếu như quản lý sản phẩm, quản lý đơn hàng, quản lý khách hàng và quản lý kho. Những chức năng này không chỉ giúp tối ưu hóa quy trình vận hành mà còn giảm thiểu sai sót, từ đó tăng cường hiệu quả kinh doanh và khả năng cạnh tranh cho chuỗi cửa hàng.

Kết quả đạt được:

- Trong quá trình thực hiện đồ án tốt nghiệp, em đã đạt được những kết quả sau đây sử dụng .NET Core MVC:

- Xây dựng ứng dụng web hiệu quả: Em đã sử dụng .NET Core MVC để xây dựng một ứng dụng web cho chuỗi cửa hàng bán hoa quả. Việc sử dụng .NET Core giúp cho ứng dụng có hiệu suất cao và dễ dàng mở rộng trong tương lai.

- Quản lý sản phẩm và đơn hàng: Em đã tích hợp tính năng quản lý sản phẩm, đơn hàng và khách hàng vào ứng dụng, sử dụng cơ sở dữ liệu SQL Server và kết hợp với Entity Framework Core để quản lý dữ liệu một cách hiệu quả và an toàn.

- Tích hợp thanh toán và giao diện người dùng: Em đã phát triển giao diện người dùng thân thiện, sử dụng nguyên tắc thiết kế Material Design và Bootstrap để tối ưu hóa trải nghiệm người dùng. Đồng thời, tích hợp các cổng thanh toán điện tử để khách hàng có thể thanh toán một cách thuận tiện và an toàn.

- Tối ưu hóa hoạt động cửa hàng: Ứng dụng giúp chuỗi cửa hàng tối ưu hóa hoạt động, từ quản lý kho, xử lý đơn hàng đến tương tác và phản hồi với khách hàng một cách hiệu quả hơn, nhằm tăng trải nghiệm và sự hài lòng của khách hàng.

Hướng phát triển đầu:

- Mở rộng tính năng và tích hợp thêm các dịch vụ: Bổ sung các tính năng như quản lý đánh giá sản phẩm, khuyến mãi, chương trình thưởng và tích điểm để khuyến khích khách hàng quay lại mua sắm thường xuyên hơn.

- Tối ưu hóa SEO và marketing online: Áp dụng các kỹ thuật SEO (Search Engine Optimization) để tăng khả năng tìm thấy của ứng dụng trên các công cụ tìm kiếm và phát triển chiến lược marketing online để thu hút người dùng mới.

- Phân tích dữ liệu và đưa ra quyết định dựa trên dữ liệu: Sử dụng công nghệ phân tích dữ liệu để hiểu rõ hơn hành vi mua hàng của khách hàng và từ đó đưa ra các chiến lược kinh doanh và marketing hiệu quả.

- Tối ưu hóa và mở rộng hệ thống: Liên tục tối ưu hóa hiệu suất của ứng dụng và mở rộng hệ thống để đáp ứng nhu cầu ngày càng tăng của khách hàng và thị trường.

- Nghiên cứu và áp dụng công nghệ mới: Theo dõi các xu hướng công nghệ mới như AI (Trí tuệ nhân tạo), Blockchain để áp dụng vào hệ thống và cải thiện trải nghiệm người dùng.

Mặc dù ứng dụng đã đáp ứng tốt các yêu cầu ban đầu, em nhận thấy vẫn còn nhiều tiềm năng để tiếp tục phát triển và cải tiến trong tương lai. Một số hướng phát triển tiềm năng bao gồm tích hợp hệ thống thanh toán trực tuyến để tạo sự thuận tiện cho khách hàng, phát triển ứng dụng di động để tăng tính tiện lợi và khả năng tiếp cận, nâng cao tính năng phân tích dữ liệu để hỗ trợ quản lý đưa ra các quyết định chiến lược dựa trên dữ liệu thực tế, và tăng cường bảo mật để bảo vệ dữ liệu và quyền riêng tư của người dùng.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

- [1]. Nguyễn Văn Hiếu, "Quản lý hệ thống bán hàng hiện đại," Nhà xuất bản Thống Kê, 2019.
- [2]. Trần Thị Mai, "Cơ sở dữ liệu và ứng dụng," Nhà xuất bản Đại học Quốc gia TP.HCM, 2018.
- [3]. Lê Minh Hoàng, "Lập trình ASP.NET và SQL Server," Nhà xuất bản Khoa học và Kỹ thuật, 2020.

Tiếng Anh:

- [1]. Adam Freeman. Pro ASP.NET Core MVC 2. Apress, 2017.
- [2]. Andrew Lock. ASP.NET Core in Action. Manning Publications, 2018.
- [3]. Dino Esposito. Programming ASP.NET Core. Microsoft Press, 2020.
- [4]. Kevin Hoffman. Building Microservices with ASP.NET Core. O'Reilly Media, 2020.
- [5]. Christian Nagel. Professional C# 7 and .NET Core 2.0. Wrox Press, 2018.
- [6]. Microsoft Official Courses (MOC). Course 20486: Developing ASP.NET Core MVC Web Applications.

Danh mục Website tham khảo:

- [1]. <https://bkhost.vn/blog/web-api>
- [2]. <https://glints.com/vn/blog/asp-net-la-gi/>
- [3]. <https://vuejs.org/>
- [4]. <https://sandbox.vnpayment.vn/apis/docs/thanh-toan-installment/installment.html>
- [5]. <https://cloud.google.com/free>.