

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO
ĐỒ ÁN TỐT NGHIỆP

THIẾT KẾ VÀ XÂY DỰNG
SÀN THƯƠNG MẠI ĐIỆN TỬ NÔNG SẢN

Ngành: Khoa học máy tính

HỘI ĐỒNG: Hội đồng 4 Khoa học Máy Tính
GVHD1: PGS Trần Minh Quang
GVHD2: ThS Bùi Tiến Đức
GVPB: ThS Lê Thị Bảo Thu
—oo—
SVTH1: Phan Thị Quỳnh Như (2011780)
SVTH2: Ngô Bích Ngọc (2010451)

Thành phố Hồ Chí Minh, Tháng 9/2024



Lời cam đoan

Chúng tôi xin cam đoan rằng đồ án này là kết quả của công việc nghiên cứu đồng đội, được thực hiện dưới sự hướng dẫn tận tâm của PGS Trần Minh Quang tại Trường Đại Học Bách Khoa - Đại học Quốc gia TP.HCM. Chúng tôi xác nhận rằng tất cả các thành viên trong nhóm đều đã đóng góp công sức và kiến thức của mình một cách tích cực để hoàn thành đồ án này.

Chúng tôi cam đoan rằng đồ án này không phải là bản sao từ bất kỳ công trình nào khác và tất cả các nguồn thông tin đã được chú thích đầy đủ theo quy tắc trích dẫn khoa học. Mọi ý kiến, thông tin hay dữ liệu từ nguồn ngoại lai đều được công bố và thực hiện theo quy tắc đạo đức và quy định của Trường Đại Học Bách Khoa - Đại học Quốc gia TP.HCM.

Chúng tôi xác nhận rằng tất cả các thành viên trong nhóm đã thực hiện nghiên cứu này với lòng tận tụy và tính chân thành cao, tuân thủ mọi nguyên tắc đạo đức nghiên cứu khoa học. Chúng tôi cũng đảm bảo rằng không có hành vi gian lận hoặc vi phạm nào đã xảy ra trong quá trình thực hiện đồ án.

Chúng tôi chấp nhận mọi trách nhiệm và hậu quả pháp lý liên quan đến nội dung của đồ án và đồng thời biết ơn sự hướng dẫn, góp ý quý báu của PGS Trần Minh Quang đã giúp chúng tôi hoàn thành đồ án này.

Thành phố Hồ Chí Minh, 28th September 2024

NGƯỜI THỰC HIỆN

Phan Thị Quỳnh Như

Ngô Bích Ngọc



Lời cảm ơn

Chúng tôi, nhóm nghiên cứu gồm hai thành viên, xin gửi lời cảm ơn chân thành đến tất cả mọi người đã đóng góp vào thành công của đồ án này.

Đầu tiên và quan trọng nhất, chúng tôi muốn bày tỏ lòng biết ơn đến PGS Trần Minh Quang . Sự tận tâm và kiến thức sâu rộng của thầy không chỉ hướng dẫn chúng tôi qua những thách thức của đồ án mà còn giúp chúng tôi phát triển kỹ năng nghiên cứu và phê bình.

Chúng tôi cũng muốn bày tỏ lòng biết ơn đặc biệt đến tất cả những người bạn, đồng nghiệp, và gia đình đã hỗ trợ chúng tôi trong suốt quá trình nghiên cứu. Sự góp ý và ý kiến của mọi người đã làm giàu thêm nội dung và chất lượng của đồ án.

Cuối cùng, nhưng không kém phần quan trọng, chúng tôi cảm ơn nhau - đối tác nghiên cứu đồng hành trong mỗi bước của đồ án. Sự hợp tác và đóng góp chung của chúng tôi đã tạo nên một sản phẩm mà chúng tôi tự hào.

Chúng tôi tin rằng đồ án này là một bước tiến quan trọng trong sự phát triển của chúng tôi và không thể đạt được nếu không có sự hỗ trợ và đóng góp của tất cả mọi người.



Tóm tắt nội dung

Sàn thương mại điện tử nông sản kết nối chuỗi cung ứng nông nghiệp, thực phẩm tại Việt Nam. Mục đích là tạo ra một nền tảng cho phép kết nối tất cả các thành phần trong chuỗi cung ứng nông nghiệp thực phẩm bao gồm: các nhà cung cấp vật tư nông nghiệp, thực phẩm, nông dân, trang trại, các đơn vị chế biến, vận chuyển, phân phối, bán lẻ kho vận,... tới người tiêu dùng.

Được xây dựng với nhiệm vụ hỗ trợ các nhà sản xuất, hộ kinh doanh để phát triển kinh doanh thương mại và đồng thời bảo vệ quyền, lợi ích hợp pháp, giá trị sản phẩm của nhà sản xuất, hộ kinh doanh. Với mục đích kết nối với các thị trường của Việt Nam ở tất cả các lĩnh vực sản xuất, kinh doanh, dịch vụ, cung ứng sản phẩm hàng hoá, nông sản,...

Bằng những hành động thiết thực, chúng tôi mong muốn vận hành sàn thương mại điện tử nông sản giúp các nhà sản xuất, hộ kinh doanh thành viên không ngừng phát triển, tăng năng lực sản xuất, kinh doanh, mở rộng thị phần trong nước.

Contents

1 Giới thiệu đề tài	9
1.1 Đặt vấn đề	9
1.2 Mục tiêu dự án	9
1.3 Phạm vi dự án	11
1.4 Nghiên cứu sản phẩm trên thị trường	11
1.4.1 FoodMap	12
1.4.2 Postmart	13
1.4.3 Sendo Farm	14
1.4.4 Kết luận	15
2 Nền tảng lý thuyết và công nghệ áp dụng	17
2.1 Kiến trúc Clean Architecture	17
2.2 Design pattern: DI and builder	18
2.3 Front-end	20
2.3.1 Next.js với TypeScript	20
2.3.2 Mantine	21
2.3.3 Vercel	21
2.4 Back-end - Database	22
2.4.1 ASP.NET Core và Entity Framework (EF) Core	22
2.4.2 PostgreSQL Database	22
3 Phân tích hệ thống	24
3.1 Phân tích nghiệp vụ chức năng người dùng	24
3.1.1 Chủ cửa hàng	24
3.1.1.1 Nghiệp vụ quản lý cửa hàng	24
3.1.1.2 Nghiệp vụ quản lý sản phẩm	25
3.1.1.3 Nghiệp vụ bán hàng	25
3.1.1.4 Nghiệp vụ hỗ trợ khách hàng	26
3.1.1.5 Quản lý ưu đãi	26
3.1.1.6 Quản lý đánh giá	27
3.1.1.7 Nghiệp vụ quản lý hậu thu hoạch	28
3.1.1.8 Quản lý đơn đặt trước	28
3.1.2 Khách hàng	30
3.1.2.1 Mua hàng	30
3.1.2.2 Đặt trước	31
3.1.2.3 Giỏ hàng và Đơn hàng	31
3.1.2.4 Nghiệp vụ thanh toán	33



3.2	Usecase diagram và đặc tả use-case	34
3.2.1	Usecase hệ thống	34
3.2.2	Đặc tả usecase	35
3.2.2.1	Đăng nhập	35
3.2.2.2	Đổi mật khẩu	36
3.2.2.3	Đăng xuất	37
3.2.2.4	Đăng ký	38
3.2.2.5	Quản lý các đơn đặt hàng	39
3.2.2.6	Quản lý các phiếu ưu đãi	40
3.2.2.7	Tìm kiếm	41
3.2.2.8	Thông báo	42
3.2.2.9	Báo cáo	43
3.2.2.10	Xem chi tiết sản phẩm	44
3.2.2.11	Quản lý Cài đặt	45
3.2.2.12	Đặt hàng	46
3.2.2.13	Xem giỏ hàng	47
3.2.2.14	Thêm vào giỏ hàng	48
3.2.2.15	Xoá khỏi giỏ hàng	49
3.2.2.16	Sử dụng phiếu ưu đãi	50
3.2.2.17	Nhận phiếu ưu đãi	51
3.2.2.18	Đánh giá	52
3.2.2.19	Thanh toán	53
3.2.2.20	Huỷ đơn đặt hàng	55
3.2.2.21	Nhận và xử lý đơn đặt hàng	56
3.2.2.22	Huỷ đơn hàng	57
3.2.2.23	Quản lý sản phẩm	58
3.2.2.24	Tạo sản phẩm	58
3.2.2.25	Xoá sản phẩm	59
3.2.2.26	Cập nhật sản phẩm	61
3.2.2.27	Quản lý phiếu ưu đãi	62
3.2.2.28	Tạo phiếu ưu đãi	63
3.2.2.29	Xoá phiếu ưu đãi	65
3.2.2.30	Cập nhật phiếu ưu đãi	66
3.2.2.31	Nhận tiền thanh toán từ các đơn hàng	67
4	Thiết kế hệ thống	68
4.1	Mô hình hoá cơ sở dữ liệu	68
4.1.1	Mô hình hóa cơ sở dữ liệu mức khái niệm	68
4.1.2	Mô hình hóa cơ sở dữ liệu mức luận lý	69
4.2	Kiến trúc hệ thống	69
4.3	API Endpoints	70
4.3.1	App Settings	70
4.3.2	Authentication	70
4.3.3	Orders	71
4.3.4	Products	72
4.3.5	Stripe Integration	73
4.3.6	Users	74



4.4	Hiện thực	75
4.4.1	Đăng nhập - Đăng ký	75
4.4.2	Trang chủ	76
4.4.3	Thông tin sản phẩm	78
4.4.4	Đặt hàng	78
4.4.5	Chỉnh sửa thông tin cá nhân	79
4.4.6	Quản lý đơn đặt hàng	80
4.4.7	Kênh cho chủ cửa hàng	81
4.4.8	Chế độ sáng - tối	83
5	Tổng kết và hướng phát triển đề tài	84
5.1	Tổng kết	84
5.2	Hướng phát triển	85

List of Figures

2.1	Minh họa Clean Architecture, nguồn [1]	17
2.2	Dependency Injection	18
2.3	Builder cho phép xây dựng các đối tượng phức tạp theo từng bước	19
2.4	Next.js và Typescript	20
2.5	Mantine	21
2.6	Vercel	21
2.7	EF Core và ASP.NET Core	22
2.8	PostgreSQL Service	23
3.1	Use-case hệ thống	34
4.1	Lược đồ thực thể mối liên kết	68
4.2	Mô hình dữ liệu quan hệ	69
4.3	Kiến trúc hệ thống	69
4.4	Trang đăng nhập	75
4.5	Trang đăng ký	76
4.6	Trang chủ	76
4.7	Danh mục sản phẩm	77
4.8	Phân loại sản phẩm theo danh mục	77
4.9	Thẻ sản phẩm	78
4.10	Trang thông tin sản phẩm chi tiết	78
4.11	Giỏ hàng	79
4.12	Form xác nhận đơn hàng	79
4.13	Trang thông tin người dùng	80
4.14	Địa chỉ người dùng	80
4.15	Trang thông tin người dùng - Quản lý đơn hàng	81
4.16	Quản lý sản phẩm	82
4.17	Thêm sản phẩm mới	82
4.18	Một số trang ở chế độ tối	83

List of Tables

3.1	Use case Đăng nhập	35
3.2	Use case Đổi mật khẩu	37
3.3	Use case Đăng xuất	37
3.4	Use case Đăng ký	39
3.5	Use case Quản lý các đơn đặt hàng	39
3.6	Use case Quản lý các phiếu ưu đãi	40
3.7	Use case Tìm kiếm	41
3.8	Use case Xem thông báo	42
3.9	Use case Báo cáo	43
3.10	Use case Xem chi tiết sản phẩm	44
3.11	Use case Quản lý cài đặt	45
3.12	Use case Đặt hàng	46
3.13	Use case Xem giỏ hàng	47
3.14	Use case Thêm vào giỏ hàng	48
3.15	Use case Xoá khỏi giỏ hàng	49
3.16	Use case Sử dụng phiếu ưu đãi	51
3.17	Use case Nhận phiếu ưu đãi	51
3.18	Use case Đánh giá	52
3.19	Use case Thanh toán	54
3.20	Use case Huỷ đơn đặt hàng	55
3.21	Use case Nhận và xử lý đơn hàng	56
3.22	Use case Huỷ đơn hàng	57
3.23	Use case Quản lý sản phẩm	58
3.24	Use case Tạo sản phẩm	59
3.25	Use case Xoá sản phẩm	60
3.26	Use case Cập nhật sản phẩm	62
3.27	Use case Quản lý phiếu ưu đãi	63
3.28	Use case Tạo phiếu ưu đãi	64
3.29	Use case Xoá phiếu ưu đãi	65
3.30	Use case Cập nhật phiếu ưu đãi	67
3.31	Use case Nhận tiền thanh toán từ các đơn hàng	67

Chapter 1

Giới thiệu đề tài

1.1 Đặt vấn đề

Ngành nông nghiệp đóng vai trò quan trọng trong nền kinh tế và cung cấp nguồn thực phẩm thiết yếu cho xã hội. Tuy nhiên, việc mua bán nông sản và dịch vụ nông nghiệp vẫn phụ thuộc chủ yếu vào các phương pháp truyền thống như sử dụng đại lý, thị trường truyền thống hoặc giao dịch trực tiếp giữa các bên liên quan. Các phương pháp này có thể gặp phải nhiều hạn chế, các nhà sản xuất và doanh nghiệp trong ngành nông nghiệp thường gặp khó khăn trong việc tiếp cận thị trường, quảng bá sản phẩm và tìm kiếm khách hàng. Đồng thời, người tiêu dùng cũng gặp rào cản khi muốn tìm kiếm và mua các sản phẩm nông nghiệp chất lượng cao từ các nhà sản xuất đáng tin cậy.

Do đó, cần có một giải pháp số hóa để tạo ra một môi trường kết nối hiệu quả giữa nhà sản xuất, doanh nghiệp và người tiêu dùng trong ngành nông nghiệp. Vì vậy nhu cầu phát triển một nền tảng thương mại điện tử chuyên về nông sản và dịch vụ nông nghiệp là hoàn toàn hợp lý.

Dự án của chúng tôi hướng tới việc cung cấp một nền tảng cho phép các nhà sản xuất và doanh nghiệp đăng ký và quảng bá sản phẩm của họ, cung cấp thông tin chi tiết về nguồn gốc, chất lượng và phương pháp sản xuất. Đồng thời, người tiêu dùng sẽ có thể dễ dàng tìm kiếm, so sánh và mua các sản phẩm nông nghiệp trực tuyến từ các nhà sản xuất và doanh nghiệp uy tín.

1.2 Mục tiêu dự án

Mục tiêu của dự án xây dựng sàn thương mại điện tử nông sản là tận dụng sức mạnh của công nghệ và nền tảng kỹ thuật số để biến đổi ngành nông nghiệp tại Việt Nam. Dự án nhằm xây dựng một thị trường trực tuyến mạnh mẽ và hiệu quả để kết nối cũng như phục vụ một loạt người dùng đa dạng, từ các doanh nghiệp nông nghiệp quy mô lớn, đến các cửa hàng nông sản nhỏ lẻ, những người sản xuất nông nghiệp và người tiêu dùng sản phẩm.

Một trong những mục tiêu chính của dự án là tạo cơ hội cho nông dân bằng cách cung cấp cho họ quyền tiếp cận trực tiếp đến một tệp khách hàng tiềm năng lớn hơn. Thông qua nền tảng Thương mại điện tử Nông sản, nông dân có thể giới thiệu sản phẩm của mình đến một đối tượng khách hàng rộng hơn, không cần phải thông qua các bên trung gian truyền thống và nhận được giá trị thích đáng hơn cho sản phẩm nông sản của mình. Mỗi kết nối trực tiếp này không chỉ tăng hiệu quả kinh doanh cho nông dân mà còn giúp họ xây dựng mối quan hệ mạnh mẽ với khách hàng của mình.

Bằng việc xây dựng một sàn thương mại điện tử nông sản, dự án nhằm tối ưu hóa quy trình



mua bán sản phẩm nông nghiệp. Nền tảng thương mại điện tử tạo điều kiện giao dịch minh bạch, loại bỏ các bên trung gian không cần thiết, giảm chi phí giao dịch và nâng cao hiệu quả thị trường. Hiệu quả nâng cao này mang lại lợi ích cho cả doanh nghiệp nông nghiệp và người tiêu dùng bằng cách cung cấp giá cạnh tranh, chất lượng sản phẩm cải thiện và đa dạng hơn. Hơn nữa, bằng cách cung cấp một nền tảng thương mại điện tử, các doanh nghiệp vừa và nhỏ có thể mở rộng phạm vi hoạt động của mình vượt xa thị trường địa phương và cạnh tranh với các doanh nghiệp lớn hơn.

Sàn thương mại điện tử nông sản lưu trữ thông tin về tất cả các tài liệu và hoạt động cũng như đối tượng tham gia giao dịch qua sàn:

- Khách hàng (người tiêu dùng) và giỏ hàng riêng, phương thức thanh toán của họ
- Cửa hàng và các sản phẩm thuộc về cửa hàng và các loại sản phẩm
- Các tài liệu ghi lại như đơn đặt hàng, hoàn trả, thanh toán và báo cáo kinh doanh cho cửa hàng
- Phiếu giảm giá và chương trình khuyến mãi
- Đơn vị vận chuyển

Các tính năng chính của hệ thống là:

- Quản lý sản phẩm:
 - Thêm, xóa hoặc chỉnh sửa thông tin sản phẩm thuộc về cửa hàng
 - Cung cấp thông tin chi tiết về sản phẩm
- Quản lý giao dịch mua bán:
 - Người dùng có giỏ hàng riêng để thêm sản phẩm
 - Thanh toán và xác nhận đơn hàng
 - Theo dõi tình trạng đơn hàng đang được giao
 - Xử lý đổi trả hoàn tiền
- Hỗ trợ mua hàng thông qua các tính năng như tìm kiếm sản phẩm, duyệt danh mục sản phẩm, xem thông tin sản phẩm,...
- Quản lý ưu đãi và khuyến mãi để thu hút khách hàng, tạo động lực mua sắm, và tăng doanh số bán hàng

Xem chi tiết ở chương 3: [Phân tích yêu cầu nghiệp vụ chức năng người dùng](#).



1.3 Phạm vi dự án

Với mong muốn xây dựng một nền tảng nhằm giúp các bên cung cấp sản phẩm nông sản và người tiêu dùng sản phẩm có một môi trường kết nối hiệu quả, quản lý và hỗ trợ các hoạt động mua bán một cách hiệu quả hơn, dự án chủ yếu nhắm vào những người sản xuất nông sản và các cửa hàng nông sản nhỏ lẻ để giúp họ có thể kết nối với khách hàng và hoạt động kinh doanh hiệu quả hơn. Khách hàng mục tiêu có thể là người tiêu dùng nông sản, một doanh nghiệp về nông nghiệp đang cần thị trường hoặc muốn mở rộng thị trường mua bán, một nhóm người muốn bắt đầu một cửa hàng nông sản nhỏ hoặc chỉ đơn giản là một cá nhân muốn bán một số nông sản tự sản xuất trực tuyến.

Dự án sẽ tập trung vào việc phát triển một sàn thương mại điện tử nông sản toàn diện. Phạm vi cụ thể của dự án bao gồm:

- Xây dựng giao diện người dùng thân thiện, dễ sử dụng cho cả người sản xuất và người tiêu dùng.
- Tạo ra trải nghiệm mua sắm trực tuyến thuận tiện và linh hoạt, giúp người dùng dễ dàng tìm kiếm và đặt mua sản phẩm.
- Tích hợp các chức năng thanh toán an toàn và thuận tiện, đảm bảo giao dịch mua bán diễn ra một cách trọn vẹn và đáng tin cậy.
- Phát triển hệ thống quản lý sản phẩm, giúp người sản xuất quản lý thông tin sản phẩm một cách hiệu quả.
- Tích hợp chức năng đánh giá và nhận xét sản phẩm từ cộng đồng người dùng, tạo điều kiện cho việc chia sẻ trải nghiệm và tăng tính minh bạch trong giao dịch mua bán.
- Liên kết với các hệ thống vận chuyển để hỗ trợ quá trình giao hàng, cung cấp thông tin vận chuyển và theo dõi tình trạng đơn hàng.

Dự án sẽ tạo ra một nền tảng mua bán nông sản trực tuyến linh hoạt, đáp ứng đa dạng nhu cầu của người dùng và tạo điều kiện thuận lợi cho sự phát triển của ngành nông nghiệp.

1.4 Nghiên cứu sản phẩm trên thị trường

Sau khi trải qua quá trình nghiên cứu, nhóm quyết định chọn FoodMap, PostMart và Sendo Farm làm ba đối tượng khảo sát chính. Nhóm đã phân tích các tính năng về hoạt động mua bán trên cơ sở nghiên cứu về 3 nền tảng này. Lý do chọn 3 nền tảng này là:

- FoodMap là một trong những startup Việt đầu tiên cung cấp sản phẩm nông nghiệp trên kênh trực tuyến với các thương hiệu riêng. FoodMap cũng đã nhận được nhiều giải thưởng trong các cuộc thi về startup, qua đó cho thấy tiềm năng phát triển của FoodMap.
- PostMart là sàn thương mại điện tử nông sản, đặc sản thuộc Tổng công ty Bưu điện Việt Nam. Khác với những sàn thương mại điện tử thuộc sở hữu nước ngoài như Shopee, Lazada, Postmart là sàn thương mại điện tử thuần Việt Nam, được phát triển với mục đích cao nhất là tạo ra một chợ số dành riêng cho các tiểu thương trong nước, cung ứng các sản phẩm hàng hóa, đặc biệt là hàng nông sản mang thương hiệu Việt Nam đến tay người tiêu dùng.

- Sendo là một sàn thương mại điện tử lớn tại Việt Nam. Nó cung cấp một nền tảng trực tuyến cho các người bán và người mua để giao dịch hàng hóa và dịch vụ. Sendo cung cấp một loạt các sản phẩm đa dạng trong nhiều lĩnh vực khác nhau, bao gồm thời trang, điện tử, gia dụng, làm đẹp, đồ chơi, đồ gia dụng, và nhiều hơn nữa. Với lợi thế có sẵn, sau khi hợp tác với một số địa phương và tạo ra một chuyên mục riêng dành cho nông sản (Sendo Farm), đây sẽ là một nền tảng đáng để tham khảo và nghiên cứu.

1.4.1 FoodMap

Nền tảng thương mại điện tử FoodMap có mặt trên thị trường từ năm 2018, là một trong những startup Việt đầu tiên cung cấp sản phẩm nông nghiệp trên kênh trực tuyến với các thương hiệu riêng là Đặc sản Ngon lành (như đường, mật ong, rau củ quả...), Maloka (trà và cà phê) và HappyNut (các loại hạt dinh dưỡng). FoodMap là mô hình kết nối trực tiếp từ nông dân và nhà sản xuất với khách hàng cả B2B (Business To Business) (70%) lẫn B2C (Business To Customer) (30%). Đến nay công ty cung ứng nông sản và thực phẩm có nguồn gốc từ hơn 300 trang trại và các nhà sản xuất nông nghiệp Việt Nam. Năm 2021, trong suốt thời gian dịch Covid-19 diễn biến phức tạp tại TP.HCM, FoodMap là một trong số ít startup được cấp phép hoạt động liên tục và đảm bảo nguồn cung cấp thực phẩm cho người dân trong giai đoạn giãn cách xã hội. Nhờ đó FoodMap đạt được mức tăng trưởng đột biến, số người dùng lẫn doanh số đều tăng đến 300%.



- Điểm mạnh:**

- Giao diện bắt mắt, thân thiện với người dùng.
- Có thể đặt hàng mà không cần đăng nhập, tạo sự tiện lợi cho người dùng mới muốn dùng thử.
- Hình thức thanh toán đa dạng.
- Đơn hàng gồm nhiều sản phẩm từ nhiều nhà cung cấp khác nhau được gom chung lại để giao, giúp người tiêu dùng nhận đơn hàng dễ dàng và tiết kiệm được chi phí giao hàng.
- Có lựa chọn nhận hàng tại kho nếu khách hàng có yêu cầu.
- Đa nền tảng (có cả trang web và ứng dụng trên điện thoại).
- Phân loại sản phẩm đa dạng, có cả phân theo đặc sản theo vùng miền, phù hợp với thị hiếu khách hàng.
- Có trang tin tức, giúp người dùng cập nhật thông tin nhanh chóng.
- Tất cả các sản phẩm đều được tích hợp mã QR, cho phép khách hàng tìm hiểu thông tin và nguồn gốc xuất xứ thông qua website hoặc ứng dụng di động.



• **Điểm yếu:**

- Khi đặt hàng mà không đăng nhập, không kiểm tra được tính xác thực của số điện thoại, địa chỉ giao hàng, và user không theo dõi được tình trạng đơn đặt hàng.
- Sau khi đặt hàng không có chốt hủy đơn.
- Trải nghiệm người dùng kém, rất nhiều chốt sau khi bấm vào bị reload lại trang.
- Giới hạn số lượng mua quá thấp, với những khách hàng có ý định mua sỉ sẽ không thể giao dịch trên sàn mà phải liên hệ trực tiếp như bình thường.

1.4.2 Postmart

Postmart là sàn thương mại điện tử được sáng lập, thuộc quyền sở hữu thương hiệu và vận hành trực tiếp bởi Tổng công ty Bưu điện Việt Nam (VietNam Post). Dựa trên cơ sở đó, Postmart đã được quy hoạch trở thành nền tảng thương mại điện tử Quốc Gia. Mục tiêu của Postmart là giới thiệu và bán các loại nông sản, đặc sản Việt Nam thông qua việc kết nối trực tiếp giữa nông dân, hộ sản xuất nhỏ lẻ với người tiêu dùng ở nhiều tỉnh, thành phố, góp phần giúp người nông dân Việt tham gia, hội nhập và phát triển kinh tế số, thúc đẩy và hỗ trợ phát triển tiêu thụ nông sản khắp Việt Nam.



• **Điểm mạnh:**

- Độ uy tín cao được khẳng định thông qua thương hiệu của VietNam Post, giúp có được sự tin tưởng từ người dùng.
- Đối tác truyền thông rộng rãi và uy tín.
- Bộ lọc đa dạng (có phân theo vùng miền), phù hợp với thị hiếu khách hàng.
- Đa nền tảng (có cả trang web và ứng dụng trên điện thoại).
- Có tính năng so sánh sản phẩm.
- Có thể hủy đơn hàng dễ dàng.
- Có thể theo dõi tình trạng đơn hàng rõ ràng như khi gửi hàng bình thường qua VietNam Post.
- Tích hợp các tính năng mua chung, bán chung và các tính năng như mạng xã hội (livestream, livechat, lướt bảng tin, bình luận, đánh giá), giúp người dùng dễ dàng tương tác trên nền tảng.



- Đối với người bán, Postmart cung cấp tiện ích bán hàng chuyên nghiệp và tài liệu hướng dẫn chi tiết.
- Không tốn phí tạo gian hàng.
- Công nghệ truy xuất nhanh chóng với mã QR kèm tính năng bảo hộ thương hiệu và chứng nhận thương hiệu nhà cung cấp uy tín.

• **Điểm yếu:**

- Với một số sản phẩm, kết quả tìm kiếm không khớp với nội dung.
- Tính năng lọc theo đặc sản vùng miền không hoạt động, khi bấm vào không hiện kết quả.
- Một số tính năng được đưa ra nhưng chưa hiện thực (như so sánh sản phẩm).
- Vì là sản phẩm của VietNam Post nên chỉ vận chuyển qua VietNam Post, không có sự lựa chọn phương thức vận chuyển đa dạng như một số sàn khác.

1.4.3 Sendo Farm

Sendo còn có tên gọi khác “Siêu Chợ Sen Đỏ”, là sàn thương mại điện tử nổi tiếng, ra đời và hoạt động dưới sự bảo trợ của tập đoàn FPT Shop. Là sàn thương mại điện tử Việt tiên phong trong việc khai phá thị trường tỉnh và thành phố cấp 2, Sendo luôn đẩy mạnh việc đưa các mặt hàng đặc sản địa phương đến khách hàng tỉnh. Đây cũng là mục tiêu hàng đầu của sàn thương mại điện tử này. Sendo Farm hoạt động như một ứng dụng nhỏ bên trong ứng dụng Sendo. Sendo Farm là mô hình đi chợ kiểu mới, đặt hàng qua ứng dụng và nhận hàng gần nhà. Cách thức hoạt động rất khác biệt so với các sàn thương mại điện tử hiện nay. Thay vì giao trực tiếp đến nơi ở của người tiêu dùng, ứng dụng sẽ cung cấp cho người dùng những điểm nhận hàng gần mình nhất, người mua sẽ chọn điểm nhận hàng và đến lấy.



• **Điểm mạnh:**

- Với lợi thế có sẵn từ Sendo, giúp Sendo Farm có sẵn một lượng lớn người tiêu dùng.
- Chuyển đổi từ Sendo thường sang Sendo Farm rất dễ dàng, việc tích hợp chung với Sendo thường giúp tận dụng được lượng khách hàng có sẵn từ Sendo.
- Chất lượng đảm bảo bởi FPT - một tập đoàn uy tín ở Việt Nam.



- Đã hợp tác xã với rất nhiều địa phương để thúc đẩy tình hình kinh tế ở các địa phương đó, điều này cũng góp phần làm tăng độ uy tín của ứng dụng.
- Liên kết với rất nhiều ví điện tử và ngân hàng, giúp việc thanh toán online dễ dàng.
- Cách thức vận hành khác biệt so với các sàn thương mại điện tử trên thị trường.
- Thời gian nhận hàng nhanh, được cam kết bởi ứng dụng - “Mua hôm nay, mai nhận ngay”.
- Nhờ cách thức vận hành đặc biệt nên luôn được miễn phí giao hàng, khách hàng chỉ cần đi quãng đường ngắn là đã mua được nông sản tươi và một số nông sản không có ở gần nơi ở. Nhờ đó thu hút một nhóm khách hàng tiềm năng khác.
- Thu hút thêm một lượng lớn người dùng với mục đích đăng ký làm điểm nhận hàng.
- Có thể chọn thời gian nhận hàng tùy ý.

- **Điểm yếu:**

- Chỉ có trên nền tảng điện thoại, chưa có website.
- Không có phương thức thanh toán trực tiếp.
- Thời gian cho phép hủy đơn quá ngắn (5 phút).
- Bộ lọc không đa dạng như một số sàn khác trên thị trường.
- Điểm nhận hàng chưa có mặt ở những địa phương xa thành phố.
- Sản phẩm không đa dạng do bị giới hạn bởi khu vực.

1.4.4 Kết luận

- **Tối ưu hóa cho các cửa hàng nhỏ tại Việt Nam:**

- Các hệ thống kể trên hướng đến cả doanh nghiệp nhỏ, vừa và lớn, nên có rất nhiều chức năng phức tạp, có thể khiến người dùng bối rối khi sử dụng. Bên cạnh đó, những gì mà nhiều cửa hàng nhỏ cần thực sự chỉ là một số chức năng nhỏ trong số rất nhiều chức năng mà hệ thống cung cấp. Ví dụ, mục tiêu chính của một số cửa hàng chỉ là bán sản phẩm, in hóa đơn và xem thống kê mà không cần quản lý hàng tồn kho.
- Do đó, hệ thống sẽ kế thừa, cải tiến cũng như đơn giản hóa một số chức năng cơ bản của các hệ thống trên như các giao dịch mua bán, hoàn trả hàng, quản lý hàng tồn kho, khách hàng, báo cáo và thống kê, sẽ dễ sử dụng hơn, đơn giản hơn và phù hợp với các nhà bán lẻ nhỏ ở Việt Nam.

- **Thêm một số chức năng tự động hóa cơ bản:**

- Hệ thống sẽ hỗ trợ một số chức năng tự động hóa cơ bản giúp người dùng giảm thiểu sai sót trong các công việc thủ công khi quản lý tồn kho như thông báo để cảnh báo sản phẩm sắp hết hàng hoặc sản phẩm sắp hết hạn. Ngoài ra, hệ thống sẽ hỗ trợ một số chức năng giúp quy trình làm việc của cửa hàng hiệu quả hơn.



- **Giao diện người dùng thân thiện hơn:**

- Một trong những yếu tố quan trọng ảnh hưởng đến quyết định của người dùng xem họ sẽ sử dụng hệ thống hay không, không chỉ là các tính năng cần thiết, tính bảo mật, hiệu suất, mà còn là giao diện người dùng. Vì giao diện người dùng là yếu tố đầu tiên thu hút người dùng quyết định sử dụng một hệ thống. Nhiều người dùng cần một giao diện thân thiện về cả thiết kế giao diện người dùng và trải nghiệm người dùng. Nên việc tập trung xây dựng một giao diện thân thiện với người dùng là một yếu tố quan trọng quyết định khả năng thành công của hệ thống.

Chapter 2

Nền tảng lý thuyết và công nghệ áp dụng

2.1 Kiến trúc Clean Architecture

Clean Architecture [1] được xây dựng trên nền tảng của tư tưởng "độc lập," kết hợp một cách hài hòa với nguyên lý thiết kế hướng đối tượng, trong đó Dependency Inversion là một biểu hiện đặc trưng. Trong bối cảnh này, độc lập có nghĩa là dự án không bị ràng buộc bởi các framework và công cụ được sử dụng trong quá trình kiểm thử.

Kiến trúc của Clean Architecture chia thành 4 lớp, mỗi lớp có quy tắc phụ thuộc rõ ràng. Các lớp bên trong được thiết kế để không biết gì về các lớp bên ngoài, thể hiện mối quan hệ phụ thuộc hướng vào bên trong. Clean Architecture giữ cho các phần khác nhau của hệ thống độc lập và linh hoạt, mà không làm suy giảm hiệu suất hay khả năng mở rộng của dự án.

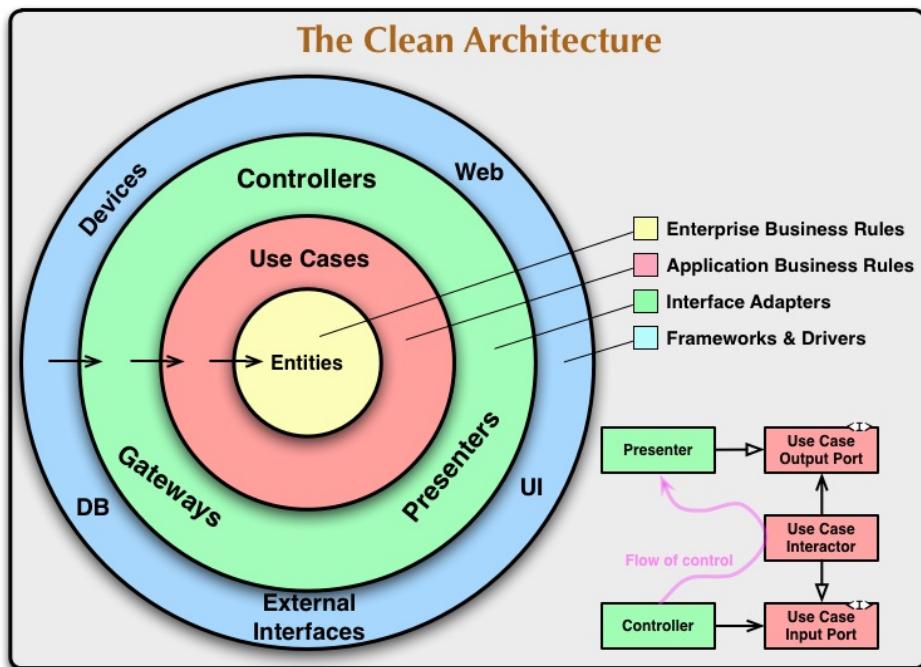


Figure 2.1: Minh họa Clean Architecture, nguồn [1]

Ở mức độ chi tiết, Thực thể đóng vai trò như những yếu tố cơ bản trong kiến trúc, đóng gói quy tắc kinh doanh toàn diện. Dù có dạng đối tượng hoặc là cấu trúc dữ liệu và chức năng,

thực thể không quan tâm đến ứng dụng cụ thể nào đang sử dụng chúng. Trong khi đó, Lớp Use Cases chứa quy tắc kinh doanh cụ thể cho ứng dụng và đảm bảo luồng dữ liệu giữa các thực thể.

Interface Adapters đóng vai trò như trung gian, chuyển đổi dữ liệu giữa định dạng thuận tiện nhất cho Use Cases và Thực thể và định dạng thuận tiện nhất cho đơn vị bên ngoài như cơ sở dữ liệu hoặc web. Cuối cùng, lớp Frameworks và Drivers chứa các công cụ và chi tiết cấp thấp nhất, như cơ sở dữ liệu và framework web, với chức năng chính là đóng vai trò như một đường ống.

Quy tắc phụ luôn được duy trì, giúp tăng cường trừu tượng hóa theo hướng vòng tròn bên trong. Mỗi lớp giữ vai trò của một hộp chứa độc lập, giúp duy trì sự cân bằng giữa linh hoạt và sự kiểm soát trong thiết kế kiến trúc Clean Architecture.

2.2 Design pattern: DI and builder

Trong thế giới phát triển phần mềm hiện đại, việc sử dụng Design Pattern không chỉ là một xu hướng mà còn là một chiến lược quan trọng để tạo ra mã nguồn dễ bảo trì và linh hoạt. Hai mô hình quan trọng, Dependency Injection (DI) và Builder Pattern, là những "công cụ" quan trọng trong "hộp công cụ" của các nhà phát triển giúp tối ưu hóa việc quản lý các thành phần của hệ thống và cung cấp khả năng linh hoạt đáng kể trong việc xây dựng đối tượng phức tạp.

Dependency Injection (DI):

DI [4] là một design pattern tập trung vào việc giảm sự phụ thuộc giữa các thành phần của hệ thống, giúp tăng tính linh hoạt và kiểm soát. Thay vì các đối tượng tạo ra và quản lý chính nó, DI chuyển trách nhiệm này đến bên ngoài, thường là một hệ thống quản lý phụ thuộc (dependency container). Điều này giúp giảm độ kết dính giữa các thành phần, làm cho mã nguồn dễ bảo trì và mở rộng hơn. Hơn nữa, DI thúc đẩy việc tái sử dụng mã nguồn, vì các đối tượng không phụ thuộc trực tiếp vào cách chúng được tạo ra.

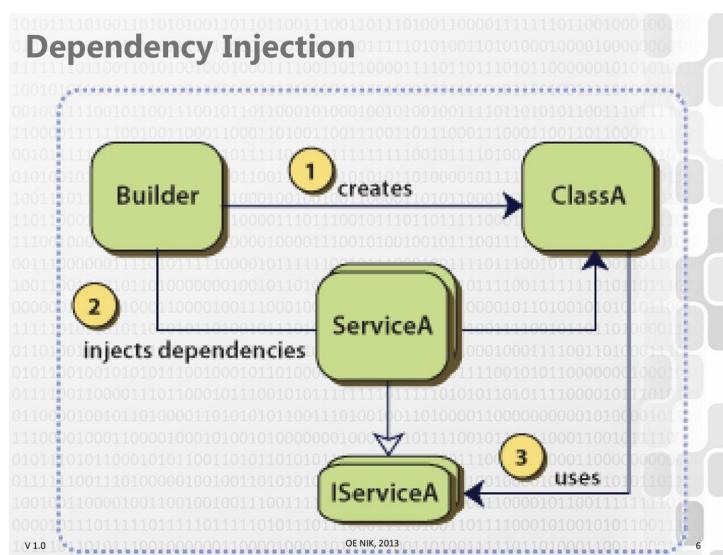


Figure 2.2: Dependency Injection

Builder Pattern:

Builder Pattern [3] tập trung vào việc xây dựng đối tượng phức tạp, giúp tách rời quá trình xây dựng và biểu diễn. Thay vì sử dụng nhiều constructor hoặc phương thức khởi tạo, Builder Pattern cung cấp một cách linh hoạt và tự nhiên để xây dựng đối tượng bằng cách sử dụng một builder object. Điều này không chỉ làm cho mã nguồn dễ đọc hơn mà còn giúp quản lý các tham số xây dựng, đặc biệt là khi có nhiều tùy chọn. Builder Pattern thường được sử dụng để xây dựng các đối tượng không thay đổi và không có sự thay đổi trạng thái, giúp giảm lỗi và làm tăng khả năng kiểm thử.

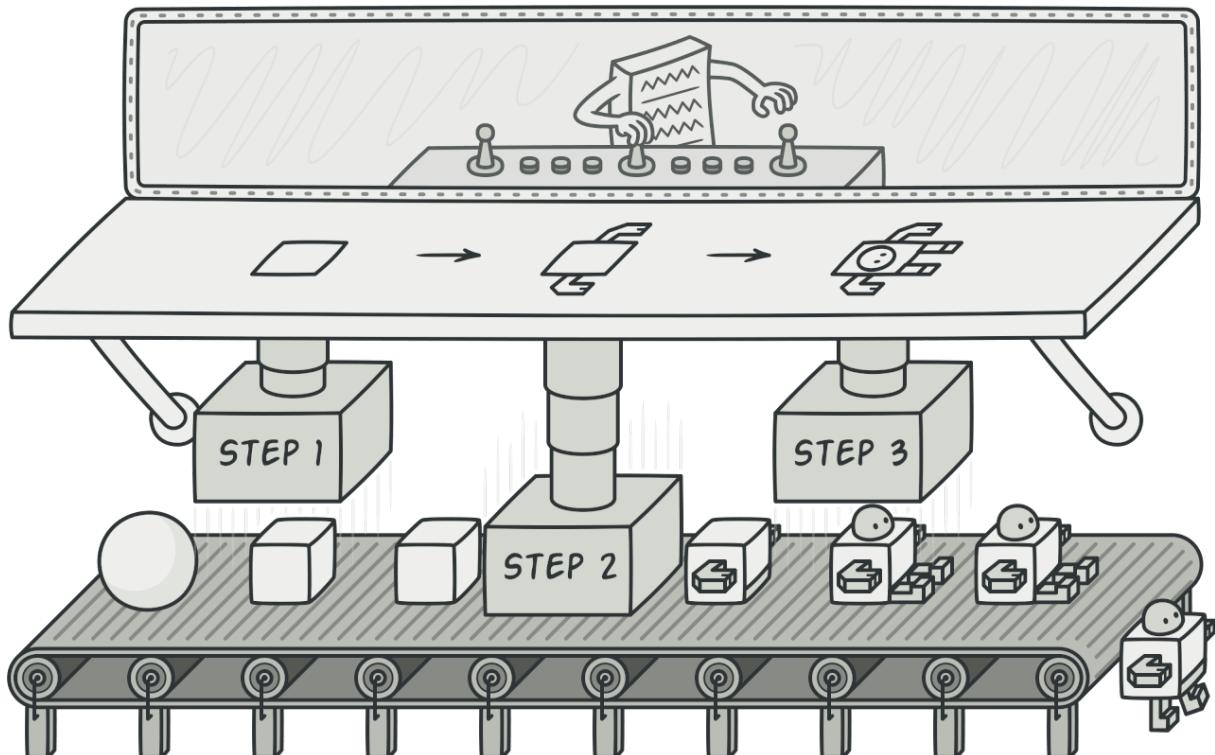


Figure 2.3: Builder cho phép xây dựng các đối tượng phức tạp theo từng bước

Tích Hợp DI và Builder Pattern:

Khi kết hợp DI và Builder Pattern, chúng ta có thể tận dụng sức mạnh của cả hai mô hình để tạo ra hệ thống linh hoạt, dễ bảo trì, và có thể mở rộng một cách dễ dàng. Dependency Injection giảm sự phụ thuộc, trong khi Builder Pattern giúp quản lý quá trình xây dựng các đối tượng phức tạp. Sự kết hợp này làm cho mã nguồn dễ đọc, dễ kiểm thử, và giảm thiểu các vấn đề liên quan đến thiết kế và xây dựng.

Bằng cách tận dụng sự mạnh mẽ của cả Dependency Injection và Builder Pattern, chúng ta có thể xây dựng các hệ thống linh hoạt và dễ bảo trì, đồng thời giảm thiểu rủi ro về lỗi và tăng tính mở rộng của mã nguồn.

2.3 Front-end

2.3.1 Next.js với TypeScript

Next.js [6] là một framework React mã nguồn mở được phát triển bởi Vercel, nhằm cung cấp một giải pháp toàn diện cho việc xây dựng ứng dụng web hiện đại. Next.js kế thừa tất cả ưu điểm của React và bổ sung thêm nhiều tính năng mạnh mẽ.

Next.js sử dụng cách tiếp cận "page-based routing" kết hợp với "component-based" của React để xây dựng giao diện. Điều này cho phép tổ chức cấu trúc ứng dụng một cách trực quan và hiệu quả, đồng thời vẫn giữ được khả năng tái sử dụng và modular hóa cao của các component. Cách tiếp cận này giúp tách biệt logic và giao diện, dễ dàng quản lý và bảo trì mã nguồn, đồng thời tạo ra một kiến trúc ứng dụng rõ ràng và dễ mở rộng.

Một trong những tính năng nổi bật của Next.js là khả năng Server-Side Rendering (SSR) và Static Site Generation (SSG). SSR giúp cải thiện đáng kể thời gian tải trang đầu tiên và tối ưu hóa SEO, trong khi SSG cho phép tạo ra các trang tĩnh tại thời điểm build, giúp tăng tốc độ và giảm tải cho server. Next.js cũng hỗ trợ Incremental Static Regeneration (ISR), cho phép cập nhật từng phần của trang tĩnh mà không cần phải build lại toàn bộ site [?].

Next.js tích hợp sẵn nhiều tính năng quan trọng như code splitting, lazy loading, và tối ưu hóa hình ảnh. Điều này giúp cải thiện hiệu suất ứng dụng một cách đáng kể mà không cần cầu hình phức tạp.

Khi kết hợp Next.js với TypeScript, ta có thể tận dụng tất cả lợi ích của việc kiểm tra kiểu tĩnh và lập trình hướng đối tượng mạnh mẽ. TypeScript giúp phát hiện lỗi trong quá trình biên dịch và cung cấp các tính năng gõ lỗi thông minh, tăng tính ổn định và độ tin cậy của ứng dụng [7].

Việc sử dụng TypeScript với Next.js còn mang lại lợi ích trong việc quản lý trạng thái và props của components. TypeScript cho phép xác định các kiểu dữ liệu chính xác cho trạng thái, props, và các API routes của Next.js, giúp tạo ra mã nguồn dễ đọc, dễ hiểu và dễ bảo trì. Điều này đặc biệt hữu ích trong các dự án lớn và phức tạp.

Next.js cũng cung cấp API Routes, cho phép xây dựng API serverless một cách dễ dàng. Kết hợp với TypeScript, ta có thể tạo ra các API an toàn về kiểu dữ liệu và dễ dàng tích hợp với phần front-end của ứng dụng.

Tóm lại, Next.js kết hợp với TypeScript là một công cụ mạnh mẽ cho phát triển ứng dụng web hiện đại. Với khả năng SSR, SSG, ISR, hiệu suất cao, kiến trúc dựa trên pages và components, tính năng mạnh mẽ của TypeScript và sự hỗ trợ từ cộng đồng, Next.js giúp tạo ra các ứng dụng web linh hoạt, có hiệu suất cao, dễ bảo trì và đáng tin cậy.

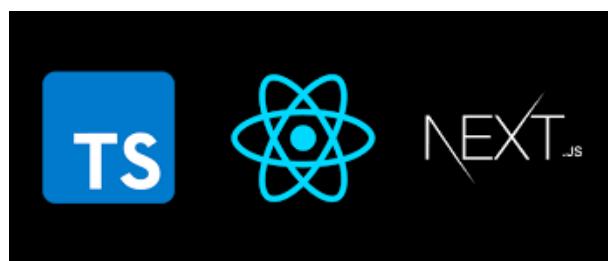


Figure 2.4: Next.js và Typescript

2.3.2 Mantine

Mantine [8] là một thư viện UI React mã nguồn mở, cung cấp các thành phần giao diện đẹp mắt, dễ sử dụng và linh hoạt. Nó cung cấp một tập hợp các thành phần giao diện hiện đại, từ các nút, biểu mẫu và bảng cho đến các thanh trượt, menu và hộp thoại. Điều này giúp tạo ra giao diện thân thiện với người dùng, hiệu quả và đồng nhất trên toàn bộ ứng dụng.

Một trong những điểm mạnh của Mantine là tính tùy chỉnh cao. Nó không chỉ cung cấp các thành phần giao diện sẵn có, mà còn cho phép người phát triển tùy chỉnh chúng dựa trên nhu cầu cụ thể. Điều này đảm bảo rằng giao diện cuối cùng sẽ phù hợp với mục tiêu thiết kế và mong muốn của người dùng.

Ngoài ra, Mantine còn cung cấp một bộ tài liệu phong phú và chi tiết, gồm hướng dẫn sử dụng, ví dụ mẫu và hướng dẫn thiết kế. Điều này giúp người phát triển nhanh chóng làm quen với Ant Design và tận dụng tối đa các tính năng mạnh mẽ của nó.

Mantine là một lựa chọn lý tưởng để xây dựng giao diện người dùng chất lượng và dễ sử dụng trong các dự án phát triển phần mềm. Với tính tương thích, tích hợp dễ dàng, tính tùy chỉnh cao, Mantine là lựa chọn thích hợp đối với công cụ phát triển giao diện người dùng.



Figure 2.5: Mantine

2.3.3 Vercel

Vercel [9] là một nền tảng triển khai và lưu trữ trang web, ứng dụng Front-end được phát triển bởi Vercel, Inc. (trước đây là Zeit). Nền tảng này được thiết kế đặc biệt để triển khai các ứng dụng web hiện đại và tĩnh dựa trên các framework như React, Next.js, Vue.js và Nuxt.js.

Với Vercel, có thể dễ dàng triển khai mã nguồn thông qua quy trình CI/CD tích hợp sẵn. Nền tảng này tự động xây dựng, triển khai và cung cấp các bản xem trước cho mỗi lần push code lên Git. Vercel cũng cung cấp các tính năng như phân phối nội dung toàn cầu thông qua mạng CDN, quản lý các bản triển khai, tích hợp các dịch vụ bên thứ ba và tự động hóa quy trình build/deploy.

Một trong những lợi ích chính của Vercel là nó cho phép tập trung vào việc xây dựng ứng dụng mà không cần phải lo lắng về việc quản lý cơ sở hạ tầng triển khai. Vercel xử lý tất cả các nhiệm vụ liên quan đến triển khai, bảo mật và mở rộng, giúp gia tăng năng suất và tập trung vào phát triển tính năng.



Figure 2.6: Vercel

2.4 Back-end - Database

2.4.1 ASP.NET Core và Entity Framework (EF) Core

ASP.NET Core [10] là một framework phát triển ứng dụng web mạnh mẽ được phát triển bởi Microsoft. Nó cung cấp một nền tảng cho việc xây dựng các ứng dụng web đa dạng và mạnh mẽ bằng cách sử dụng ngôn ngữ lập trình C#.

Entity Framework(EF) Core là một ORM (Object-Relational Mapping) framework cũng do Microsoft phát triển, giúp tương tác và quản lý cơ sở dữ liệu trong ứng dụng. Nó giúp lập trình viên làm việc với dữ liệu cơ sở dữ liệu bằng cách sử dụng đối tượng và cung cấp một lớp truy xuất giúp ẩn đi các chi tiết về cơ sở dữ liệu thực tế.

Khi kết hợp sử dụng ASP.NET Core và Entity Framework Core, người phát triển có thể xây dựng các ứng dụng web mạnh mẽ và dễ dàng quản lý cơ sở dữ liệu. Thông qua Entity Framework Core, dữ liệu được quản lý trong các đối tượng .NET, giúp giảm thiểu việc phải viết các câu truy vấn SQL trực tiếp.Thêm vào đó, ASP.NET Core cung cấp nền tảng cho việc xây dựng các giao diện người dùng tương tác, cho phép người dùng tương tác với dữ liệu được quản lý bởi Entity Framework Core.

Tóm lại, sự kết hợp giữa ASP.NET Core và Entity Framework Core cung cấp cho các nhà phát triển một cách tiếp cận toàn diện để xây dựng ứng dụng web chất lượng cao với khả năng tương tác với dữ liệu hiệu quả.



Figure 2.7: EF Core và ASP.NET Core

2.4.2 PostgreSQL Database

PostgreSQL Database [12] là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở mạnh mẽ và đáng tin cậy, được phát triển bởi cộng đồng PostgreSQL Global Development Group. Với hơn 30 năm phát triển, PostgreSQL đã trở thành một trong những lựa chọn hàng đầu cho các ứng dụng yêu cầu độ tin cậy cao, tính toàn vẹn dữ liệu, và khả năng xử lý khối lượng lớn.

Dự án đã tận dụng sức mạnh của PostgreSQL thông qua dịch vụ Neon, một nền tảng cung cấp PostgreSQL trên cloud. Neon cung cấp một giải pháp serverless, cho phép tập trung vào việc phát triển ứng dụng mà không cần lo lắng về việc quản lý cơ sở hạ tầng. Điều này giúp tối ưu hóa chi phí vận hành và tăng tính linh hoạt cho hệ thống.

Tính năng ACID (Atomicity, Consistency, Isolation, Durability) của PostgreSQL đảm bảo tính toàn vẹn dữ liệu cao, điều này đặc biệt quan trọng đối với các ứng dụng yêu cầu độ tin cậy

cao như hệ thống của nhóm. Ngoài ra, PostgreSQL còn cung cấp các tính năng nâng cao như partitioning, replication, và full-text search, giúp tối ưu hóa hiệu suất và khả năng mở rộng của hệ thống.

Về mặt bảo mật, PostgreSQL cung cấp nhiều lớp bảo vệ, bao gồm xác thực mạnh mẽ, mã hóa dữ liệu, và kiểm soát truy cập chi tiết. Khi kết hợp với các tính năng bảo mật của Neon, có thể đảm bảo rằng dữ liệu của người dùng luôn được bảo vệ ở mức cao nhất.

PostgreSQL cũng nổi tiếng với khả năng mở rộng và hiệu suất cao. Nó có thể xử lý hàng triệu bản ghi một cách hiệu quả, đồng thời hỗ trợ các truy vấn phức tạp và các giao dịch đồng thời. Điều này cho phép ứng dụng phục vụ số lượng lớn người dùng mà không gặp vấn đề về hiệu suất.

Cuối cùng, việc sử dụng PostgreSQL trên nền tảng Neon cho phép tận dụng các tính năng như sao lưu tự động, khôi phục điểm thời gian, và scaling tự động. Điều này giúp đảm bảo tính khả dụng cao và khả năng phục hồi nhanh chóng trong trường hợp xảy ra sự cố.

Tóm lại, việc lựa chọn PostgreSQL trên nền tảng Neon đã cung cấp một giải pháp cơ sở dữ liệu mạnh mẽ, linh hoạt và đáng tin cậy. Nó không chỉ đáp ứng các yêu cầu hiện tại của dự án mà còn cung cấp nền tảng vững chắc cho sự phát triển trong tương lai.



Figure 2.8: PostgreSQL Service

Chapter 3

Phân tích hệ thống

3.1 Phân tích nghiệp vụ chức năng người dùng

3.1.1 Chủ cửa hàng

3.1.1.1 Nghiệp vụ quản lý cửa hàng

1. Tạo cửa hàng: Doanh nghiệp hoặc cá nhân cần tạo cửa hàng trên tài khoản của mình tại trang web thương mại điện tử để bắt đầu kinh doanh.
2. Tạo sản phẩm bán: Cửa hàng cần tạo sản phẩm và cung cấp đầy đủ thông tin về sản phẩm, bao gồm tên sản phẩm, mô tả sản phẩm, giá cả, hình ảnh, và thông tin vận chuyển.
3. Thêm số lượng sẵn có: Cửa hàng cần cập nhật số lượng sản phẩm sẵn có để khách hàng có thể biết được sản phẩm còn hàng hay đã hết hàng.
4. Thêm mô tả cho sản phẩm: Mô tả sản phẩm là một yếu tố quan trọng giúp khách hàng hiểu rõ về sản phẩm. Cửa hàng cần cung cấp mô tả sản phẩm đầy đủ và chi tiết.
5. Tạo các danh mục sản phẩm: Danh mục sản phẩm giúp khách hàng dễ dàng tìm kiếm sản phẩm. Cửa hàng nên tạo các danh mục sản phẩm rõ ràng và logic để giúp khách hàng tìm kiếm sản phẩm dễ dàng hơn.
6. Điều chỉnh giá thành của sản phẩm: Giá thành của sản phẩm là một yếu tố quan trọng ảnh hưởng đến quyết định mua hàng của khách hàng. Cửa hàng nên điều chỉnh giá thành của sản phẩm phù hợp với thị trường để tăng doanh số bán hàng.
7. Tạo các combo sản phẩm: Combo sản phẩm là một cách hiệu quả để tăng giá trị cho khách hàng. Cửa hàng có thể tạo các combo sản phẩm với giá ưu đãi để thu hút khách hàng mua nhiều sản phẩm hơn.
8. Thêm các chương trình tặng quà khi mua sản phẩm: Các chương trình tăng quà khi mua sản phẩm là một cách hiệu quả để khuyến khích khách hàng mua hàng. Cửa hàng có thể tạo các chương trình tăng quà khi mua sản phẩm để thu hút khách hàng mới và tăng doanh số bán hàng cho các sản phẩm bán chạy.



3.1.1.2 Nghiệp vụ quản lý sản phẩm

Nghiệp vụ sản phẩm trên sàn thương mại điện tử nông sản bao gồm quá trình quản lý và tiếp thị sản phẩm để thu hút khách hàng, tạo ra doanh số bán hàng, và đảm bảo sự hài lòng của khách hàng. Quản lý sản phẩm trên sàn thương mại điện tử nông sản là một phần quan trọng của hoạt động kinh doanh và đòi hỏi sự chú ý đến chi tiết, hiểu biết về thị trường và nhu cầu của khách hàng, cũng như khả năng tiếp tục tối ưu hóa sản phẩm và chiến dịch tiếp thị.

1. Quản lý Sản Phẩm:

- Thêm, chỉnh sửa, và xóa sản phẩm trên sàn thương mại điện tử.
- Cung cấp thông tin chi tiết về sản phẩm, bao gồm tên, mô tả, hình ảnh, giá cả, số lượng tồn kho, và các thuộc tính kèm theo.

2. Phân Loại Sản Phẩm: Tạo danh mục và phân loại sản phẩm để giúp khách hàng dễ dàng tìm kiếm và duyệt qua các sản phẩm tương tự.

3. Thông Tin sản phẩm: Cung cấp thông tin chi tiết về sản phẩm, ví dụ như trọng lượng, kích thước, xuất xứ, thời hạn sử dụng, và hướng dẫn sử dụng.

4. Giá và Chiết Khấu: - Xác định giá cả cho sản phẩm, bao gồm giá gốc và giá bán.

- Áp dụng chiết khấu hoặc ưu đãi đặc biệt nếu có.

5. Quản lý Tồn Kho: Theo dõi số lượng tồn kho của từng sản phẩm và cập nhật thông tin về số lượng sản phẩm còn lại tức thì.

6. Quản lý Đánh Giá và Nhận Xét:

- Cho phép khách hàng đánh giá và viết nhận xét về sản phẩm.
- Kiểm duyệt và quản lý đánh giá và nhận xét để đảm bảo tính trung thực và hữu ích.

7. Quản lý Sản Phẩm Liên Quan: Kết nối các sản phẩm tương tự hoặc có liên quan để thúc đẩy giao dịch chéo (cross-selling) và gợi ý sản phẩm.

8. Tối ưu hóa Sản Phẩm: Sản phẩm nên được cập nhật đều đặn để cải thiện tính năng, chất lượng, hoặc cách sử dụng để đáp ứng nhu cầu khách hàng ngày càng biến đổi.

9. Chăm sóc và Hỗ trợ Sản Phẩm: Cung cấp hỗ trợ kỹ thuật, chăm sóc sau bán hàng, và hướng dẫn sử dụng sản phẩm cho khách hàng.

10. Quản lý Sản Phẩm Bán Chạy và Lưu Lượng: Theo dõi sản phẩm nào đang bán chạy và đảm bảo rằng sản phẩm luôn có sẵn trong kho để đáp ứng nhu cầu.

3.1.1.3 Nghiệp vụ bán hàng

1. Xác nhận đơn hàng:

- Người bán xác nhận đơn hàng và chuẩn bị sản phẩm cho quá trình giao hàng.
- Hệ thống gửi thông báo xác nhận đơn hàng đến người mua.



2. Giao hàng:

- Người bán chọn phương thức vận chuyển và gửi sản phẩm đến địa chỉ giao hàng được xác định trong đơn hàng.
- Hệ thống cung cấp thông tin vận chuyển để người mua có thể theo dõi quá trình vận chuyển.

3. Nhận và kiểm tra sản phẩm:

- Người mua nhận sản phẩm và kiểm tra tính chất và chất lượng của sản phẩm.
- Họ có thể đánh giá sản phẩm sau khi nhận hàng.

4. Hỗ trợ khách hàng sau bán hàng: Sàn thương mại điện tử cung cấp dịch vụ hỗ trợ khách hàng sau bán hàng để giải quyết mọi thắc mắc hoặc khiếu nại của người mua.

5. Theo dõi lịch sử đơn hàng: Hệ thống lưu trữ lịch sử đơn hàng của người mua và người bán, giúp họ theo dõi trạng thái và thông tin chi tiết về đơn hàng trước đây.

6. Đánh giá và đánh giá sản phẩm: Sau khi nhận sản phẩm, người mua có thể đánh giá và viết nhận xét về sản phẩm và dịch vụ của người bán để cung cấp phản hồi cho cộng đồng mua sắm trực tuyến.

3.1.1.4 Nghiệp vụ hỗ trợ khách hàng

1. Hướng Dẫn Sử Dụng Sản Phẩm: Cung cấp hướng dẫn sử dụng chi tiết cho sản phẩm để giúp khách hàng tận dụng tối đa sản phẩm mà họ đã mua.
2. Xử Lý Phản Hồi Khách Hàng: Lắng nghe phản hồi từ khách hàng để cải thiện dịch vụ và sản phẩm.
3. Trả lời Câu hỏi và Thắc mắc: Hỗ trợ người mua trả lời câu hỏi và giải đáp thắc mắc liên quan đến sản phẩm, cách sử dụng, vận chuyển, và chính sách đổi trả.
4. Phản hồi và Đánh giá: Thu thập phản hồi và đánh giá từ người mua để cải thiện chất lượng sản phẩm và dịch vụ.
5. Chăm sóc Khách hàng Sau Bán hàng: Cung cấp dịch vụ chăm sóc khách hàng sau bán hàng để đảm bảo họ hài lòng với sản phẩm và dịch vụ.

3.1.1.5 Quản lý ưu đãi

Nghiệp vụ ưu đãi và khuyến mãi trên sàn thương mại điện tử nông sản đóng một vai trò quan trọng trong việc thu hút khách hàng, tạo động lực mua sắm, và tăng doanh số bán hàng. Dưới đây là một số chi tiết về nghiệp vụ này:

1. Tạo Chương Trình Khuyến Mãi:

- Xác định mục tiêu và đối tượng của chương trình khuyến mãi (Ví dụ: giảm giá sản phẩm nông sản trong mùa thu).
- Quyết định loại ưu đãi, chẳng hạn như giảm giá tiền mặt, vận chuyển miễn phí, quà tặng kèm, hoặc voucher.



2. Xác định Điều Kiện Áp Dụng: Quy định điều kiện và hạn chế áp dụng cho ưu đãi, bao gồm thời gian, số lượng, loại sản phẩm, và quy định về sử dụng.
3. Quản Lý Mã Giảm Giá Và Voucher:
 - Tạo và quản lý các mã giảm giá và voucher để phân phối cho khách hàng.
 - Theo dõi và kiểm soát việc sử dụng mã giảm giá để đảm bảo tính hợp lệ và tuân thủ các quy định.
4. Quảng Cáo Khuyến Mãi:
 - Tạo chiến dịch quảng cáo để thông báo về chương trình khuyến mãi.
 - Sử dụng các kênh quảng cáo trực tuyến như quảng cáo trên mạng xã hội, email marketing, và quảng cáo trực tuyến để tiếp cận khách hàng.
5. Theo Dõi Hiệu Suất: Theo dõi hiệu suất chương trình khuyến mãi bằng cách thu thập dữ liệu về số lượng sử dụng, doanh số bán hàng tăng, và ROI (Return on Investment).
6. Lập Kế Hoạch Và Thời Gian:
 - Xác định lịch trình khuyến mãi, bao gồm thời gian bắt đầu và kết thúc của chương trình.
 - Đảm bảo rằng sản phẩm và dịch vụ có sẵn trong kho để đáp ứng nhu cầu.
7. Hỗ trợ Khách Hàng: Cung cấp hỗ trợ cho khách hàng liên quan đến chương trình khuyến mãi, bao gồm hướng dẫn sử dụng mã giảm giá và giải quyết khiếu nại.

3.1.1.6 Quản lý đánh giá

1. Tạo khuyến mãi dựa trên đánh giá: Sử dụng đánh giá tích cực để tạo chương trình khuyến mãi và đánh giá sản phẩm nổi bật.
2. Theo Dõi Phản Hồi Của Khách Hàng: Theo dõi phản hồi từ khách hàng về sản phẩm và dịch vụ để cải thiện chất lượng và trải nghiệm của họ.
3. Quản Lý Đánh Giá Gắn Kèm Hình Ảnh: Cho phép khách hàng đính kèm hình ảnh và video vào đánh giá sản phẩm.
4. Hiển Thị Đánh Giá Trên Trang Sản Phẩm: Đánh giá và đánh giá sản phẩm thường được hiển thị trên trang sản phẩm để giúp người mua xem được đánh giá sản phẩm trước khi mua.
5. Phản Hồi Đánh Giá: Cho phép người bán hoặc quản trị viên trả lời các đánh giá để giải quyết các vấn đề hoặc cung cấp thông tin bổ sung.
6. Hiển Thị Đánh Giá Gắn Kèm Hình Ảnh: Cho phép khách hàng đính kèm hình ảnh hoặc video vào đánh giá sản phẩm để cung cấp thêm thông tin.
7. Đánh Giá Về Dịch Vụ Giao Hàng: Cho phép khách hàng đánh giá dịch vụ giao hàng, bao gồm thời gian giao hàng và tình trạng sản phẩm khi nhận hàng.
8. Tạo Bài Viết Đánh Giá Tương Tác: Cho phép người dùng tương tác với các đánh giá bằng cách bình luận hoặc bỏ phiếu cho những đánh giá có ích.



3.1.1.7 Nghiệp vụ quản lý hậu thu hoạch

Nghiệp vụ quản lý hậu thu hoạch kết hợp với logistics giúp tối ưu hóa các hoạt động sau thu hoạch, bao gồm bảo quản, vận chuyển nông sản. Trên hệ thống, chủ cửa hàng có thể theo dõi chi tiết thông tin sản phẩm như chất lượng, số lượng, thời gian thu hoạch. Họ cũng quản lý các hoạt động logistics như lập lịch trình vận chuyển, theo dõi tình trạng giao hàng theo thời gian thực. Hệ thống còn giúp giám sát tình trạng kho bãi, cảnh báo khi sản phẩm sắp hết hạn sử dụng.

1. Quản lý thông tin sản phẩm:

- Theo dõi thông tin chi tiết về từng loại sản phẩm như tên, số lượng, chất lượng, thời gian thu hoạch.
- Phân loại, đánh mã định danh cho từng lô hàng.
- Cập nhật tình trạng sản phẩm trong các giai đoạn bảo quản, vận chuyển.

2. Quản lý hoạt động logistics:

- Lập kế hoạch và theo dõi lịch trình vận chuyển, bao gồm phương tiện, tuyến đường, thời gian giao hàng.
- Theo dõi tình trạng vận chuyển theo thời gian thực, như vị trí hiện tại của các lô hàng.

3. Quản lý kho bãi:

- Theo dõi tình trạng kho, như nhiệt độ, độ ẩm, diện tích kho trống.
- Quản lý việc nhập, xuất, tồn kho các lô sản phẩm.
- Cảnh báo khi sản phẩm sắp hết hạn sử dụng hoặc cần phải xử lý.

3.1.1.8 Quản lý đơn đặt trước

Đối với một số sản phẩm nông sản, không thể sản xuất hàng loạt và bảo quản lâu dài, cũng như một số sản phẩm chưa đến mùa vụ thì không thể đưa lên sàn bán được. Khi đó, khách hàng có thể tiến hành đặt trước các sản phẩm này và được giao ngay sau khi có hàng. Nghiệp vụ quản lý đơn đặt trước là một trong những hoạt động quan trọng, nó giúp chủ cửa hàng tiếp nhận, xử lý và quản lý các đơn đặt hàng từ khách hàng một cách hiệu quả. Nghiệp vụ này kết hợp chặt chẽ với công tác logistics để đảm bảo hàng hóa được giao đúng thời gian và địa điểm.

Đa số các hoạt động trong nghiệp vụ này đều giống nghiệp vụ bán hàng, chẳng hạn như tiếp nhận đơn đặt hàng, cập nhật tình trạng đơn, quản lý kho hàng, v.v. đều được thực hiện trên hệ thống của cửa hàng. Tuy nhiên có một số hoạt động cần phải cập nhật thêm các vấn đề liên quan tới tình trạng thu hoạch, sản xuất sản phẩm và thời gian giao hàng.

1. Tiếp nhận đơn đặt trước từ khách hàng: Cửa hàng tiếp nhận đặt trước các sản phẩm nông sản chưa có sẵn, xác nhận các thông tin như tên sản phẩm, số lượng, thời gian giao hàng mong muốn, thông tin liên hệ khách hàng, và xác nhận lại về thời gian giao hàng dự kiến.
2. Xác nhận khả năng cung ứng: Chủ cửa hàng kiểm tra kế hoạch sản xuất, thu hoạch để đảm bảo có đủ sản phẩm để đáp ứng yêu cầu của khách hàng. Sau đó sẽ xác nhận với khách hàng về khả năng cung ứng.
3. Theo dõi quá trình sản xuất, thu hoạch: Theo dõi tiến độ sản xuất, thu hoạch để đảm bảo đủ số lượng sản phẩm cho đơn hàng và cập nhật thông tin cho khách hàng về tiến độ và dự kiến thời gian giao hàng.



4. Quản lý và cập nhật tình trạng đơn đặt trước:

- Xem và quản lý danh sách các đơn đặt hàng từ khách hàng. Xác nhận khả năng cung cấp sản phẩm theo số lượng và thời gian giao hàng yêu cầu.
- Cập nhật tình trạng đơn hàng (đang chờ thu hoạch, đang vận chuyển, đã giao) vào hệ thống.
- Thông báo định kỳ cho khách hàng về tình trạng đơn hàng của họ.

5. Quản lý giao nhận:Lên lịch giao hàng cho từng đơn đặt hàng, đảm bảo giao đúng thời gian và xử lý các vấn đề phát sinh trong quá trình giao hàng.



3.1.2 Khách hàng

3.1.2.1 Mua hàng

1. Tìm Kiếm Sản Phẩm: Người mua truy cập sàn thương mại điện tử và tìm kiếm sản phẩm nông sản theo nhu cầu và sở thích cá nhân.
2. Duyệt Danh Mục Sản Phẩm: Khám phá danh mục sản phẩm, danh mục, và các chế độ lọc để tìm sản phẩm phù hợp.
3. Xem Thông Tin Sản Phẩm: Đọc thông tin chi tiết về sản phẩm, bao gồm hình ảnh, mô tả, giá cả, thuộc tính kỹ thuật, và đánh giá từ người dùng khác.
4. Thêm Sản Phẩm Vào Giỏ Hàng: Chọn sản phẩm và thêm vào giỏ hàng để chuẩn bị cho việc thanh toán.
5. Kiểm Tra Giỏ Hàng: Xem lại danh sách sản phẩm trong giỏ hàng, kiểm tra số lượng và giá cả, và chỉnh sửa nếu cần thiết.
6. Đặt Hàng: Tiến hành đặt hàng bằng cách chọn phương thức thanh toán và giao hàng, nhập thông tin giao hàng, và xác nhận đơn đặt hàng.
7. Thanh Toán: Thực hiện thanh toán bằng cách sử dụng thẻ tín dụng, thẻ ghi nợ, chuyển khoản ngân hàng, hoặc các phương thức thanh toán trực tuyến khác.
8. Xác Nhận Đơn Hàng: Nhận được xác nhận đơn đặt hàng từ sàn thương mại điện tử thông qua email hoặc ứng dụng di động.
9. Theo Dõi Đơn Hàng: Theo dõi trạng thái đơn hàng và thông tin vận chuyển thông qua sàn thương mại điện tử hoặc hệ thống theo dõi đơn hàng trực tuyến.
10. Nhận Sản Phẩm: Nhận sản phẩm nông sản được giao tới địa chỉ giao hàng đã cung cấp và kiểm tra chất lượng và số lượng sản phẩm.
11. Đánh Giá và Nhận Xét Sản Phẩm: Sau khi sử dụng sản phẩm, người mua có thể đánh giá và viết nhận xét về sản phẩm trên sàn thương mại điện tử để chia sẻ thông tin với người dùng khác.
12. Lưu Trữ Lịch Sử Mua Hàng: Sàn thương mại điện tử lưu trữ lịch sử mua hàng của người dùng để thuận tiện cho việc theo dõi và tái đặt hàng trong tương lai.
13. Hỗ Trợ Khách Hàng: Cung cấp hỗ trợ khách hàng liên quan đến quá trình mua hàng, đổi trả, và các câu hỏi khác.
14. Phí Vận Chuyển và Thuế: Trong quá trình đặt hàng, người mua cần kiểm tra các khoản phí vận chuyển và thuế để đảm bảo hiểu rõ giá tổng cộng của đơn hàng.
15. Phương Thức Thanh Toán: Sàn thương mại điện tử nông sản thường cung cấp nhiều phương thức thanh toán khác nhau, bao gồm thẻ tín dụng, thẻ ghi nợ, PayPal, ví điện tử, và chuyển khoản ngân hàng,...
16. Đánh Giá Tình Trạng Sản Phẩm: Trước khi thanh toán, người mua nên kiểm tra kỹ tình trạng sản phẩm để đảm bảo rằng sản phẩm không bị hỏng hoặc thất lạc.



17. Lựa Chọn Phương Thức Vận Chuyển: Người mua có thể chọn phương thức vận chuyển phù hợp với họ, bao gồm giao hàng tận nơi, lấy hàng tại cửa hàng, hoặc vận chuyển ưu đãi.
18. Theo Dõi Đơn Hàng: Cung cấp mã theo dõi để người mua có thể theo dõi trạng thái vận chuyển của đơn hàng và biết khi nào sản phẩm sẽ được giao đến.
19. An Toàn Thanh Toán: Đảm bảo tính an toàn của quá trình thanh toán trực tuyến bằng cách sử dụng các trang web có tích hợp mã hóa và cung cấp hướng dẫn về an toàn thanh toán.
20. Xử Lý Thanh Toán Hậu Kiểm: Người mua nên kiểm tra tài khoản ngân hàng hoặc thẻ tín dụng sau khi thanh toán để xác nhận rằng giao dịch đã được xử lý một cách chính xác.
21. Hỗ Trợ Trực Tuyến: Cung cấp hỗ trợ trực tuyến qua chat trực tiếp, điện thoại hoặc email để giúp người mua giải quyết các vấn đề liên quan đến đơn hàng.
22. Chương Trình Khách Hàng Thân Thiết: Cung cấp các chương trình khách hàng thân thiết và tích điểm để khuyến khích người mua quay lại và mua sắm thường xuyên.
23. Xác Nhận Giao Dịch: Người mua nên nhận được xác nhận giao dịch sau khi hoàn tất việc thanh toán để làm bằng chứng cho đơn hàng của họ.

3.1.2.2 Đặt trước

Như đã nói trên nghiệp vụ Quản lý đơn đặt trước, khách hàng có thể tiến hành đặt trước đối với một số sản phẩm không có sẵn hay các loại nông sản chưa thu hoạch ngay được. Về cơ bản, nghiệp vụ này cũng có các hoạt động giống như nghiệp vụ Mua hàng, chỉ có một số điểm khác biệt liên quan đến vấn đề thời gian.

1. Xác định nhu cầu và lựa chọn nhà cung cấp:
 - Khách hàng xác định loại sản phẩm nông sản cần mua và thời gian dự kiến nhận hàng.
 - Tìm kiếm và lựa chọn các nhà cung cấp uy tín trên hệ thống.
2. Đặt hàng trước: Khách hàng thực hiện đặt hàng trước sản phẩm mong muốn thông qua tính năng đặt trước trên hệ thống.
3. Thỏa thuận các điều khoản: Trên nền tảng, khách hàng và chủ cửa hàng thỏa thuận các điều khoản như giá cả, phương thức thanh toán và thời gian dự kiến giao hàng. Các thông tin này được ghi nhận trong đơn hàng đặt trước.
4. Theo dõi tiến độ sản xuất và giao hàng: Khách hàng có thể theo dõi tình hình sản xuất, thu hoạch thông qua thông tin cập nhật trên nền tảng.
5. Nhận hàng và thanh toán: Khách hàng nhận hàng đúng số lượng, chất lượng và thời gian đã thỏa thuận, Kiểm tra sản phẩm và thực hiện thanh toán theo đơn hàng đặt trước.

3.1.2.3 Giỏ hàng và Đơn hàng

1. Thêm Sản Phẩm Vào Giỏ Hàng: Người mua có thể chọn sản phẩm nông sản và thêm chúng vào giỏ hàng của họ thông qua một nút "Thêm vào giỏ hàng" hoặc biểu tượng tương tự.
2. Xem Giỏ Hàng: Khách hàng có thể xem lại danh sách các sản phẩm đã chọn bằng cách truy cập vào giỏ hàng từ giao diện của họ.



3. Sửa Sản Phẩm Trong Giỏ Hàng: Cho phép người mua thay đổi số lượng sản phẩm hoặc loại bỏ sản phẩm khỏi giỏ hàng nếu họ muốn thay đổi đơn hàng của mình.
4. Tính Tổng Tiền: Hệ thống tự động tính tổng tiền cho toàn bộ đơn hàng bằng cách cộng tất cả các món hàng và tính toán thuế và phí vận chuyển (nếu có).
5. Sử Dụng Mã Giảm Giá hoặc Voucher: Cho phép khách hàng áp dụng mã giảm giá hoặc voucher để nhận được ưu đãi hoặc giảm giá trên tổng giá trị đơn hàng.
6. Chọn Phương Thức Thanh Toán: Khách hàng có thể chọn phương thức thanh toán ưa thích, bao gồm thẻ tín dụng, thẻ ghi nợ, ví điện tử, và chuyển khoản ngân hàng.
7. Đặt Đơn Hàng: Khách hàng cần xác nhận đơn hàng và tiến hành đặt hàng sau khi đã xem xét và chỉnh sửa lại giỏ hàng.
8. Lưu Giỏ Hàng: Cho phép người mua lưu giỏ hàng để hoàn tất đơn hàng sau này hoặc để đánh giá lại danh sách sản phẩm.
9. Kiểm Tra Số Lượng Còn Trong Kho: Hệ thống cần kiểm tra số lượng sản phẩm còn trong kho để đảm bảo rằng sản phẩm có sẵn để bán và không bị thiếu hàng.
10. Thông Báo Về Sản Phẩm Hết Hạn: Thông báo cho người mua nếu có sản phẩm trong giỏ hàng đã hết hàng hoặc không còn sẵn.
11. Tạo Đơn Hàng Tự Động: Sau khi khách hàng đặt hàng thành công, hệ thống tự động tạo một đơn hàng và gửi xác nhận đơn hàng đến khách hàng.
12. Quản Lý Lịch Sử Giỏ Hàng: Lưu trữ lịch sử giỏ hàng của khách hàng để họ có thể xem lại và theo dõi các đơn hàng trước đó.
13. Hiển Thị Thông Tin Sản Phẩm: Cho phép người mua xem thông tin chi tiết về sản phẩm trong giỏ hàng, bao gồm tên sản phẩm, giá, mô tả, và hình ảnh.
14. Tính Toán Tổng Tiền Chi Phí: Tính tổng tiền của các sản phẩm trong giỏ hàng, bao gồm giá sản phẩm, thuế, và phí vận chuyển (nếu có).
15. Xem Lại Đơn Hàng: Khách hàng có thể xem lại đơn hàng để đảm bảo rằng mọi thông tin đều chính xác trước khi hoàn tất thanh toán.
16. Lưu Trữ Giỏ Hàng Tạm Thời: Hệ thống cần lưu trữ giỏ hàng tạm thời để khách hàng có thể tiếp tục mua sắm và quay lại giỏ hàng sau khi đã đăng nhập vào tài khoản của họ.
17. Xử Lý Mã Giảm Giá hoặc Voucher: Cho phép người mua nhập mã giảm giá hoặc voucher để nhận ưu đãi hoặc giảm giá trên tổng giá trị đơn hàng.
18. Sử Dụng Địa Chỉ Giao Hàng Khác Nhau: Cho phép khách hàng sử dụng địa chỉ giao hàng khác với địa chỉ mặc định trong tài khoản của họ nếu cần thiết.
19. Chọn Phương Thức Thanh Toán Phù Hợp: Hệ thống cần hỗ trợ nhiều phương thức thanh toán và hiển thị chúng để người mua có thể chọn phương thức thuận tiện nhất.
20. Tạo Một Đơn Đặt Hàng Chính Thức: Khi khách hàng hoàn tất quá trình xem giỏ hàng và chọn phương thức thanh toán, hệ thống tạo một đơn đặt hàng chính thức.



21. Xác Nhận Đơn Hàng: Hệ thống gửi xác nhận đơn hàng đến khách hàng qua email hoặc thông báo trên trang web để thông báo rằng giao dịch đã thành công.
22. Theo Dõi Trạng Thái Đơn Hàng: Cung cấp cho khách hàng khả năng theo dõi trạng thái đơn hàng, từ khi đặt hàng cho đến khi sản phẩm được giao đến.
23. Hỗ Trợ Thanh Toán An Toàn: Đảm bảo tính an toàn của quá trình thanh toán bằng cách sử dụng kỹ thuật mã hóa và các biện pháp bảo mật khác.
24. Phát Triển Hệ Thống Giả Hàng Thân Thiết: Phát triển tính năng giả hàng thân thiết để lưu trữ thông tin giả hàng của khách hàng và tạo trải nghiệm mua sắm liền mạch trên nhiều thiết bị.
25. Tích Hợp Hệ Thống Thanh Toán Bên Ngoài: Tích hợp với cổng thanh toán bên ngoài như PayPal, Stripe, hoặc các dịch vụ thanh toán trực tuyến khác để xử lý thanh toán.

3.1.2.4 Nghệp vụ thanh toán

1. Lựa Chọn Phương Thức Thanh Toán: Cho phép khách hàng chọn phương thức thanh toán, bao gồm thẻ tín dụng, thẻ ghi nợ, ví điện tử, chuyển khoản ngân hàng, hoặc các phương thức thanh toán trực tuyến khác.
2. Nhập Thông Tin Thanh Toán: Khách hàng cần cung cấp thông tin thanh toán, bao gồm số thẻ, ngày hết hạn, mã bảo mật, và thông tin liên quan để thực hiện thanh toán.
3. Xác Thực Thanh Toán: Hệ thống cần thực hiện xác thực thông tin thanh toán để đảm bảo tính xác thực và bảo mật giao dịch.
4. Ghi Nhận Giao Dịch: Ghi nhận thông tin giao dịch, bao gồm số tiền, sản phẩm hoặc dịch vụ mua sắm, và thông tin liên quan về giao dịch.
5. Phân Loại Giao Dịch: Phân loại giao dịch thành các loại khác nhau, ví dụ như thanh toán cho đơn hàng, đặt cọc, hoàn tiền, và các loại giao dịch tài chính khác.
6. Xử Lý Thanh Toán: Gửi thông tin thanh toán đến cơ quan xử lý thanh toán hoặc cổng thanh toán để thực hiện giao dịch tài chính.
7. Theo Dõi Trạng Thái Thanh Toán: Theo dõi trạng thái thanh toán để biết khi nào giao dịch đã hoàn thành và tiền đã được chuyển đến tài khoản người bán.
8. Xác Nhận Thanh Toán: Gửi xác nhận thanh toán đến khách hàng để thông báo rằng giao dịch đã thành công.
9. Bảo Mật Thanh Toán: Đảm bảo tính an toàn và bảo mật thông tin thanh toán bằng cách sử dụng mã hóa và các biện pháp bảo mật khác.
10. Phí Vận Chuyển và Thuế: Tính toán và cộng thêm phí vận chuyển và thuế vào tổng số tiền thanh toán nếu áp dụng.
11. Lưu Trữ Lịch Sử Thanh Toán: Lưu trữ lịch sử các giao dịch thanh toán và cung cấp cho khách hàng truy cập vào lịch sử thanh toán của họ.
12. Phát Triển Chính Sách Thanh Toán: Xây dựng và quản lý chính sách liên quan đến thanh toán, bao gồm chính sách đổi trả và hoàn tiền.

3.2 Usecase diagram và đặc tả use-case

3.2.1 Usecase hệ thống

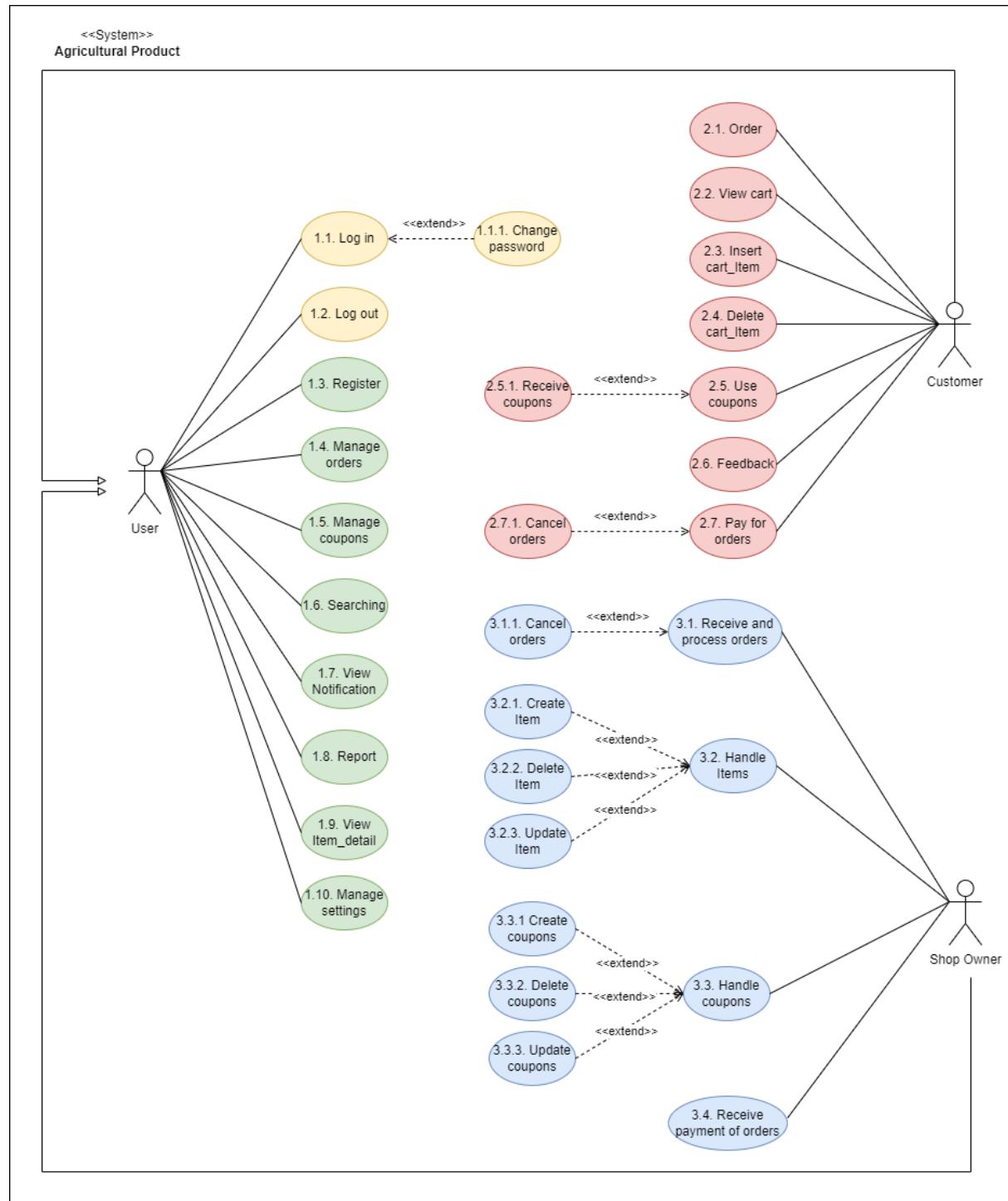


Figure 3.1: Use-case hệ thống



3.2.2 ĐẶC TẢ USECASE

3.2.2.1 Đăng nhập

Use Case ID	UC-1.1
Use Case Name	Log in
Actor(s)	Khách hàng, chủ cửa hàng
Description	Đăng nhập vào hệ thống
Priority	Cao
Trigger	Người dùng muốn đăng nhập vào hệ thống
Pre-Conditions	Tài khoản người dùng đã được tạo sẵn. 1. Tài khoản người dùng đã được phân quyền. 2. Thiết bị của người dùng đã được kết nối internet.
Post-Conditions	Người dùng đăng nhập hệ thống thành công. Hệ thống ghi nhận đăng nhập thành công vào Activity Log.
Basic flow	1. Hệ thống hiển thị giao diện đăng nhập 2. Người dùng nhập thông tin đăng nhập vào các trường tương ứng trên giao diện và chọn nút đăng nhập 3. Hệ thống kiểm tra thông tin đăng nhập 4. Hệ thống thông báo đăng nhập thành công 5. Hệ thống hiển thị giao diện màn hình chính
Alternative flow	Alternative 2.1-1: tại bước 2 2a. Người dùng chọn phương thức đăng nhập bằng tài khoản của bên thứ ba 2a1. Hệ thống chuyển sang màn hình đăng nhập của bên thứ ba 2a2. Người dùng nhập tài khoản và chọn lệnh đăng nhập 3a. Bên thứ ba xác thực thông tin đăng nhập thành công và cho phép người dùng truy cập ứng dụng <i>Tiếp tục bước 4</i>
Exception flow	Exception 2.1-1: tại bước 3 1a. Nếu không tìm thấy tài khoản, sai mật khẩu, hệ thống thông báo cho người dùng <i>Quay lại bước 4</i>
Extended points	Change password
Business Rules	BR2.1-1: Thông tin đăng nhập phải được xác thực với cơ sở dữ liệu người dùng để đảm bảo tính chính xác.
Non-Functional Requirement	NFR2.1-1: Bảo mật - Thông tin đăng nhập phải được bảo vệ một cách an toàn và chỉ có người dùng có quyền truy cập vào tài khoản

Table 3.1: Use case Đăng nhập



3.2.2.2 Đổi mật khẩu

Use Case ID	UC-1.1.1
Use Case Name	Change password
Actor(s)	Khách hàng, chủ cửa hàng
Description	Thay đổi mật khẩu của tài khoản
Priority	Cao
Trigger	Người dùng quên mật khẩu tài khoản đăng nhập và muốn đăng nhập vào hệ thống
Pre-Conditions	Tài khoản người dùng đã được tạo sẵn. 1. Tài khoản người dùng đã được phân quyền. 2. Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập.
Post-Conditions	Người dùng đổi mật khẩu thành công. Hệ thống ghi nhận lại sự thay đổi mật khẩu vào cơ sở dữ liệu.
Basic flow	1. Hệ thống hiển thị giao diện thay đổi mật khẩu 2. Người dùng nhập thông tin cần thiết (mật khẩu hiện tại, mật khẩu mới và xác nhận mật khẩu mới) 3. Người dùng ấn vào nút "Thay đổi mật khẩu" 4. Hệ thống kiểm tra mật khẩu hiện tại 5. Hệ thống kiểm tra mật khẩu mới và xác nhận mật khẩu mới để đảm bảo tính hợp lệ 6. Hệ thống thông báo đổi mật khẩu thành công 7. Hệ thống quay lại trang đăng nhập
Alternative flow	Alternative 2.1.1-1: Tại bước 3 Nếu người dùng muốn hủy bỏ thay đổi mật khẩu: 3a. Người dùng ấn vào nút "Quay lại" trên giao diện thay đổi mật khẩu Alternative 2.1.1-2: Tại bước 4 Nếu mật khẩu hiện tại không chính xác: 4a. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại mật khẩu <i>Quay lại bước 2</i> Alternative 2.1.1-3: Tại bước 5 Nếu mật khẩu không đúng quy định hoặc mật khẩu mới và xác nhận mật khẩu mới không khớp: 5a. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin thay đổi mật khẩu <i>Quay lại bước 2</i>
Exception flow	Không



Business Rules	BR2.1.1-1: Mật khẩu phải đáp ứng các yêu cầu về độ dài, ký tự đặc biệt, chữ hoa, chữ thường, số, ...
Non-Functional Requirement	NFR2.1.1-1: Bảo mật - Mật khẩu của người dùng phải được bảo vệ một cách an toàn trong quá trình thay đổi và lưu trữ

Table 3.2: Use case *Đổi mật khẩu*

3.2.2.3 Đăng xuất

Use Case ID	UC-1.2
Use Case Name	Log out
Actor(s)	Khách hàng, chủ cửa hàng
Description	Đăng xuất khỏi hệ thống
Priority	Cao
Trigger	Người dùng muốn đăng xuất khỏi hệ thống
Pre-Conditions	Actor đã đăng nhập và đang ở màn hình chính.
Post-Conditions	Tài khoản được đăng xuất khỏi hệ thống
Basic Flow	<ol style="list-style-type: none">Người dùng ấn vào icon tài khoản trên thanh navbarHệ thống hiển thị một popup nhỏ với nhiều tùy chọn khác nhau cho tài khoảnNgười dùng chọn đăng xuấtHệ thống xóa phiên làm việcHệ thống quay trở lại trang đăng nhập
Alternative flow	Alternative 2.2-1: tại bước 3 1a. Nếu người dùng chọn hủy đăng xuất, quay lại màn hình chính
Exception Flow	Không
Business Rules	Không
Non-Functional Requirement	NFR2.2-1: Bảo mật - Hệ thống phải đảm bảo rằng người dùng được đăng xuất an toàn và thông tin cá nhân của họ không bị lộ khi đăng xuất NFR2.2-1: Giao diện người dùng - Giao diện đăng xuất phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng

Table 3.3: Use case *Đăng xuất*



3.2.2.4 Đăng ký

Use Case ID	UC-1.3
Use Case Name	Register
Actor(s)	Khách hàng, chủ cửa hàng
Description	Người dùng đăng ký tài khoản mới
Priority	Cao
Trigger	Người dùng có nhu cầu muốn tạo tài khoản
Pre-Conditions	Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng ký tài khoản.
Post-Conditions	Người dùng đăng ký thành công
Basic flow	<ol style="list-style-type: none">Người dùng truy cập vào trang đăng ký trong hệ thốngHệ thống hiển thị giao diện đăng ký tài khoảnNgười dùng nhập thông tin cá nhân vào các trường tương ứng trên giao diệnNgười dùng ấn vào nút "Đăng ký"Hệ thống kiểm tra tính hợp lệ của thông tin đăng kýHệ thống gửi email xác thực đến địa chỉ email đã được người dùng cung cấpNgười dùng kiểm tra hộp thư đến và nhấp vào liên kết xác thực trong emailHệ thống xác minh email và thông báo đăng ký thành công
Alternative flow	Alternative 2.3-1: tại bước 3 3a. Người dùng chọn đăng ký bằng tài khoản có sẵn của bên thứ ba 3a1. Hệ thống chuyển sang màn hình đăng nhập của bên thứ ba đã chọn 3a2. Người dùng nhập đăng nhập vào tài khoản 3a3. Bên thứ ba xác nhận lại yêu cầu liên kết tài khoản 4a. Người dùng nhấn nút xác nhận liên kết tài khoản 5a. Hệ thống thông báo đăng ký thành công
Exception flow	Exception 2.3-1: tại bước 5 5a. Nếu thông tin đăng ký tài khoản không hợp lệ, thông báo cho người dùng <i>Quay lại bước 3</i>
Business Rules	BR2.3-1: Mật khẩu phải đáp ứng các yêu cầu về độ dài, ký tự đặc biệt, chữ hoa, chữ thường, số, ...



Non-Functional Requirement	NFR2.3-1: Bảo mật - Hệ thống phải đảm bảo an toàn thông tin cá nhân của người dùng trong quá trình đăng ký NFR2.3-2: Giao diện người dùng - Giao diện đăng ký phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng
-----------------------------------	---

Table 3.4: Use case Đăng ký

3.2.2.5 Quản lý các đơn đặt hàng

Use Case ID	UC-1.4
Use Case Name	Manage orders
Actor(s)	Khách hàng, chủ cửa hàng
Description	Hệ thống sẽ cung cấp cho người dùng danh sách các đơn đặt hàng theo thứ tự từ mới nhất đến cũ nhất
Priority	Cao
Trigger	Người dùng muốn xem danh sách các đơn đặt hàng
Pre-Conditions	Actor đã đăng nhập thành công
Post-Conditions	Hệ thống trả về danh sách các đơn đặt hàng
Basic Flow	1. Actor truy cập vào trang quản lý đơn hàng 2. Hệ thống hiển thị cho Actor danh sách các đơn đặt hàng theo thứ tự từ mới nhất đến cũ nhất
Alternative Flow	Không
Exception Flow	Không
Business Rules	BR2.4-1: Chỉ khách hàng và chủ cửa hàng mới có quyền truy cập và quản lý đơn hàng của mình trong hệ thống
Non-Functional Requirement	NFR2.4-1: Bảo mật - Hệ thống phải đảm bảo an toàn thông tin đơn hàng và chỉ cho phép người có quyền truy cập chỉnh sửa đơn hàng NFR2.4-2: Giao diện người dùng - Giao diện quản lý đơn hàng phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho Actor

Table 3.5: Use case Quản lý các đơn đặt hàng



3.2.2.6 Quản lý các phiếu ưu đãi

Use Case ID	UC-1.5
Use Case Name	Manage coupons
Actor(s)	Khách hàng, chủ cửa hàng
Description	Hệ thống sẽ cung cấp danh sách các phiếu ưu đãi mà người dùng đang sở hữu
Priority	Cao
Trigger	Người dùng muốn xem các phiếu ưu đãi mà mình đang sở hữu
Pre-Conditions	Actor đã đăng nhập thành công
Post-Conditions	Hệ thống trả về danh sách các phiếu ưu đãi
Basic Flow	1. Actor truy cập vào trang quản lý phiếu ưu đãi 2. Hệ thống hiển thị cho Actor danh sách các phiếu ưu đãi mà actor đang sở hữu.
Alternative Flow	Không
Exception Flow	Không
Business Rules	BR2.5-1: Chỉ khách hàng và chủ cửa hàng mới có quyền truy cập và quản lý phiếu ưu đãi của mình trong hệ thống
Non-Functional Requirement	NFR2.5-1: Giao diện người dùng - Giao diện quản lý phiếu ưu đãi phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho Actor

Table 3.6: Use case Quản lý các phiếu ưu đãi



3.2.2.7 Tìm kiếm

Use Case ID	UC-1.6
Use Case Name	Searching
Actor(s)	Khách hàng, Chủ cửa hàng
Description	Hệ thống cung cấp khả năng tìm kiếm sản phẩm hoặc cửa hàng trong hệ thống: tên sản phẩm, danh mục, tên cửa hàng,
Priority	Cao
Trigger	Người dùng muốn tìm kiếm
Pre-Conditions	Người dùng đã truy cập thành công vào hệ thống
Post-Conditions	Hệ thống trả về các kết quả của thông tin cần tìm
Basic Flow	<ol style="list-style-type: none">Actor nhập nội dung lên thanh tìm kiếmSau khi nhập nội dung, nhấn Enter hoặc nhấn nút "Tìm kiếm"Hệ thống tìm và lọc sản phẩm theo thông tin được cung cấpHệ thống hiển thị cho Actor danh sách các sản phẩm hoặc cửa hàng có liên quanActor chọn sản phẩm hoặc cửa hàng mong muốn
Alternative Flow	Alternative 2.6-1: tại bước 4 Nếu không có kết quả tìm kiếm 4a. Hệ thống hiển thị thông báo không tìm thấy kết quả phù hợp <i>Quay lại bước 1</i>
Exception Flow	Không
Business Rules	BR2.6-1: Kết quả tìm kiếm được hiển thị dựa trên thông tin nhập và tùy chọn tìm kiếm ưu tiên của actor (khoảng cách, lượt mua, đánh giá,...)
Non-Functional Requirement	NFR2.6-1: Hiệu suất - Hệ thống phải xử lý nhanh chóng và hiệu quả yêu cầu tìm kiếm NFR2.6-2: Giao diện người dùng - Giao diện tìm kiếm phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng

Table 3.7: Use case Tìm kiếm



3.2.2.8 Thông báo

Use Case ID	UC-1.7
Use Case Name	View Notification
Actor(s)	Khách hàng, chủ cửa hàng
Description	Xem danh sách các thông báo về các thông tin liên quan
Priority	Cao
Trigger	Người dùng muốn nhận thông báo
Pre-Conditions	Actor đã đăng nhập thành công
Post-Conditions	Hệ thống trả về danh sách các thông báo liên quan đến người dùng.
Basic Flow	<ol style="list-style-type: none">Người dùng truy cập vào trang thông báoHệ thống hiển thị danh sách các thông báo mới và cũ cho người dùngNgười dùng xem nội dung của mỗi thông báo trong danh sáchNgười dùng kết thúc xem thông báo và đóng danh sách thông báo
Alternative Flow	Alternative 2.8-1: tại bước 2 Nếu không có thông báo nào: 2a. Hệ thống hiển thị thông tin không có thông báo nào <i>Tiếp tục bước 5</i>
Exception Flow	Không
Business Rules	BR2.8-1: Người dùng có thể đánh dấu thông báo đã đọc và xóa thông báo không cần thiết
Non-Functional Requirement	NFR2.8-1: Bảo mật - Hệ thống phải đảm bảo an toàn thông tin của thông báo và chỉ cho phép người dùng có quyền truy cập xem thông báo

Table 3.8: Use case Xem thông báo



3.2.2.9 Báo cáo

Use Case ID	UC-1.8
Use Case Name	Report
Actor(s)	Khách hàng, Chủ cửa hàng
Description	Hệ thống cung cấp khả năng báo cáo sự cố kỹ thuật của trang web hoặc báo cáo về tài khoản giả mạo
Priority	Cao
Trigger	Người dùng muốn báo cáo về 1 tài khoản hoặc sự cố kỹ thuật của trang web
Pre-Conditions	Actor đã đăng nhập thành công
Post-Conditions	Hệ thống trả về báo cáo thành công
Basic Flow	<ol style="list-style-type: none">Actor truy cập vào trang báo cáoHệ thống hiển thị giao diện để người dùng nhập thông tin báo cáoNgười dùng nhập nội dung của bản báo cáoNgười dùng nhấn nút gửi báo cáoHệ thống ghi nhận nội dung báo cáo và trả về thông báo Báo cáo thành công.
Alternative Flow	Alternative 2.9-1: tại bước 4 4a. Nếu người dùng không muốn gửi báo cáo nữa, có thể nhấn nút "Quay lại" hoặc chuyển hướng sang trang khác
Exception Flow	Không
Business Rules	Không
Non-Functional Requirement	NFR2.9-1: Giao diện người dùng - Giao diện báo cáo phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng NFR2.9-2: Bảo mật - Hệ thống phải đảm bảo an toàn thông tin của báo cáo và chỉ cho phép những người có liên quan có quyền truy cập xem báo cáo

Table 3.9: Use case Báo cáo



3.2.2.10 Xem chi tiết sản phẩm

Use Case ID	UC-1.9
Use Case Name	View Item_detail
Actor(s)	Khách hàng, chủ cửa hàng
Description	Người dùng có thể xem chi tiết của sản phẩm: tên sản phẩm, nguồn gốc xuất xứ, giá, hình ảnh, mô tả,...
Priority	Cao
Trigger	Người dùng muốn xem thông tin chi tiết của sản phẩm
Pre-Conditions	Người dùng đã truy cập vào nền tảng và tìm thấy thẻ sản phẩm cần xem
Post-Conditions	Hệ thống trả về thông tin của sản phẩm mà người dùng muốn xem
Basic Flow	1. Người dùng nhấn chọn một sản phẩm cụ thể 2. Hệ thống lấy thông tin của sản phẩm 3. Hệ thống hiển thị trang thông tin chi tiết về sản phẩm
Alternative Flow	Không
Exception Flow	Không
Business Rules	Không
Non-Functional Requirement	NFR2.11-1: Giao diện người dùng - Giao diện xem chi tiết sản phẩm phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng.

Table 3.10: Use case Xem chi tiết sản phẩm



3.2.2.11 Quản lý Cài đặt

Use Case ID	UC-1.10
Use Case Name	Manage Setting
Actor(s)	Khách hàng, chủ cửa hàng
Description	Hệ thống cung cấp các cài đặt chi tiết mà người dùng có thể tùy chọn và thay đổi
Priority	Cao
Trigger	Người dùng muốn thay đổi cài đặt
Pre-Conditions	Actor đã đăng nhập thành công
Post-Conditions	Hệ thống thiết lập lại theo cài đặt của người dùng
Basic Flow	<ol style="list-style-type: none">Người dùng truy cập vào trang Cài đặtHệ thống hiển thị danh sách các cài đặt mà người dùng có thể tùy chọn và thay đổiNgười dùng thực hiện thay đổi cài đặtNgười dùng nhấn Lưu thay đổiHệ thống lưu trữ các thay đổi và cập nhật cài đặt mới
Alternative Flow	Alternative 2.10-1: tại bước 4 4a. Nếu người dùng không muốn thay đổi cài đặt nữa, có thể nhấn vào nút "Hủy" hoặc chuyển hướng sang trang khác
Exception Flow	Không
Business Rules	Không
Non-Functional Requirement	NFR2.10-1: Giao diện người dùng - Giao diện quản lý cài đặt phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng NFR2.10-2: Đồng bộ hóa - Hệ thống phải đồng bộ với các nền tảng và thiết bị khác nhau để cài đặt được áp dụng đồng bộ trên tất cả thiết bị của cùng một người dùng

Table 3.11: Use case Quản lý cài đặt



3.2.2.12 Đặt hàng

Use Case ID	UC-2.1
Use Case Name	Order
Actor(s)	Khách hàng
Description	Đặt hàng
Priority	Cao
Trigger	Người dùng muốn mua sản phẩm
Pre-Conditions	Actor đã đăng nhập thành công
Post-Conditions	Hệ thống lên đơn hàng và gửi đến chủ cửa hàng tương ứng
Basic Flow	<ol style="list-style-type: none">Khách hàng chọn sản phẩm muốn mua trong giỏ hàngKhách hàng xác nhận các thông tin khác liên quan đến việc thanh toán và thông tin giao hàngHệ thống xác nhận thông tin đặt hàng và tính toán tổng số tiền phải thanh toánKhách hàng bấm nút đặt hàngHệ thống xác nhận và xử lý thanh toánHệ thống hiển thị xác nhận đơn hàng cho khách hàng, bao gồm thông tin chi tiết về đơn hàng, số lượng, giá cả, phương thức thanh toán, và địa chỉ giao hàng
Alternative Flow	Alternative 3.1-1: tại bước 1 Nếu sản phẩm không còn hàng hoặc không có đủ số lượng 1a. Hệ thống hiển thị thông báo cho khách hàng biết rằng sản phẩm đã hết hàng hoặc không có đủ số lượng
Exception Flow	Không
Business Rules	BR3.1-1: Vị trí giao hàng phải được xác định trên map để chắc chắn có tồn tại BR3.1-2: Nếu chọn thanh toán trả trước, phải hoàn tất thanh toán trong 24 giờ
Non-Functional Requirement	NFR3.1-1: Giao diện người dùng - Giao diện đặt hàng phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho khách hàng NFR3.1-2: Bảo mật - Thông tin thanh toán và thông tin khách hàng phải được bảo vệ an toàn và không bị lộ ra ngoài NFR3.1-3: Khả dụng (Availability) - Hệ thống cần có khả năng hoạt động liên tục và sẵn sàng sử dụng để khách hàng có thể đặt hàng bất kỳ lúc nào

Table 3.12: Use case Đặt hàng



3.2.2.13 Xem giỏ hàng

Use Case ID	UC-2.2
Use Case Name	View cart
Actor(s)	Khách hàng
Description	Người dùng có thể xem danh sách các món hàng đã được thêm vào giỏ hàng của mình
Priority	Cao
Trigger	Người dùng muốn xem giỏ hàng
Pre-Conditions	<ul style="list-style-type: none">– Actor đã đăng nhập thành công– Actor đã thêm sản phẩm vào giỏ hàng
Post-Conditions	Hệ thống trả về danh sách các món hàng trong giỏ hàng
Basic Flow	<ol style="list-style-type: none">1. Actor chọn biểu tượng giỏ hàng Giỏ hàng trên thanh navbar2. Hệ thống lấy thông tin về giỏ hàng3. Hệ thống hiển thị cho Actor danh sách các món hàng có trong giỏ hàng.
Alternative Flow	Alternative 3.3-1: tại bước 3 3a. Nếu giỏ hàng trống, hệ thống hiển thị trang xem giỏ hàng với thông báo rằng giỏ hàng trống
Exception Flow	Không
Business Rules	Không
Non-Functional Requirement	NFR3.3-1: Giao diện người dùng - Giao diện xem giỏ hàng phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho khách hàng

Table 3.13: Use case Xem giỏ hàng



3.2.2.14 Thêm vào giỏ hàng

Use Case ID	UC-2.3
Use Case Name	Insert cart_Item
Actor(s)	Khách hàng
Description	Khách hàng có thể thêm 1 sản phẩm bất kì vào giỏ hàng của mình
Priority	Cao
Trigger	Người dùng muốn thêm sản phẩm vào giỏ hàng
Pre-Conditions	Actor đã đăng nhập thành công
Post-Conditions	Hệ thống ghi nhận sản phẩm vào giỏ hàng của người dùng
Basic Flow	<ol style="list-style-type: none">Actor chọn một sản phẩm muốn thêm vào giỏ hàngActor nhấn Thêm vào giỏ hàngHệ thống ghi nhận sản phẩm vào giỏ hàng của actorHệ thống cập nhật tổng số lượng sản phẩm trên icon giỏ hàng của actor
Alternative Flow	Alternative 3.4-1: tại bước 3 3a. Hệ thống hiển thị thông báo cho người dùng biết rằng sản phẩm không còn đủ số lượng 3a1. Người dùng có thể điều chỉnh số lượng sản phẩm hoặc tiếp tục mua sắm
Exception Flow	Không
Business Rules	BR3.4-1: Sản phẩm chỉ có thể được thêm vào giỏ hàng nếu số lượng sản phẩm đủ trong kho
Non-Functional Requirement	NFR3.4-1: Giao diện người dùng - Giao diện thêm sản phẩm vào giỏ hàng phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng

Table 3.14: Use case Thêm vào giỏ hàng



3.2.2.15 Xoá khỏi giỏ hàng

Use Case ID	UC-2.4
Use Case Name	Delete cart_Item
Actor(s)	Khách hàng
Description	Xoá sản phẩm khỏi giỏ hàng
Priority	Cao
Trigger	Người dùng muốn xoá sản phẩm không cần thiết nữa ra khỏi giỏ hàng
Pre-Conditions	Actor đã đăng nhập thành công
Post-Conditions	Hệ thống cập nhật lại giỏ hàng sau khi đã xoá
Basic Flow	<ol style="list-style-type: none">Actor chọn Giỏ hàngActor di chuyển tới mục sản phẩm muốn xoáActor nhấn biểu tượng thùng rác.Hệ thống cập nhật lại giỏ hàng sau khi xoáHệ thống hiển thị xoá thành công.
Alternative Flow	Không
Exception Flow	Không
Business Rules	Không
Non-Functional Requirement	NFR3.5-1: Giao diện người dùng - Giao diện xóa sản phẩm trong giỏ hàng phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng

Table 3.15: Use case Xoá khỏi giỏ hàng



3.2.2.16 Sử dụng phiếu ưu đãi

Use Case ID	UC-2.5
Use Case Name	Use coupons
Actor(s)	Khách hàng
Description	Khách hàng sử dụng phiếu ưu đãi để được ưu đãi lúc thanh toán
Priority	Cao
Trigger	Người dùng muốn được ưu đãi khi thanh toán
Pre-Conditions	Actor đang chọn các món hàng để tạo đơn đặt hàng
Post-Conditions	Đơn đặt hàng của người dùng được áp dụng phiếu ưu đãi và được ưu đãi
Basic Flow	<ol style="list-style-type: none">Actor chọn mục Sử dụng phiếu ưu đãiActor chọn phiếu ưu đãi phù hợpHệ thống xem xét phiếu ưu đãi có hợp lệ hay khôngHệ thống chấp nhận phiếu ưu đãi, đơn hàng được ưu đãiActor nhấn đặt hàngHệ thống ghi nhận phiếu ưu đãi đã được sử dụng và loại bỏ khỏi kho phiếu ưu đãi.
Alternative Flow	Alternative 3.6-1: tại bước 4 Nếu phiếu ưu đãi không hợp lệ: 4a. Hệ thống hiển thị thông báo lỗi cho người dùng, cho biết mã phiếu ưu đãi không hợp lệ 4a1. Người dùng có thể thử lại hoặc tiếp tục mua sắm mà không sử dụng phiếu ưu đãi Alternative 3.6-2: tại bước 46 Nếu tại thời điểm đặt hàng phiếu ưu đãi đã hết lượt sử dụng 6b. Hệ thống hiển thị thông báo lỗi cho người dùng, cho biết mã phiếu ưu đãi đã hết lượt sử dụng 6b1. Người dùng có thể thử lại hoặc tiếp tục mua sắm mà không sử dụng phiếu ưu đãi
Exception Flow	Không
Extended points	Nhận phiếu ưu đãi
Business Rules	BR3.6-1: Phiếu ưu đãi chỉ có thể được sử dụng khi người dùng đã đăng nhập thành công BR3.6-2: Mã phiếu ưu đãi phải hợp lệ và chưa hết hạn



Non-Functional Requirement	NFR3.6-1: Khả năng mở rộng - Hệ thống phải có khả năng mở rộng để xử lý việc sử dụng phiếu ưu đãi cho hàng ngàn khách hàng cùng một lúc mà không ảnh hưởng đến hiệu suất NFR3.6-1: Giao diện người dùng - Giao diện sử dụng phiếu ưu đãi phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng
-----------------------------------	--

Table 3.16: Use case Sử dụng phiếu ưu đãi

3.2.2.17 Nhận phiếu ưu đãi

Use Case ID	UC-2.5.1
Use Case Name	Receive coupons
Actor(s)	Khách hàng
Description	Khách hàng có thể lưu phiếu ưu đãi trong lúc đang xem thông tin các sản phẩm.
Priority	Cao
Trigger	phiếu ưu đãi hấp dẫn hiện lên trong lúc người dùng đang tìm kiếm sản phẩm
Pre-Conditions	Actor đang xem thông tin các sản phẩm
Post-Conditions	Hệ thống ghi nhận phiếu ưu đãi vừa lưu vào kho phiếu ưu đãi của người dùng
Basic Flow	1. Actor chọn một sản phẩm bất kỳ để xem thông tin chi tiết 2. Actor lướt xem sản phẩm 3. Hệ thống hiển thị các phiếu ưu đãi đang có của cửa hàng đó. 4. Người dùng nhấn Lưu để lưu phiếu ưu đãi vào kho phiếu của mình.
Alternative Flow	Không
Exception Flow	Không
Business Rules	BR3.6.1-1: Người dùng chỉ có thể nhận được phiếu ưu đãi khi đã đăng nhập thành công.
Non-Functional Requirement	NFR3.6.1-1: Tính nhất quán - Hệ thống phải đảm bảo tính nhất quán khi gửi và nhận phiếu ưu đãi, đảm bảo rằng phiếu ưu đãi chỉ được gửi và nhận một lần duy nhất NFR3.6.1-2: Khả năng mở rộng - Hệ thống phải có khả năng mở rộng để xử lý việc nhận và gửi phiếu ưu đãi cho hàng ngàn người dùng cùng một lúc mà không ảnh hưởng đến hiệu suất

Table 3.17: Use case Nhận phiếu ưu đãi



3.2.2.18 Đánh giá

Use Case ID	UC-2.6
Use Case Name	Feedback
Actor(s)	Khách hàng
Description	Khách hàng có thể đánh giá trải nghiệm của mình khi mua 1 món hàng
Priority	Cao
Trigger	Người dùng muốn đánh giá sản phẩm
Pre-Conditions	Actor đã mua hàng thành công
Post-Conditions	Hệ thống ghi nhận đánh giá sản phẩm của khách hàng
Basic Flow	<ol style="list-style-type: none">Actor chọn Đơn hàng đã được giao thành côngActor chọn phần Đánh giá sản phẩmActor nhập nội dung đánh giáHệ thống ghi nhận đánh giáHệ thống thông báo đánh giá thành công
Alternative Flow	Không
Exception Flow	Không
Business Rules	BR3.7-1: Người dùng chỉ có thể đánh giá khi đơn hàng đã được giao thành công BR3.7-2: Các trường thông tin bắt buộc trong phản hồi phải được điền đầy đủ
Non-Functional Requirement	NFR3.7-1: Tính nhất quán - Hệ thống phải đảm bảo tính nhất quán trong việc ghi nhận và theo dõi phản hồi từ người dùng NFR3.7-2: Giao diện người dùng - Giao diện gửi phản hồi phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng

Table 3.18: Use case Đánh giá



3.2.2.19 Thanh toán

Use Case ID	UC-2.7
Use Case Name	Pay for orders
Actor(s)	Khách hàng
Description	Người dùng có thể thanh toán các đơn hàng đã đặt trên hệ thống bằng các phương thức thanh toán khác nhau: thanh toán bằng COD, thanh toán bằng ví điện tử,....
Priority	Cao
Trigger	Người dùng muốn thanh toán cho đơn hàng
Pre-Conditions	<ul style="list-style-type: none">- Actor đã đăng nhập thành công- Actor đang chọn các món hàng để tạo đơn đặt hàng
Post-Conditions	Trạng thái thanh toán của đơn hàng được cập nhật trong hệ thống
Basic Flow	<ol style="list-style-type: none">1. Người dùng chọn một trong các phương thức thanh toán2. Hệ thống ghi nhận thông tin phương thức thanh toán3. Hệ thống gửi yêu cầu thanh toán đến Cổng thanh toán (Payment Gateway)4. Cổng thanh toán xử lý yêu cầu thanh toán và chuyển hướng người dùng đến trang thanh toán của phương thức thanh toán được chọn5. Người dùng hoàn tất thanh toán trên trang thanh toán của phương thức thanh toán6. Cổng thanh toán xác nhận kết quả thanh toán và chuyển hướng người dùng trở lại hệ thống7. Hệ thống hiển thị thông báo thanh toán thành công và cập nhật trạng thái thanh toán của đơn hàng
Alternative Flow	<p>Alternative 3.8-1: tại bước 2 Nếu thông tin về phương thức thanh toán của người dùng chưa được lưu trữ từ trước:</p> <ol style="list-style-type: none">2a. Hệ thống yêu cầu người dùng nhập thông tin thanh toán2a1. Người dùng nhập thông tin thanh toán tương ứng và nhấn xác nhận2a2. Hệ thống yêu cầu người dùng xác thực lại phương thức thanh toán2a3. Người dùng cung cấp thông tin xác thực theo yêu cầu2a4. Hệ thống xác thực lại phương thức thanh toán <p>Alternative 3.8-1: tại bước 3 Nếu người dùng chọn các phương thức thanh toán COD:</p> <ol style="list-style-type: none">3b. Hệ thống xác nhận phương thức thanh toán của khách hàng4b. Hệ thống cập nhật trạng thái thanh toán của đơn hàng
Exception Flow	Không



Business Rules	BR3.8-1: Đối với các phương thức thanh toán trả trước, phương thức phải được xác thực trước khi được chọn
Non-Functional Requirement	NFR3.8-1: Giao diện người dùng - Giao diện người dùng phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng trong quá trình chọn phương thức thanh toán NFR3.8-2: Bảo mật - Hệ thống phải đảm bảo an toàn thông tin, bao gồm phương thức thanh toán, và không cho phép truy cập trái phép

Table 3.19: Use case Thanh toán



3.2.2.20 Huỷ đơn đặt hàng

Use Case ID	UC-2.7.1
Use Case Name	Cancel orders
Actor(s)	Khách hàng
Description	Khách hàng có thể hủy đơn đặt hàng chưa nhận
Priority	Cao
Trigger	Khách hàng muốn hủy đơn đặt hàng chưa nhận
Pre-Conditions	Actor đã đăng nhập thành công
Post-Conditions	Đơn đặt hàng được huỷ thành công và thông báo về việc huỷ đơn hàng được thông báo cho khách hàng
Basic Flow	<ol style="list-style-type: none">Khách hàng chọn tùy chọn "Hủy đơn hàng" ở đơn hàng đang trong tiến trìnhKhách hàng chọn lý do hủy đơn đặt hàng theo yêu cầu của hệ thốngHệ thống hiển thị xác nhận yêu cầu hủy đơn đặt hàng và yêu cầu khách hàng xác nhận hành động nàyKhách hàng xác nhận yêu cầu hủy đơn đặt hàngHệ thống hủy đơn đặt hàng và thông báo cho khách hàng về việc hủy thành công
Alternative Flow	Alternative 3.8.3-1: tại bước 4 Nếu người dùng từ chối xác nhận huỷ đơn đặt hàng: 4a. Hệ thống hiển thị thông báo rằng quá trình huỷ đơn đặt hàng bị hủy bỏ 4a1. Quá trình huỷ đơn đặt hàng kết thúc
Exception Flow	Exception 3.8.3-1: tại bước 1 Nếu đơn hàng đã được gửi đi hoặc đang trong quá trình vận chuyển: 1a. Hệ thống hiển thị thông báo lỗi rằng không thể huỷ đơn đặt hàng trong khi đơn hàng đang trong quá trình vận chuyển <i>Use case kết thúc</i>
Business Rules	BR3.8.3-1: Chỉ được hủy đơn hàng trước khi đơn hàng đã được gửi đi hoặc đang trong quá trình vận chuyển
Non-Functional Requirement	NFR3.8.3-1: Giao diện người dùng - Giao diện người dùng phải được thiết kế thân thiện, dễ sử dụng và cung cấp trải nghiệm tốt cho người dùng trong quá trình huỷ đơn hàng

Table 3.20: Use case Huỷ đơn đặt hàng



3.2.2.21 Nhận và xử lý đơn đặt hàng

Use Case ID	UC-3.1
Use Case Name	Receive and process orders
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng nhận và xử lý các đơn hàng từ khách hàng
Priority	Cao
Trigger	Chủ cửa hàng nhận được đơn hàng từ khách hàng
Pre-Conditions	Chủ cửa hàng đã đăng nhập vào hệ thống.
Post-Conditions	Các đơn hàng đã được chủ cửa hàng nhận và xử lý thành công
Basic Flow	<ol style="list-style-type: none">Chủ cửa hàng truy cập vào trang quản lý đơn hàng trong hệ thốngHệ thống hiển thị danh sách các đơn hàng đang chờ xử lýChủ cửa hàng chọn một đơn hàng để xem thông tin chi tiếtHệ thống hiển thị thông tin chi tiết về đơn hàngChủ cửa hàng xem xét thông tin đơn hàng và quyết định xử lý đơn hàng theo các hành động có sẵn (chấp nhận đơn hàng, từ chối đơn hàng hoặc liên hệ với khách hàng để yêu cầu thêm thông tin)Sau khi hoàn tất xử lý đơn hàng, chủ cửa hàng cập nhật trạng thái của đơn hàng thành "Đã xử lý" và thông báo cho khách hàng về trạng thái và thông tin vận chuyển của đơn hàng
Alternative Flow	Không
Exception Flow	Không
Business Rules	BR4.1-1: Chủ cửa hàng chỉ có quyền nhận và xử lý đơn hàng từ khách hàng BR4.1-2: Các đơn hàng phải được xử lý theo trình tự thời gian nhận được.
Non-Functional Requirement	NFR4.1-1: Bảo mật - Hệ thống phải bảo vệ thông tin đơn hàng và thông tin khách hàng khỏi truy cập trái phép và đảm bảo tính bảo mật của dữ liệu NFR4.1-3: Khả năng mở rộng - Hệ thống phải hỗ trợ xử lý nhiều đơn hàng cùng một lúc và có khả năng mở rộng để đáp ứng tải cao trong tương lai

Table 3.21: Use case Nhận và xử lý đơn hàng



3.2.2.22 Huỷ đơn hàng

Use Case ID	UC-3.1.1
Use Case Name	Cancel orders
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng hủy các đơn hàng đã được nhận từ khách hàng
Priority	Cao
Trigger	Chủ cửa hàng muốn hủy đơn hàng
Pre-Conditions	<ul style="list-style-type: none">– Chủ cửa hàng đã đăng nhập vào hệ thống– Đơn hàng đã được chủ cửa hàng nhận và chưa được xử lý hoặc giao hàng
Post-Conditions	Đơn hàng đã được hủy bởi chủ cửa hàng
Basic Flow	<ol style="list-style-type: none">1. Chủ cửa hàng truy cập vào trang quản lý đơn hàng trong hệ thống2. Hệ thống hiển thị danh sách các đơn hàng đang chờ xử lý3. Chủ cửa hàng chọn một đơn hàng để xem thông tin chi tiết4. Hệ thống hiển thị thông tin chi tiết về đơn hàng5. Chủ cửa hàng cập nhật trạng thái của đơn hàng thành "Đã hủy" và ghi rõ lý do hủy6. Hệ thống xác nhận yêu cầu hủy đơn của chủ cửa hàng7. Hệ thống thông báo cho khách hàng biết về việc hủy đơn hàng và cung cấp thông tin về việc hoàn trả tiền nếu có
Alternative Flow	Không
Exception Flow	Không
Business Rules	BR4.1.1-1: Chủ cửa hàng chỉ có quyền hủy các đơn hàng chưa được xử lý hoặc giao hàng BR4.1.1-2: Đơn hàng đã bị hủy không thể được khôi phục
Non-Functional Requirement	NFR4.1.1-1: Tính nhất quán - Hệ thống phải đảm bảo tính nhất quán trong việc cập nhật trạng thái và thông báo hủy đơn hàng cho khách hàng NFR4.1.1-1: Bảo mật - Hệ thống phải đảm bảo tính bảo mật của thông tin đơn hàng và thông tin khách hàng trong quá trình hủy đơn hàng

Table 3.22: Use case Huỷ đơn hàng



3.2.2.23 Quản lý sản phẩm

Use Case ID	UC-3.2
Use Case Name	Handle Items
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng quản lý sản phẩm trong cửa hàng
Priority	Cao
Trigger	Chủ cửa hàng muốn tạo sản phẩm mới
Pre-Conditions	Chủ cửa hàng đã đăng nhập vào hệ thống
Post-Conditions	Sản phẩm mới được thêm, xóa, sửa trong hệ thống
Basic Flow	<ol style="list-style-type: none">Chủ cửa hàng truy cập vào trang quản lý sản phẩmHệ thống hiển thị danh sách các sản phẩm hiện có trong cửa hàngChủ cửa hàng chọn tạo sản phẩm mới, thay đổi thông tin hoặc xóa sản phẩmChủ cửa hàng tiến hành lưu thay đổiHệ thống kiểm tra tính hợp lệ của thông tin thay đổiHệ thống thông báo cho chủ cửa hàng biết kết quả của thao tác quản lý sản phẩm
Alternative Flow	Không
Exception Flow	Exception 4.2-1: Nếu tài khoản của chủ cửa hàng bị vô hiệu hóa và không có quyền truy cập hoặc thực hiện thao tác quản lý sản phẩm, hệ thống sẽ hiển thị thông báo lỗi tương ứng và quá trình sẽ kết thúc
Business Rules	BR4.2-1: Chủ cửa hàng chỉ có quyền quản lý sản phẩm trong cửa hàng của mình BR4.2-2: Sản phẩm phải có giá trị và thuộc tính hợp lệ
Non-Functional Requirement	NFR4.2-1: Giao diện người dùng - Giao diện người dùng của quy trình quản lý sản phẩm phải được thiết kế một cách trực quan và thân thiện NFR4.2-2: Tính mở rộng - Hệ thống phải có khả năng mở rộng để hỗ trợ việc quản lý hàng nghìn sản phẩm trong cửa hàng mà không gây ảnh hưởng đến hiệu suất hoặc sự ổn định của hệ thống

Table 3.23: Use case Quản lý sản phẩm

3.2.2.24 Tạo sản phẩm

Use Case ID	UC-3.2.1
--------------------	----------



Use Case Name	Create Item
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng tạo sản phẩm mới để thêm vào cửa hàng
Priority	Cao
Trigger	Chủ cửa hàng muốn tạo sản phẩm mới
Pre-Conditions	Chủ cửa hàng đã đăng nhập vào hệ thống
Post-Conditions	Sản phẩm mới được tạo trong hệ thống
Basic Flow	<ol style="list-style-type: none">Chủ cửa hàng truy cập vào trang quản lý sản phẩmHệ thống hiển thị danh sách các sản phẩm hiện có trong cửa hàngChủ cửa hàng chọn tạo sản phẩm mớiHệ thống hiển thị form để chủ cửa hàng nhập thông tin chi tiết về sản phẩmChủ cửa hàng nhập đầy đủ thông tin chi tiết về sản phẩmChủ cửa hàng tiến hành lưu sản phẩm mớiHệ thống kiểm tra tính hợp lệ của thông tin và tạo sản phẩm mớiHệ thống thông báo cho chủ cửa hàng biết về việc tạo sản phẩm thành công và cập nhật sản phẩm lên nền tảng
Alternative Flow	Alternative 4.2-1: tại bước 7 7a.Nếu thông tin nhập vào không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu chủ cửa hàng nhập lại thông tin đúng <i>Quay lại bước 5</i>
Exception Flow	Exception 4.2-1: Nếu tài khoản của chủ cửa hàng bị vô hiệu hóa và không có quyền truy cập hoặc thực hiện thao tác quản tạo sản phẩm, hệ thống sẽ hiển thị thông báo lỗi tương ứng và quá trình tạo sản phẩm mới sẽ kết thúc
Business Rules	BR4.21-1: Chủ cửa hàng chỉ có quyền tạo sản phẩm trong cửa hàng của mình BR4.2-2: Tên sản phẩm phải là duy nhất trong cửa hàng BR4.2-3: Sản phẩm phải có giá trị và thuộc tính hợp lệ
Non-Functional Requirement	NFR4.2-1: Giao diện người dùng - Giao diện người dùng của quy trình tạo sản phẩm phải được thiết kế một cách trực quan và thân thiện, giúp chủ cửa hàng dễ dàng nhập thông tin và tạo sản phẩm NFR4.2-2: Tính mở rộng - Hệ thống phải có khả năng mở rộng để hỗ trợ việc tạo hàng ngàn sản phẩm trong cửa hàng mà không gây ảnh hưởng đến hiệu suất hoặc sự ổn định của hệ thống

Table 3.24: Use case Tạo sản phẩm

3.2.2.25 Xoá sản phẩm



Use Case ID	UC-3.2.2
Use Case Name	Delete Item
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng xóa một sản phẩm khỏi cửa hàng
Priority	Cao
Trigger	Chủ cửa hàng muốn xóa một sản phẩm khỏi cửa hàng
Pre-Conditions	Chủ cửa hàng đã đăng nhập vào hệ thống.
Post-Conditions	Sản phẩm được xóa khỏi cửa hàng và không còn hiển thị trong danh sách sản phẩm
Basic Flow	<ol style="list-style-type: none">Chủ cửa hàng truy cập vào trang quản lý sản phẩm trong hệ thốngHệ thống hiển thị danh sách các sản phẩm hiện có trong cửa hàngChủ cửa hàng nhấn chọn icon thùng rác trên sản phẩm cần xóaHệ thống hiển thị thông tin về sản phẩm và yêu cầu xác nhận xóaChủ cửa hàng xác nhận xóa sản phẩmHệ thống xóa sản phẩm khỏi cửa hàngHệ thống thông báo cho chủ cửa hàng biết về việc xóa sản phẩm thành công
Alternative Flow	Alternative 4.2.1-1: tại bước 5 5a. Nếu chủ cửa hàng không xác nhận xóa sản phẩm, quá trình xóa sản phẩm kết thúc và sản phẩm không bị xóa khỏi cửa hàng
Exception Flow	Exception 4.2.1-1: Nếu tài khoản của chủ cửa hàng bị vô hiệu hóa và không có quyền truy cập hoặc thực hiện thao tác xóa sản phẩm, hệ thống sẽ hiển thị thông báo lỗi tương ứng và quá trình xóa sản phẩm sẽ kết thúc
Business Rules	BR4.2.1-1: Chủ cửa hàng chỉ có quyền xóa sản phẩm trong cửa hàng của mình
Non-Functional Requirement	NFR4.2.1-1: Giao diện người dùng - Giao diện người dùng của quy trình xóa sản phẩm phải được thiết kế một cách trực quan và thân thiện, giúp chủ cửa hàng dễ dàng xóa sản phẩm

Table 3.25: Use case Xoá sản phẩm



3.2.2.26 Cập nhật sản phẩm

Use Case ID	UC-3.2.3
Use Case Name	Update Item
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng cập nhật thông tin của một sản phẩm trong cửa hàng
Priority	Cao
Trigger	Chủ cửa hàng muốn cập nhật thông tin của một sản phẩm trong cửa hàng
Pre-Conditions	Chủ cửa hàng đã đăng nhập vào hệ thống
Post-Conditions	Thông tin của sản phẩm được cập nhật thành công và hiển thị đúng trong danh sách sản phẩm
Basic Flow	<ol style="list-style-type: none">Chủ cửa hàng truy cập vào trang quản lý sản phẩm trong hệ thốngHệ thống hiển thị danh sách các sản phẩm hiện có trong cửa hàngChủ cửa hàng chọn sản phẩm cần cập nhật thông tinHệ thống hiển thị thông tin chi tiết về sản phẩm và cho phép chỉnh sửa các thông tinChủ cửa hàng cập nhật thông tin sản phẩmChủ cửa hàng xác nhận hoàn tất cập nhật thông tin sản phẩmHệ thống kiểm tra tính hợp lệ của thông tin và cập nhật thông tin sản phẩm trong cửa hàngHệ thống thông báo cho chủ cửa hàng biết về việc cập nhật thông tin sản phẩm thành công
Alternative Flow	Alternative 4.2.2-1: tại bước 6 6a. Nếu chủ cửa hàng không xác nhận hoàn tất cập nhật thông tin sản phẩm, quá trình cập nhật thông tin kết thúc và thông tin sản phẩm không thay đổi Alternative 4.2.2-2: tại bước 7 7a. Nếu kiểm tra thấy thông tin cập nhật không hợp lệ, hệ thống sẽ hủy cập nhật và thông báo cho chủ cửa hàng
Exception Flow	Exception 4.2.2-1: Nếu tài khoản của chủ cửa hàng bị vô hiệu hóa và không có quyền truy cập hoặc thực hiện thao tác cập nhật sản phẩm, hệ thống sẽ hiển thị thông báo lỗi tương ứng và quá trình cập nhật sản phẩm sẽ kết thúc
Business Rules	BR4.2.2-1: Chủ cửa hàng chỉ có quyền cập nhật thông tin sản phẩm trong cửa hàng của mình



Non-Functional Requirement	NFR4.2.2-1: Giao diện người dùng - Giao diện người dùng của quy trình cập nhật sản phẩm phải được thiết kế một cách trực quan và thân thiện, giúp chủ cửa hàng dễ dàng nhập thông tin và cập nhật sản phẩm
-----------------------------------	---

Table 3.26: Use case Cập nhật sản phẩm

3.2.2.27 Quản lý phiếu ưu đãi

Use Case ID	UC-3.3
Use Case Name	Handle Items
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng quản lý phiếu ưu đãi trong cửa hàng
Priority	Cao
Trigger	Chủ cửa hàng muốn tạo phiếu ưu đãi mới
Pre-Conditions	Chủ cửa hàng đã đăng nhập vào hệ thống
Post-Conditions	Phiếu ưu đãi mới được thêm, xóa, sửa trong hệ thống
Basic Flow	1. Chủ cửa hàng truy cập vào trang quản lý phiếu ưu đãi 2. Hệ thống hiển thị danh sách các phiếu ưu đãi hiện có trong cửa hàng 3. Chủ cửa hàng chọn tạo phiếu ưu đãi mới, thay đổi thông tin hoặc xóa phiếu ưu đãi 4. Chủ cửa hàng tiến hành lưu thay đổi 5. Hệ thống kiểm tra tính hợp lệ của thông tin thay đổi 6. Hệ thống thông báo cho chủ cửa hàng biết kết quả của thao tác quản lý phiếu ưu đãi
Alternative Flow	Không
Exception Flow	Exception 4.2-1: Nếu tài khoản của chủ cửa hàng bị vô hiệu hóa và không có quyền truy cập hoặc thực hiện thao tác quản lý phiếu ưu đãi, hệ thống sẽ hiển thị thông báo lỗi tương ứng và quá trình sẽ kết thúc
Business Rules	BR4.21-1: Chủ cửa hàng chỉ có quyền quản lý phiếu ưu đãi trong cửa hàng của mình BR4.2-2: Phiếu ưu đãi phải có giá trị và thuộc tính hợp lệ



Non-Functional Requirement	NFR4.2-1: Giao diện người dùng - Giao diện người dùng của quy trình quản lý phiếu ưu đãi phải được thiết kế một cách trực quan và thân thiện NFR4.2-2: Tính mở rộng - Hệ thống phải có khả năng mở rộng để hỗ trợ việc quản lý hàng nghìn phiếu ưu đãi trong cửa hàng mà không gây ảnh hưởng đến hiệu suất hoặc sự ổn định của hệ thống
-----------------------------------	--

Table 3.27: Use case Quản lý phiếu ưu đãi

3.2.2.28 Tạo phiếu ưu đãi

Use Case ID	UC-3.3.1
Use Case Name	Create coupons
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng tạo một phiếu ưu đãi để áp dụng cho sản phẩm trong cửa hàng
Priority	Cao
Trigger	Chủ cửa hàng muốn tạo một phiếu ưu đãi mới
Pre-Conditions	Chủ cửa hàng đã đăng nhập vào hệ thống
Post-Conditions	phiếu ưu đãi mới được tạo thành công và có thể sử dụng để áp dụng cho sản phẩm
Basic Flow	<ol style="list-style-type: none">Chủ cửa hàng truy cập vào trang quản lý phiếu ưu đãi trong hệ thốngHệ thống hiển thị danh sách các phiếu ưu đãi hiện có của cửa hàngChủ cửa hàng chọn tạo phiếu ưu đãi mớiHệ thống yêu cầu chủ cửa hàng cung cấp thông tin cho phiếu ưu đãiChủ cửa hàng cung cấp thông tin cho phiếu ưu đãi theo yêu cầuChủ cửa hàng xác nhận hoàn tất tạo phiếu ưu đãiHệ thống kiểm tra tính hợp lệ của thông tin và tạo phiếu ưu đãi mớiHệ thống thông báo cho chủ cửa hàng biết về việc tạo phiếu ưu đãi thành công và cập nhật lên nền tảng
Alternative Flow	Alternative 4.3-1: tại bước 6 6a. Nếu chủ cửa hàng không xác nhận hoàn tất tạo phiếu ưu đãi, quá trình tạo phiếu ưu đãi kết thúc và phiếu ưu đãi không được tạo



Exception Flow	Exception 4.3-1: Nếu tài khoản của chủ cửa hàng bị vô hiệu hóa và không có quyền truy cập hoặc thực hiện thao tác tạo phiếu ưu đãi, hệ thống sẽ hiển thị thông báo lỗi tương ứng và quá trình tạo phiếu ưu đãi sẽ kết thúc
Business Rules	BR4.3-1: Chủ cửa hàng chỉ có quyền tạo phiếu ưu đãi trong cửa hàng của mình BR4.3-2: Mã phiếu ưu đãi phải là duy nhất trong cửa hàng
Non-Functional Requirement	NFR4.3-1: Giao diện người dùng - Giao diện người dùng của quy trình tạo phiếu ưu đãi phải được thiết kế một cách trực quan và thân thiện, giúp chủ cửa hàng dễ dàng nhập thông tin và tạo phiếu ưu đãi NFR4.3-2: Tính mở rộng - Hệ thống phải có khả năng mở rộng để hỗ trợ việc tạo hàng ngàn phiếu ưu đãi trong cửa hàng mà không gây ảnh hưởng đến hiệu suất hoặc sự ổn định của hệ thống

Table 3.28: Use case Tao phiếu ưu đãi



3.2.2.29 Xoá phiếu ưu đãi

Use Case ID	UC-3.3.2
Use Case Name	Delete coupons
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng xóa một phiếu ưu đãi khỏi cửa hàng
Priority	Cao
Trigger	Chủ cửa hàng muốn xóa một phiếu ưu đãi khỏi cửa hàng
Pre-Conditions	Chủ cửa hàng đã đăng nhập vào hệ thống.
Post-Conditions	phiếu ưu đãi được xóa khỏi cửa hàng và không còn hiển thị trong danh sách phiếu ưu đãi
Basic Flow	<ol style="list-style-type: none">Chủ cửa hàng truy cập vào trang quản lý phiếu ưu đãi trong hệ thốngHệ thống hiển thị danh sách các phiếu ưu đãi hiện có trong cửa hàngChủ cửa hàng nhấn chọn icon thùng rác trên phiếu ưu đãi cần xóaHệ thống hiển thị thông tin về phiếu ưu đãi và yêu cầu xác nhận xóaChủ cửa hàng xác nhận xóa phiếu ưu đãiHệ thống xóa phiếu ưu đãi khỏi cửa hàngHệ thống thông báo cho chủ cửa hàng biết về việc xóa phiếu ưu đãi thành công
Alternative Flow	Alternative 4.3.1-1: tại bước 5 5a. Nếu chủ cửa hàng không xác nhận xóa phiếu ưu đãi, quá trình xóa phiếu ưu đãi kết thúc và phiếu ưu đãi không bị xóa khỏi cửa hàng
Exception Flow	Exception 4.3.1-1: Nếu tài khoản của chủ cửa hàng bị vô hiệu hóa và không có quyền truy cập hoặc thực hiện thao tác xóa phiếu ưu đãi, hệ thống sẽ hiển thị thông báo lỗi tương ứng và quá trình xóa phiếu ưu đãi sẽ kết thúc
Business Rules	BR4.3.1-1: Chủ cửa hàng chỉ có quyền xóa phiếu ưu đãi trong cửa hàng của mình
Non-Functional Requirement	NFR4.3.1-1: Giao diện người dùng - Giao diện người dùng của quy trình xóa phiếu ưu đãi phải được thiết kế một cách trực quan và thân thiện, giúp chủ cửa hàng dễ dàng xóa phiếu ưu đãi

Table 3.29: Use case Xóa phiếu ưu đãi



3.2.2.30 Cập nhật phiếu ưu đãi

Use Case ID	UC-3.3.3
Use Case Name	Update coupons
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng cập nhật thông tin của một phiếu ưu đãi trong cửa hàng
Priority	Cao
Trigger	Chủ cửa hàng muốn cập nhật thông tin của một phiếu ưu đãi trong cửa hàng
Pre-Conditions	Chủ cửa hàng đã đăng nhập vào hệ thống
Post-Conditions	Thông tin của phiếu ưu đãi được cập nhật thành công và hiển thị đúng trong danh sách phiếu ưu đãi
Basic Flow	<ol style="list-style-type: none">Chủ cửa hàng truy cập vào trang quản lý phiếu ưu đãi trong hệ thốngHệ thống hiển thị danh sách các phiếu ưu đãi hiện có trong cửa hàngChủ cửa hàng chọn phiếu ưu đãi cần cập nhật thông tinHệ thống hiển thị thông tin chi tiết về phiếu ưu đãi và cho phép chỉnh sửa các thông tinChủ cửa hàng cập nhật thông tin phiếu ưu đãiChủ cửa hàng xác nhận hoàn tất cập nhật thông tin phiếu ưu đãiHệ thống kiểm tra tính hợp lệ của thông tin và cập nhật thông tin phiếu ưu đãi trong cửa hàngHệ thống thông báo cho chủ cửa hàng biết về việc cập nhật thông tin phiếu ưu đãi thành công
Alternative Flow	<p>Alternative 4.2.2-1: tại bước 6 6a.Nếu chủ cửa hàng không xác nhận hoàn tất cập nhật thông tin phiếu ưu đãi, quá trình cập nhật thông tin kết thúc và thông tin phiếu ưu đãi không thay đổi</p> <p>Alternative 4.2.2-2: tại bước 7 7a. Nếu kiểm tra thấy thông tin cập nhật không hợp lệ, hệ thống sẽ hủy cập nhật và thông báo cho chủ cửa hàng</p>
Exception Flow	Exception 4.2.2-1: Nếu tài khoản của chủ cửa hàng bị vô hiệu hóa và không có quyền truy cập hoặc thực hiện thao tác cập nhật phiếu ưu đãi, hệ thống sẽ hiển thị thông báo lỗi tương ứng và quá trình cập nhật phiếu ưu đãi sẽ kết thúc
Business Rules	BR4.2.2-1: Chủ cửa hàng chỉ có quyền cập nhật thông tin phiếu ưu đãi trong cửa hàng của mình



Non-Functional Requirement	NFR4.2.2-1: Giao diện người dùng - Giao diện người dùng của quy trình cập nhật phiếu ưu đãi phải được thiết kế một cách trực quan và thân thiện, giúp chủ cửa hàng dễ dàng nhập thông tin và cập nhật phiếu ưu đãi
-----------------------------------	---

Table 3.30: Use case Cập nhật phiếu ưu đãi

3.2.2.31 Nhận tiền thanh toán từ các đơn hàng

Use Case ID	UC-3.4
Use Case Name	Receive payments of orders
Actor(s)	Chủ cửa hàng
Description	Chủ cửa hàng nhận tiền thanh toán từ các đơn hàng của khách hàng
Priority	Cao
Trigger	Khách hàng hoàn tất thanh toán cho đơn hàng
Pre-Conditions	<ul style="list-style-type: none">– Chủ cửa hàng đã đăng nhập vào hệ thống– Đơn hàng đã được khách hàng đặt và thanh toán thành công
Post-Conditions	Tiền thanh toán được nhận thành công và đơn hàng được cập nhật trạng thái thanh toán trong hệ thống
Basic Flow	<ol style="list-style-type: none">1. Hệ thống ghi nhận thanh toán thành công từ khách hàng cho một đơn hàng2. Hệ thống tự động chuyển tiền thanh toán nhận được vào tài khoản của chủ cửa hàng3. Hệ thống cập nhật đơn hàng với trạng thái thanh toán thành công4. Hệ thống thông báo cho chủ cửa hàng biết về việc nhận tiền thanh toán thành công
Alternative Flow	Không
Exception Flow	Không
Business Rules	BR4.4-1: Tiền thanh toán chỉ được chuyển vào tài khoản của chủ cửa hàng sau khi khách hàng hoàn tất thanh toán cho đơn hàng
Non-Functional Requirement	NFR4.4-1: Bảo mật - Hệ thống phải đảm bảo tính bảo mật và an toàn trong việc chuyển tiền vào tài khoản của chủ cửa hàng

Table 3.31: Use case Nhận tiền thanh toán từ các đơn hàng

Chapter 4

Thiết kế hệ thống

4.1 Mô hình hóa cơ sở dữ liệu

4.1.1 Mô hình hóa cơ sở dữ liệu mức khái niệm

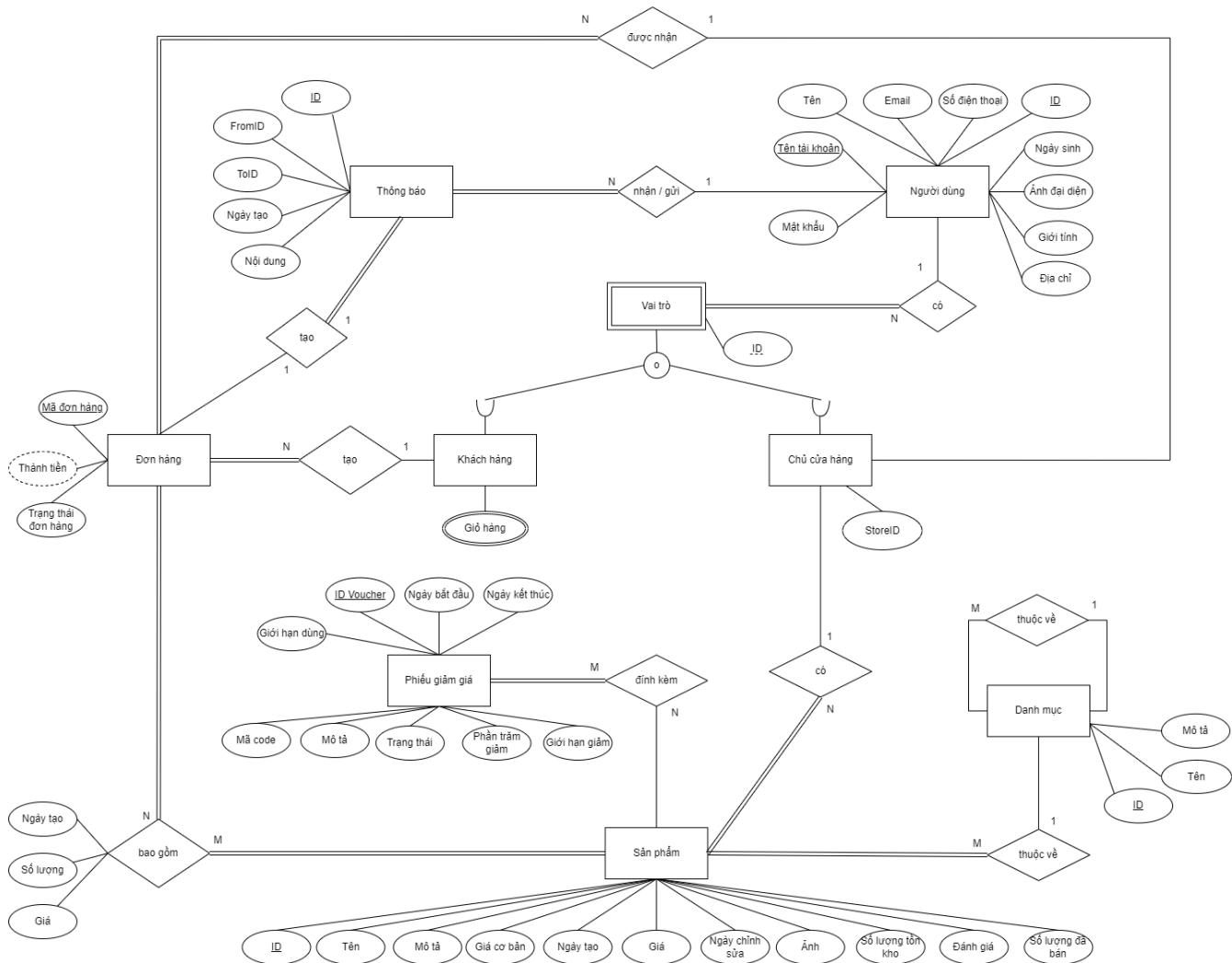


Figure 4.1: Lược đồ thực thể mối liên kết

4.1.2 Mô hình hóa cơ sở dữ liệu mức luận lý

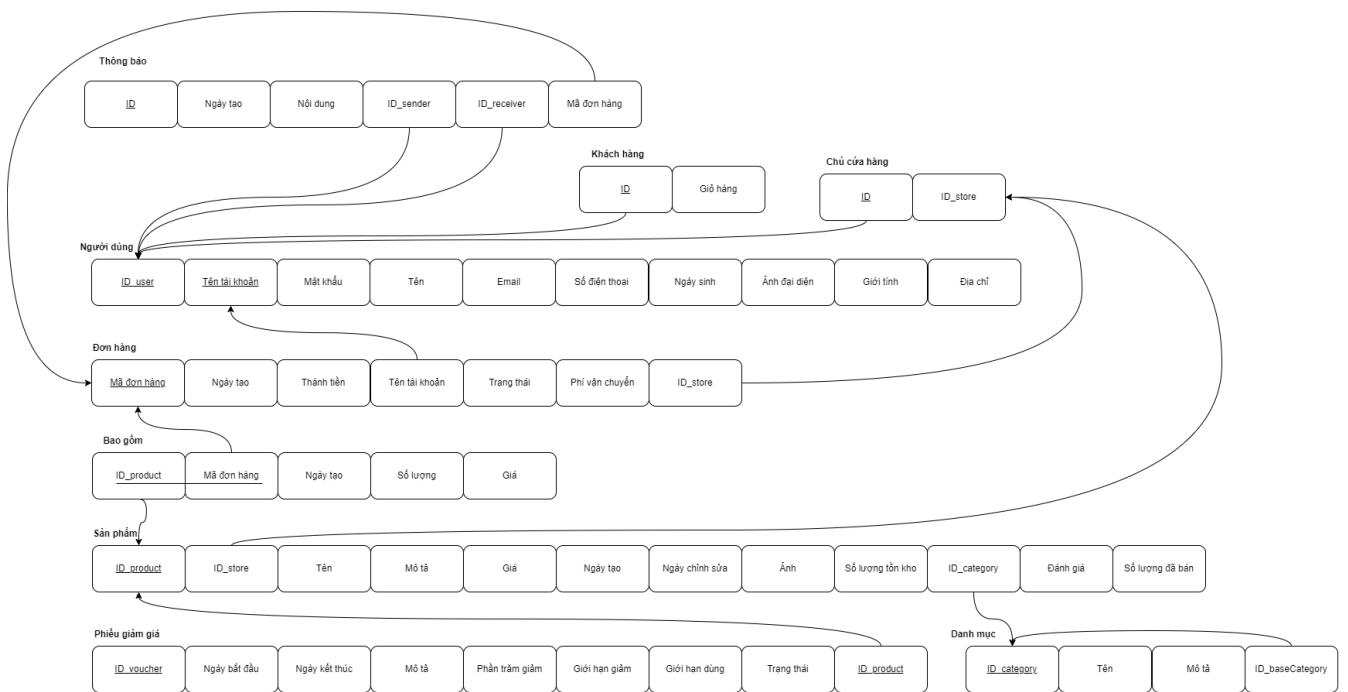


Figure 4.2: Mô hình dữ liệu quan hệ

4.2 Kiến trúc hệ thống

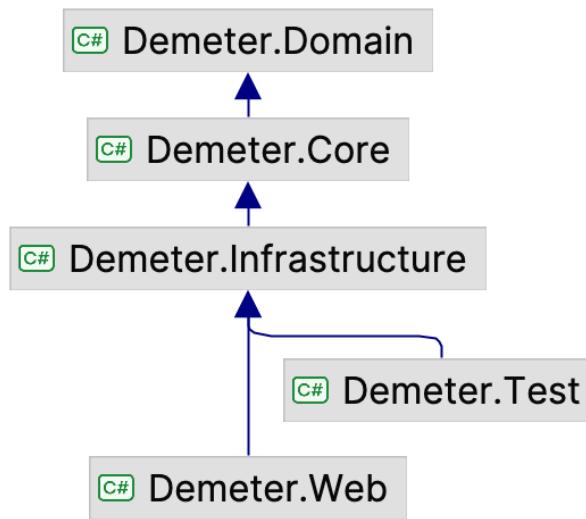


Figure 4.3: Kiến trúc hệ thống



4.3 API Endpoints

4.3.1 App Settings

/api/settings - GET

Operation	GET
Description	Retrieve application settings.
Responses	200: Success

/api/settings - POST

Operation	POST
Description	Create new application setting.
Request Body	AppSettings schema (JSON)
Responses	200: Success

/api/settings - DELETE

Operation	DELETE
Description	Delete a specific setting based on ID.
Parameters	id: string (required)
Responses	200: Success

/api/settings/update - POST

Operation	POST
Description	Update multiple settings in a single request.
Request Body	Array of AppSettings schema (JSON)
Responses	200: Success

4.3.2 Authentication

/api/auth/signup - POST

Operation	POST
Description	Register a new account.
Request Body	Account schema (JSON)
Responses	200: Success

/api/auth/login - POST

Operation	POST
Description	Authenticate a user and return a token.
Request Body	LoginInfo schema (JSON)
Responses	200: Success



/api/auth/logout - POST

Operation	POST
Description	Log out a user and invalidate the session token.
Responses	200: Success

4.3.3 Orders

/api/orders - GET

Operation	GET
Description	Retrieve all orders.
Responses	200: Success

/api/orders - POST

Operation	POST
Description	Create a new order.
Request Body	Orders schema (JSON)
Responses	200: Success

/api/orders - DELETE

Operation	DELETE
Description	Delete a specific order based on ID.
Parameters	id: string (required)
Responses	200: Success

/api/orders/voucher - GET

Operation	GET
Description	Retrieve all vouchers associated with orders.
Responses	200: Success

/api/orders/voucher - DELETE

Operation	DELETE
Description	Delete a voucher based on the given ID.
Parameters	id: string (required)
Responses	200: Success

/api/orders/items - GET

Operation	GET
Description	Retrieve all items in all orders.
Responses	200: Success



/api/orders/items - DELETE

Operation	DELETE
Description	Delete an item from an order based on the given ID.
Parameters	id: string (required)
Responses	200: Success

/api/orders/update - POST

Operation	POST
Description	Update multiple orders in a single request.
Request Body	Array of Orders schema (JSON)
Responses	200: Success

4.3.4 Products

/api/products - GET

Operation	GET
Description	Retrieve all products with details.
Responses	200: Success, returns an array of products.

/api/products - POST

Operation	POST
Description	Create a new product entry with detailed specifications.
Request Body	Products schema (JSON)
Responses	200: Success, product created.

/api/products - DELETE

Operation	DELETE
Description	Delete a specific product based on product ID.
Parameters	id: string (required)
Responses	200: Success, product deleted.

/api/products/search - GET

Operation	GET
Description	Search for products by name.
Parameters	name: string (required)
Responses	200: Success, returns matching products.

/api/products/id - GET

Operation	GET
Description	Retrieve detailed information about a product by ID.
Parameters	id: string (required)
Responses	200: Success, returns product details.



/api/products/categories - GET

Operation	GET
Description	Retrieve all product categories.
Responses	200: Success, returns categories array.

/api/products/categories - DELETE

Operation	DELETE
Description	Delete a specific category based on category ID.
Parameters	id: string (required)
Responses	200: Success, category deleted.

/api/products/prices - GET

Operation	GET
Description	Retrieve pricing information for all products.
Responses	200: Success, returns prices array.

/api/products/prices - DELETE

Operation	DELETE
Description	Delete pricing information based on price ID.
Parameters	id: string (required)
Responses	200: Success, price information deleted.

/api/products/update - POST

Operation	POST
Description	Update multiple product entries in a single request.
Request Body	Array of Products schema (JSON)
Responses	200: Success, products updated.

4.3.5 Stripe Integration

/api/stripe/customer/add - POST

Operation	POST
Description	Adds a new customer to Stripe with credit card details.
Request Body	Includes Email, Name, and Credit Card details (Name, Number, Expiration, CVC).
Responses	200: Success, returns the created Stripe customer object.



/api/stripe/payment/add - POST

Operation	POST
Description	Processes a payment via Stripe for a customer.
Request Body	AddStripePayment schema, includes customer ID, receipt email, and payment details.
Responses	200: Success, returns the Stripe payment object.

/api/stripe/create-checkout-session - POST

Operation	POST
Description	Creates a Stripe checkout session for handling payments on the client side.
Request Body	CreateCheckoutSessionStripeRequest schema, includes product and pricing details.
Responses	200: Success, returns session details for client-side handling.

4.3.6 Users

/api/users - GET

Operation	GET
Description	Retrieve a list of all users.
Responses	200: Success, returns an array of user objects.

/api/users/update - POST

Operation	POST
Description	Update multiple user profiles in a single request.
Request Body	Array of Users schema (JSON), allowing batch updates.
Responses	200: Success, users updated.

/api/users/account - GET

Operation	GET
Description	Retrieve account details for the logged-in user.
Responses	200: Success, returns user account details.

/api/users/account - DELETE

Operation	DELETE
Description	Delete a user account based on the given ID.
Parameters	id: string, UUID (required)
Responses	200: Success, account deleted.



4.4 Hiện thực

4.4.1 Đăng nhập - Đăng ký

Trang đăng nhập được thiết kế để người dùng có thể đăng nhập vào tài khoản của họ. Giao diện bao gồm một form đăng nhập với hai trường nhập liệu cho tên người dùng và mật khẩu, cùng với nút "Đăng nhập". Trang cũng cung cấp liên kết "Quên mật khẩu" để người dùng khôi phục mật khẩu nếu cần, và liên kết "Đăng ký" để chuyển hướng người dùng đến trang đăng ký.

Đăng Nhập

Nhập email của bạn dưới đây để đăng nhập vào tài khoản của bạn

Email *

Mật khẩu *

eye

[Quên mật khẩu?](#)

Đăng Nhập

[Đăng nhập với Google](#)

Bạn chưa có tài khoản? [Đăng ký](#)

Figure 4.4: Trang đăng nhập

Trang đăng ký cho phép người dùng tạo một tài khoản mới trên hệ thống. Trang có cấu trúc tương tự với trang đăng nhập, bao gồm form đăng ký với các trường nhập liệu cho tên người dùng, địa chỉ email, mật khẩu và xác nhận mật khẩu, cùng với nút "Đăng ký". Ngoài ra, trang cũng cung cấp liên kết "Đăng nhập" để người dùng có thể chuyển hướng đến trang đăng nhập.



Đăng Ký

Nhập thông tin của bạn dưới đây để tạo tài khoản mới

Email *

Mật khẩu *

Xác nhận mật khẩu *

Đăng Ký

Đăng ký với Google

Bạn đã có tài khoản? [Đăng nhập](#)

Figure 4.5: Trang đăng ký

4.4.2 Trang chủ

Trang chủ hiển thị thông tin tổng quan về các loại danh mục sản phẩm và danh sách các sản phẩm nông sản.

RoadSide

Danh mục	Ảnh	Tên	Đơn vị	Giá	Đutton
Sữa các loại		Sữa	Unit	36.000	Mua
Rau, củ, nấm, trái cây		Táo	Táo đỏ tươi sạch	55.000	Mua
Thịt, cá, trứng, hải sản		Chuối	Chuối chín cây	10.000	Mua
Gạo, bột, đồ khô		Sữa	Sữa tươi tiệt trùng	29.000	Mua
Dầu ăn, nước chấm, gia vị		Khoai tây	Khoai tây tươi từ trang trại	29.000	Mua
Trà, cà phê, mật ong		Dưa lưới	Dưa lưới tròn ruột cam	50.800	Mua
Thực phẩm sấy các loại		Bơ	Bơ	25.000	Mua
		Ói	Ói ruột đỏ	22.900	Mua
		Mận	Mận	37.200	Mua
		Cam sành	Cam sành	54.000	Mua
		Nho xanh	Nho xanh	54.000	Mua
		Xoài cát Hòa Lộc	Xoài cát Hòa Lộc	30.400	Mua
		Mít	Mít	30.800	Mua
		Chanh dây	Chanh dây	30.800	Mua
		Cóc	Cóc	32.400	Mua
		Sapoche	Hồng xiêm	33.200	Mua

Figure 4.6: Trang chủ



Phía bên trái là danh mục các loại sản phẩm được phân loại theo từng đặc trưng riêng.

The screenshot shows a mobile application interface for 'RoadSide'. On the left, there is a sidebar with the 'RoadSide' logo at the top. Below it, a list of categories is shown with arrows indicating they can be expanded:

- Sữa các loại
- Rau, củ, nấm, trái cây
- Thịt, cá, trứng, hải sản
- Gạo, bột, đồ khô
- Dầu ăn, nước chấm, gia vị
- Trà, cà phê, mật ong
- Thực phẩm sấy các loại

On the right, a main content area also features the 'RoadSide' logo. It displays a list of sub-categories under the selected category 'Thịt, cá, trứng, hải sản', which is highlighted with a green border:

- Thịt bò
- Cá, hải sản, khô
- Thịt gà, vịt, chim
- Thịt heo
- Trứng gà, vịt, cút
- Gạo, bột, đồ khô
- Dầu ăn, nước chấm, gia vị
- Trà, cà phê, mật ong
- Thực phẩm sấy các loại

Figure 4.7: Danh mục sản phẩm

Sau khi chọn loại sản phẩm cần mua, danh sách các sản phẩm đã được phân loại sẽ hiện ra ở bên phải.

This screenshot shows the application after a category has been selected. The sidebar now highlights the 'Nấm các loại' category. The main content area displays a grid of mushroom products with their names, unit prices, and a 'Buy' button:

Loại Nấm	Tên Sản Phẩm	Đơn Vị	Giá	Đutton
Nấm rơm	Nấm rơm	Unit	120.000	Mua
Nấm đùi gà	Nấm đùi gà	Unit	165.000	Mua
Nấm bào ngư trắng	Nấm bào ngư trắng	Unit	65.000	Mua
Nấm kim châm	Nấm kim châm	Unit	80.000	Mua
Nấm linh chi nâu	Nấm linh chi nâu	Unit	100.000	Mua
Nấm bào ngư xám	Nấm bào ngư xám	Unit	80.000	Mua
Nấm hương	Nấm hương	Unit	210.000	Mua
Nấm linh chi trắng	Nấm linh chi trắng	Unit	200.000	Mua

Figure 4.8: Phân loại sản phẩm theo danh mục

4.4.3 Thông tin sản phẩm

Ở trang chủ mỗi sản phẩm sẽ được hiển thị dưới dạng một thẻ sản phẩm, bao gồm các thông tin cơ bản như tên, giá, hình ảnh mô tả và giá giảm (nếu có).



Figure 4.9: Thẻ sản phẩm

Nếu bấm vào thẻ sản phẩm, sẽ được chuyển hướng đến trang xem thông tin chi tiết của sản phẩm đó.

This screenshot shows the detailed product page for Red Apples. At the top left is the "RoadSide" logo. On the right are icons for a magnifying glass, a user profile, and a cart. The main image is a large red apple and a slice. To the right, the product name "Táo" is displayed above a "JOHN DOE" badge. Below the name is a rating of "5★ | Đã bán 1 khuyến mãi". The original price "41.250đ" is shown in red, with "55.000đ" crossed out below it. The text "Táo đỏ tươi sạch" follows. A dropdown menu for "Chọn mẫu mã" is open, showing "Cam" as the selected option. At the bottom are two buttons: a blue "Thêm vào giỏ" (Add to cart) and a green "Mua ngay" (Buy now).

Figure 4.10: Trang thông tin sản phẩm chi tiết

4.4.4 Đặt hàng

Sau khi đăng nhập vào hệ thống, nếu nhấn vào biểu tượng giỏ hàng trên thanh navbar, trang giỏ hàng sẽ hiện ra. Khách hàng có thể xem thông tin về các sản phẩm mà người dùng đã thêm vào giỏ hàng, bao gồm thông tin về tên, hình ảnh, giá và số lượng. Ở đây người dùng có thể xem lại



và điều chỉnh các mục trong giỏ hàng trước khi đặt hàng. Người dùng có thể thay đổi số lượng hoặc xóa sản phẩm khỏi giỏ hàng.

Sản Phẩm	Đơn Giá	Số Lượng	Số Tiền	Xóa	Tim sản phẩm tương tự
Táo Miễn phí 15 ngày	đ55,000	1	đ55,000	Xóa	Tim sản phẩm tương tự
Chuối Miễn phí 15 ngày	đ10,000	2	đ20,000	Xóa	Tim sản phẩm tương tự
Bơ Miễn phí 15 ngày	đ25,000	2	đ50,000	Xóa	Tim sản phẩm tương tự
Ổi Miễn phí 15 ngày	đ22,900	2	đ45,800	Xóa	Tim sản phẩm tương tự

Chọn Tất Cả (4)

Đưa vào mục Đã thích

Tổng thanh toán (7 Sản phẩm): 170,800đ

Mua Hàng

Figure 4.11: Giỏ hàng

Bấm nút "Đặt hàng" để tiến hành tạo đơn hàng, một modal được hiện lên để xác nhận lại thông tin đặt hàng và phương thức thanh toán.

Địa Chỉ Nhận Hàng
Trần Hào Tuấn Kiệt (+84) 355749742

Tầng 5, Tòa Nhà Pijico, Số 186, Điện Biên Phủ, Phường Võ Thị Sáu, Quận 3, TP. Hồ Chí Minh

Thay đổi

Sản phẩm

Táo Loại: Táo	đ55,000	1	đ55,000
Chuối Loại: Chuối	đ10,000	2	đ20,000
Bơ Loại: Bơ	đ25,000	2	đ50,000
Ổi Loại: Ổi	đ22,900	2	đ45,800

GIẢM 61.000 - VOUCHER CỦA SHOP Chọn Voucher Khác

Vận chuyển

Vận Chuyển Nhanh - đ0
Đơn vị nhận hàng từ 26 Tháng 7 - 31 Tháng 7

Lời nhắn:
Lưu ý cho Người bán...

Tổng số tiền (7 sản phẩm): đ170,800

Thanh toán

Figure 4.12: Form xác nhận đơn hàng

Sau khi tiến hành đặt hàng và thanh toán thành công sẽ hiện thông báo đặt hàng thành công.

4.4.5 Chỉnh sửa thông tin cá nhân

Sau khi đăng nhập, người dùng có thể nhấp vào biểu tượng hình người trên thanh navbar để truy cập trang thông tin người dùng. Đây là nơi mà người dùng có thể quản lý và cập nhật thông tin



cá nhân của họ. Trang này cung cấp cho người dùng một giao diện để xem và chỉnh sửa thông tin cá nhân, giúp tạo ra trải nghiệm cá nhân hóa cho khách hàng.

The screenshot shows the 'Thông tin cá nhân' (Personal Information) section. It includes fields for Name (Họ & Tên), Date of Birth (Ngày sinh), Gender (Giới tính), Phone Number (Số điện thoại), and Email (Email). There are also sections for 'Bảo mật' (Security) with options to change password (Đổi mật khẩu) or generate a PIN (Thiết lập mã PIN), and 'Liên kết mạng xã hội' (Social Media Links) for Facebook and Google. A blue 'Lưu thay đổi' (Save changes) button is at the bottom.

Figure 4.13: Trang thông tin người dùng

Trang thông tin người dùng cũng cung cấp khả năng thay đổi mật khẩu. Điều này cho phép người dùng cập nhật mật khẩu của mình một cách an toàn và bảo mật.

Người dùng cũng có thể thiết lập nhiều địa chỉ cho tài khoản của mình.

The screenshot shows the 'Số địa chỉ' (Address) section. It lists two addresses: 'TUẤN KIẾT TRẦN' and 'TRẦN HÀ TUẤN KIẾT'. Each address has a 'Chỉnh sửa' (Edit) button. A green 'XÓA' (Delete) button is visible between the two address entries. A blue '+ Thêm địa chỉ mới' (Add new address) button is at the bottom.

Figure 4.14: Địa chỉ người dùng

4.4.6 Quản lý đơn đặt hàng

Ở trang profile cũng cung cấp thông tin đơn hàng của người dùng, bao gồm các thông tin về đơn hàng đang trong quá trình vận chuyển hoặc các đơn đã hoàn thành trước đây như số đơn hàng, ngày đặt hàng, sản phẩm đã mua và trạng thái đơn hàng. Điều này giúp người dùng theo dõi và quản lý các đơn hàng của họ một cách thuận tiện.



RoadSide

Trang chủ / Đơn mua

Tất cả

Tất cả Chờ xác nhận Chờ lấy hàng Đang giao Đã giao Đơn Hủy Trả hàng/Hoàn tiền

Bạn có thể tìm kiếm theo tên Shop, ID đơn hàng hoặc Tên Sản phẩm

Đơn hàng 1					CHỜ XÁC NHẬN
Sản Phẩm	Đơn Giá	Số Lượng	Số Tiền		
Táo	55,000	1	55,000		
Chuối	10,000	2	20,000		
Bơ	25,000	2	50,000		
Ói	22,900	2	45,800		
Thành tiền:	170,800				

Đơn hàng 2					CHỜ XÁC NHẬN
Sản Phẩm	Đơn Giá	Số Lượng	Số Tiền		
Táo	55,000	1	55,000		

Figure 4.15: Trang thông tin người dùng - Quản lý đơn hàng

4.4.7 Kênh cho chủ cửa hàng

Mỗi tài khoản đều có thể tạo cửa hàng riêng cho mình, sẽ có một số trang riêng cho người dùng để quản lý cửa hàng và sản phẩm.

- Quản lý sản phẩm:

- Tính năng quản lý sản phẩm cho phép chủ cửa hàng xem và quản lý các sản phẩm hiện có trong cửa hàng. Mỗi sản phẩm sẽ được hiển thị với các thông tin cơ bản như mã sản phẩm, tên, loại, giá, số lượng tồn kho và trạng thái hiện tại của sản phẩm. Ngoài ra, chủ cửa hàng cũng có thể xóa hay chỉnh sửa thông tin sản phẩm ở trang này.



Tên sản phẩm	Đã bán	Giá	Kho hàng	Mô tả	Thao tác
Táo	10	55000	100	Táo đỏ tươi sạch	Xem chi tiết
Chuối	5	10000	100	Chuối chín cát	Xem chi tiết
Sữa	10	36000	100	Sữa tươi tiệt trùng	Xem chi tiết
Khoai tây	2	29000	100	Khoai tây tươi từ trang trại	Xem chi tiết
Dưa lưới	10	50800	100	Dưa lưới tròn ruột cam	Xem chi tiết
Bơ	2	25000	100	Bơ	Xem chi tiết
Ói	4	22900	100	Ói ruột đỏ	Xem chi tiết
Mận	5	37200	100	Mận	Xem chi tiết

Figure 4.16: Quản lý sản phẩm

- Chủ cửa hàng có thể lọc và tìm kiếm sản phẩm theo tên cũng như các đặc điểm khác.
- Thông tin của sản phẩm cũng có thể được chỉnh sửa ở đây.
- Thêm sản phẩm mới: Chủ cửa hàng có thể thêm sản phẩm mới cho cửa hàng của mình thông qua tính năng này. Trong trang thêm sản phẩm, chủ cửa hàng sẽ phải điền một form với các trường thông tin chi tiết về sản phẩm như tên, hình ảnh, loại, mô tả,...

Thông tin cơ bản

Hình ảnh sản phẩm

+

Thêm hình ảnh

Kéo thả tệp tại đây để tải lên. Lưu ý chỉ chấp nhận tệp có đuôi .jpeg kích thước dưới 30MB.

Chọn tệp

Tên sản phẩm	Nhập tên sản phẩm
Chọn danh mục	Chọn danh mục
Mô tả sản phẩm	Nhập mô tả của sản phẩm

Thông tin bán hàng

Thương hiệu	Vui lòng nhập	Xuất xứ	Vui lòng nhập
Thành phần	Vui lòng nhập	Đơn vị bán	kg / hộp / 100g
Đơn giá (đồng)	100000	Số lượng	Vui lòng nhập số nguyên

Figure 4.17: Thêm sản phẩm mới



4.4.8 Chế độ sáng - tối

Ngoài ra trang web còn hỗ trợ chuyển đổi chế độ sáng - tối cho giao diện.

The figure consists of three screenshots of the RoadSide e-commerce platform, each showing a different page in dark mode:

- Screenshot 1: Product Catalog (Giới thiệu)**

This screenshot shows a grid of product cards. Each card includes a product image, name, unit, price, and a green "Mua" (Buy) button. The products listed are Rau muống nước, Nấm rơm, Dựng heo, Rau mồng tai, Sườn cốt lết, Nạc đầm heo, Tiêu xanh, Hành tây, Cam sành, and Ớt chuông xanh-vàng-đỏ.

Product	Unit	Price	Action
Rau muống nước	Unit	20.000	Mua
Nấm rơm	Unit	120.000	Mua
Dựng heo	Unit	135.000	Mua
Rau mồng tai	Unit	25.000	Mua
Sườn cốt lết	Unit	144.000	Mua
Nạc đầm heo	Unit	200.000	Mua
Tiêu xanh	Unit	9.000	Mua
Hành tây	Unit	35.000	Mua
Cam sành	Unit	54.000	Mua
Ớt chuông xanh-vàng-đỏ	Unit	60.000	Mua
- Screenshot 2: Shopping Cart (Giỏ hàng)**

This screenshot shows the shopping cart interface. It lists items with checkboxes, product images, names, descriptions, unit prices, quantities, total amounts, and buttons for removing items or finding similar products.

Sản Phẩm	Đơn Giá	Số Lượng	Số Tiền	Actions
Chuối Miễn phí 15 ngày	đ10.000 đ7,500	2	đ20,000	Xóa Tim sản phẩm tương tự
Sapoché Miễn phí 15 ngày	đ33,200	2	đ66,400	Xóa Tim sản phẩm tương tự
Rau muống nước Miễn phí 15 ngày	đ20,000	1	đ20,000	Xóa Tim sản phẩm tương tự

Total amount: 86,400đ | Mua Hàng (Buy Now)
- Screenshot 3: User Profile (Tài khoản của tôi)**

This screenshot shows the user profile page. It includes sections for personal information (Họ & Tên: Nguyễn Văn A, Ngày sinh: 19/12/2002, Giới tính: Nam), security (Đổi mật khẩu, Thiết lập mã PIN, Yêu cầu xóa tài khoản), social links (Facebook, Google), and a "Lưu thay đổi" (Save changes) button.

Figure 4.18: Một số trang ở chế độ tối

Chapter 5

Tổng kết và hướng phát triển đề tài

5.1 Tổng kết

Trong quá trình nghiên cứu và triển khai dự án, nhóm chúng tôi đã đạt được các thành tựu quan trọng sau đây:

- Nghiên cứu kỹ lưỡng và phân tích các sản phẩm tương tự trên thị trường để xác định các tính năng cần thiết cho hệ thống.
- Phân tích chi tiết các nghiệp vụ liên quan đến các đối tượng người dùng như chủ cửa hàng và khách hàng.
- Hoàn thành mô hình hóa cơ sở dữ liệu ở cấp độ khái niệm và luận lý.
- Thiết kế và mô tả chi tiết các usecase trong hệ thống thông qua sơ đồ usecase.
- Lựa chọn và áp dụng công nghệ phù hợp cho dự án:
 - Sử dụng ASP.NET cho Back-end và Entity Framework cho tương tác cơ sở dữ liệu.
 - Triển khai cơ sở dữ liệu trên PostgreSQL Database.
 - Áp dụng thành công kiến trúc hệ thống theo hướng Clean Architecture để đảm bảo tính sạch sẽ, rõ ràng và dễ bảo trì của mã nguồn.
 - Áp dụng các design pattern để thiết kế module, giúp các module trong hệ thống liên kết chặt chẽ và dễ dàng mở rộng.
 - Kết nối chặt chẽ giữa hệ thống và cơ sở dữ liệu.
 - Sử dụng Next.js cùng TypeScript và Mantine để phát triển giao diện người dùng.
- Thiết kế kiến trúc mạng kết nối giữa người dùng và hệ thống, đảm bảo tính bảo mật và hiệu quả truyền tải dữ liệu.
- Triển khai hệ thống thành công trên Vercel, tăng cường khả năng truy cập và ổn định của hệ thống.

Chi tiết về quá trình hiện thực và các giao diện người dùng được trình bày cụ thể ở chương 4.4: [Hiện thực](#).



5.2 Hướng phát triển

Trong tương lai, chúng tôi kỳ vọng hệ thống này không chỉ được duy trì mà còn mở rộng các tính năng hiện tại, hướng tới các mục tiêu sau:

- Tối ưu hóa trải nghiệm người dùng bằng cách thiết kế giao diện thân thiện, đẹp mắt, và dễ sử dụng.
- Mở rộng phương thức thanh toán, tích hợp thêm các giải pháp thanh toán mới để tăng cường sự tiện lợi và an toàn cho người tiêu dùng và nhà cung cấp.
- Cải thiện an ninh mạng, áp dụng các biện pháp bảo mật mạnh mẽ để đảm bảo an toàn cho dữ liệu và thông tin cá nhân của người dùng.
- Phát triển chức năng đánh giá và nhận xét sản phẩm, qua đó xây dựng một cộng đồng mua sắm minh bạch và tin cậy.
- Khám phá và áp dụng Trí Tuệ Nhân Tạo (AI) và Học Máy (Machine Learning) để cải tiến trải nghiệm mua sắm, cụ thể là:
 - Phân loại và lọc sản phẩm tự động, giúp người dùng dễ dàng tìm kiếm và lựa chọn.
 - Phát triển các mô hình so sánh sản phẩm, đề xuất các lựa chọn thay thế phù hợp dựa trên giá, đánh giá và tính năng.
 - Triển khai các mô hình gợi ý sản phẩm dựa trên lịch sử mua hàng và hành vi duyệt web của người dùng để cung cấp gợi ý cá nhân hóa.
- Tích hợp tính năng gợi ý thông minh vào giao diện người dùng, giúp hiển thị các sản phẩm được đề xuất tùy theo hành vi tìm kiếm của từng người dùng.

Những hướng phát triển này không chỉ nhằm mục đích nâng cao chất lượng trải nghiệm người dùng mà còn đảm bảo hệ thống "Sàn Thương Mại Điện Tử Nông Sản" không chỉ thích ứng linh hoạt với các biến động của thị trường mà còn vượt qua kỳ vọng của người dùng, từng bước hoàn thiện và phát triển.

Bibliography

- [1] The Clean Architecture, Robert C. Martin (Uncle Bob),
<https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>
- [2] Design Patterns, ©2014-2023 Refactoring.Guru,
<https://refactoring.guru/design-patterns>
- [3] Builder, © 2014-2023 Refactoring.Guru,
<https://refactoring.guru/design-patterns/builder>
- [4] Dependency injection in ASP.NET Core, © Microsoft 2023,
<https://learn.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-8.0>
- [5] Pickbazar web,
<https://pickbazar-react.vercel.app/>
- [6] Next.js documentation,
<https://nextjs.org/docs>
- [7] Using Typescript, ©2023 Meta Open Source,
<https://react.dev/learn/typescript>
- [8] Getting Started, Mantine,
<https://mantine.dev/getting-started/>
- [9] Get started with Vercel,
<https://vercel.com/docs/getting-started-with-vercel/>
- [10] Tutorial: Get started with EF Core in an ASP.NET MVC web app, © Microsoft 2023,
<https://learn.microsoft.com/en-us/aspnet/core/data/ef-mvc/intro?view=aspnetcore-7.0>
- [11] Entity Framework Core – Database first, Jenx.si ©2023,
<https://www.jenx.si/2020/04/24/entity-framework-core-database-first/>
- [12] PostgreSQL documentation,
<https://www.postgresql.org/docs/current/>