

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HÙNG YÊN



BÀI TẬP LỚN
ỨNG DỤNG CỦA HÀNG BÁN HẢI SẢN

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM
HƯỚNG CHUYÊN SÂU: CÔNG NGHỆ WEB

SINH VIÊN: PHẠM THỊ ÁNH NGỌC
MÃ SINH VIÊN: 10121885
MÃ LỚP: 125212
HƯỚNG DẪN: NGUYỄN HỮU ĐÔNG

HÙNG YÊN – 2023

NHẬN XÉT

Nhận xét của giảng viên hướng dẫn:

[illegible]

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan bài tập lớn “Ứng dụng cửa hàng bán hải sản” là kết quả thực hiện của bản thân em dưới sự hướng dẫn của thầy Nguyễn Hữu Đông.

Những phần sử dụng tài liệu tham khảo trong đề án đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong đề án và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

Hưng Yên, ngày ... tháng ... năm.....

Sinh viên

Ngọc

Phạm Thị Ánh Ngọc

LỜI CẢM ƠN

Để có thể hoàn thành bài tập lớn này, lời đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện đồ án môn học này.

Đặc biệt em xin chân thành cảm ơn thầy Nguyễn Hữu Đông đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện bài tập lớn vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được bài tập lớn này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy giáo, Cô giáo về những kết quả triển khai trong bài tập lớn.

Em xin trân trọng cảm ơn!

MỤC LỤC

NHẬN XÉT	2
DANH MỤC CÁC THUẬT NGỮ	7
DANH MỤC CÁC BẢNG	8
DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ	10
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	12
1.1. Lý do chọn đề tài	12
1.2. Mục tiêu của đề tài	12
1.2.1 Mục tiêu tổng quát	12
1.2.2 Mục tiêu cụ thể	12
1.3. Giới hạn và phạm vi của đề tài	13
1.3.1 Đối tượng nghiên cứu	13
1.3.2 Phạm vi nghiên cứu	13
1.4. Nội dung thực hiện	13
1.5. Phương pháp tiếp cận	14
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	15
2.1. Quy trình phát triển phần mềm	15
2.1.1 Waterfall model – Mô hình thác nước:	17
2.1.2 V-Shaped Model – Mô hình chữ V:	18
2.1.3 Spiral Model – Mô hình xoắn ốc:	19
2.2. Thiết kế giao diện web với HTML, CSS	20
2.2.1. HTML (Hypertext Markup Language)	20
2.2.2. CSS (Cascading Style Sheet language)	23
2.3. Lập trình phía front-end	26
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	30
3.1 Phát biểu bài toán	30
3.2 Đặc tả yêu cầu phần mềm	32
3.2.1 Các yêu cầu chức năng	32
3.2.2 Biểu đồ lớp thực thể	45

3.2.3 Cơ sở dữ liệu.....	51
3.2.4 Các yêu cầu phi chức năng	55
3.3 Thiết kế giao diện	56
CHƯƠNG 4: TRIỂN KHAI WEBSITE.....	59
4.1 Triển khai các chức năng cho phân hệ người dùng	59
4.1.1 Trang chủ: Gợi ý cho bạn (Ví dụ).....	59
4.1.2 Trang Xem Chi Tiết Sản Phẩm (Ví dụ).....	62
4.2 Triển khai các chức năng cho phân hệ quản trị nội dung (nếu có).....	65
4.2.1 Đăng nhập quản trị(Ví dụ).....	65
4.2.2 Quản lý sản phẩm(Ví dụ).....	68
4.3 Kiểm thử và triển khai ứng dụng.....	74
4.3.1 Kiểm thử	74
4.3.2 Đóng gói ứng dụng	74
4.3.3 Triển khai ứng dụng.....	74
KẾT LUẬN	75
TÀI LIỆU THAM KHẢO.....	77

DANH MỤC CÁC THUẬT NGỮ

STT	Từ viết tắt	Cụm từ tiếng anh	Diễn giải
1	HTML	Hypertext Markup Language	Ngôn ngữ đánh dấu siêu văn bản
2	CSS	Cascading Style Sheet	Ngôn ngữ thiết kế
3	DAL	DataAccessLayer	
4	BUS	BusinessLayer	

DANH MỤC CÁC BẢNG

Bảng 2.1 Một số thẻ trong HTML	23
Bảng 2.2 Các bộ của CSS	24
Bảng 3.1 Dòng sự kiện chính	33
Bảng 3.2 Dòng sự kiện phụ(Kiểm tra mã trùng).....	34
Bảng 3.3 Dòng sự kiện phụ(Kiểm tra tính hợp lệ của dữ liệu)	34
Bảng 3.4 Dòng sự kiện chính	35
Bảng 3.5 Dòng sự kiện phụ(Kiểm tra mã trùng).....	36
Bảng 3.6 Dòng sự kiện phụ(Kiểm tra tính hợp lệ của dữ liệu)	37
Bảng 3.7 Dòng sự kiện chính	38
Bảng 3.8 Dòng sự kiện phụ(Kiểm tra mã trùng).....	39
Bảng 3.9 Dòng sự kiện phụ(Kiểm tra tính hợp lệ của dữ liệu)	39
Bảng 3.10 Dòng sự kiện chính	40
Bảng 3.11 Dòng sự kiện phụ(Kiểm tra mã trùng).....	41
Bảng 3.12 Dòng sự kiện phụ(Kiểm tra sự hợp lệ của dữ liệu)	41
Bảng 3.13 Dòng sự kiện chính	42
Bảng 3.14 Dòng sự kiện phụ(Kiểm tra mã trùng).....	43
Bảng 3.15 Dòng sự kiện phụ(Kiểm tra sự hợp lệ của dữ liệu)	43
Bảng 3.16 Dòng sự kiện chính phân hệ người dùng	44
Bảng 3.17 Danh sách thuộc tính bảng Quản Trị	45
Bảng 3.18 Danh sách phương thức bảng Quản Trị	46
Bảng 3.19 Danh sách thuộc tính bảng Khách hàng	46
Bảng 3.20 Danh sách phương thức bảng Khách Hàng	47
Bảng 3.21 Danh sách thuộc tính bảng Sản phẩm.....	47
Bảng 3.22 Danh sách phương thức bảng Sản phẩm	47
Bảng 3.23 Danh sách thuộc tính bảng Chi tiết sản phẩm	48
Bảng 3.24 Danh sách thuộc tính bảng Loại sản phẩm	48
Bảng 3.25 Danh sách phương thức bảng Loại sản phẩm.....	49
Bảng 3.26 Danh sách thuộc tính bảng Hóa Đơn.....	49

Bảng 3.27 Danh sách phương thức bảng Hóa Đơn.....	50
Bảng 3.28 Danh sách thuộc tính Chi tiết hóa đơn.....	50
Bảng 3.29 Bảng LoạiSP	52
Bảng 3.30 Bảng SanPham.....	52
Bảng 3.31 Bảng ChiTietSP	53
Bảng 3.32 Bảng HoaDon	53
Bảng 3.33 Bảng ChiTietHD	53
Bảng 3.34 Bảng QuanTri	54
Bảng 3.35 Bảng KháchHang.....	54

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 2.1 Mô hình thác nước	17
Hình 2.2 Mô hình chữ V	18
Hình 2.3 Mô hình xoắn ốc.....	19
Hình 2.4 Hình ảnh ví dụ HTML.....	22
Hình 3.1 Usecase quản lý khách hàng	33
Hình 3.2 Usecase quản lý sản phẩm	35
Hình 3.3 Usecase quản lý loại sản phẩm	37
Hình 3.4 Usecase quản lý đơn hàng.....	39
Hình 3.5 Usecase QLQuanTri.....	41
Hình 3.6 Usecase phân hệ người dùng.....	44
Hình 3.7 Biểu đồ lớp	51
Hình 3.8 Cơ sở dữ liệu	51
Hình 3.9 Giao diện phần người dùng gợi ý cho bạn	56
Hình 3.10 Giao diện phần người dùng chi tiết sản phẩm.....	57
Hình 3.11 Giao diện đăng nhập quản trị	57
Hình 3.12 Giao diện quản lý sản phẩm.....	58
Hình 4.1 Bố cục trang Home phần gợi ý cho bạn bằng các thẻ HTML	59
Hình 4.2 Định dạng bằng CSS cho trang Home phần gợi ý cho bạn.....	60
Hình 4.3 Javascript cho trang Home phần gợi ý cho bạn	60
Hình 4.4 Tầng DAL cho phần gợi ý cho bạn.....	61
Hình 4.5 Tầng BUS cho phần gợi ý cho bạn	61
Hình 4.6 Tầng Controller cho phần gợi ý cho bạn.....	62
Hình 4.7 Bố cục trang xem chi tiết sản phẩm bằng các thẻ HTML.....	62
Hình 4.8 Định dạng bằng CSS cho trang xem chi tiết sản phẩm.....	63
Hình 4.9 Javascript cho trang xem chi tiết sản phẩm.....	63
Hình 4.10 Tầng DAL cho trang xem chi tiết sản phẩm	64
Hình 4.11 Tầng BUS cho trang xem chi tiết sản phẩm.....	64

Hình 4.12 Tầng Controller cho trang xem chi tiết sản phẩm.....	64
Hình 4.13 Bố cục trang đăng nhập quản trị bằng các thẻ HTML	65
Hình 4.14 Định dạng bằng CSS cho trang Đăng nhập quản trị	66
Hình 4.15 Javascript cho trang Đăng nhập quản trị	67
Hình 4.16 Tầng DAL cho trang Đăng nhập quản trị	67
Hình 4.17 Tầng BUS cho trang đăng nhập quản trị.....	68
Hình 4.18 Tầng Controller cho trang đăng nhập quản trị	68
Hình 4.19 Bố cục trang Quản lý sản phẩm	69
Hình 4.20 Định dạng bằng CSS cho trang Quản lý sản phẩm	70
Hình 4.21 Javascript cho trang quản lí sản phẩm	70
Hình 4.22 Javascript cho trang quản lí sản phẩm chức năng thêm, sửa, xóa.....	71
Hình 4.23 Tầng DAL cho trang quản lí sản phẩm	72
Hình 4.24 Tầng BUS cho trang quản lí sản phẩm	73
Hình 4.25 Tầng Controller cho trang quản lí sản phẩm.....	73

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Trong thời đại hiện đại hóa hiện nay, không chỉ công nghệ phát triển theo từng ngày từng giờ mà ngay cả các hoạt động dịch vụ trao đổi mua bán trong cuộc sống thường ngày cũng có tính chất thay đổi hiện đại, tiện ích hơn. Và dịch vụ mua bán hải sản là một trong số những hoạt động đang rất phát triển ngày nay.

Ngày xưa, mỗi lần chúng ta muốn mua hải sản phải đến chợ hoặc siêu thị để mua. Điều này tốn rất nhiều thời gian và công sức. Vì thế, các Website mua bán online đã và đang mọc lên để giải quyết nhu cầu mua bán online.

Nắm bắt được xu hướng đó, em muốn tạo một Website về ngành dịch vụ này để giúp người dùng có thể tham khảo mua các loại hải sản một cách thuận tiện và dễ dàng.

1.2. Mục tiêu của đề tài

1.2.1 Mục tiêu tổng quát

Đề tài mua thực phẩm sẽ là đơn vị uy tín trong lĩnh vực cung cấp các sản phẩm chất lượng tới người tiêu dùng và các phương pháp quản lý thuận tiện cho người chủ người điều hành Website.

Trang web của hệ thống mua thực phẩm sẽ hiển thị các dịch vụ cho bạn như : Cho phép người mua, tối thiểu hóa các bước mua, quản lý được số lượng hàng hóa mua trong ngày,... Với những dịch vụ tiện ích hướng tới một Website mua cấp đa dạng phong phú về chủng loại sản phẩm, sự thuận tiện mua.

1.2.2 Mục tiêu cụ thể

Trang web được thiết kế sẽ có các mục để giúp người dùng dễ tìm hiểu về các loại cập hiện đang có trên thị trường ngày nay cung cấp các thông tin đầy đủ của sản phẩm.

Các dịch vụ chính của Website của hàng bán hải sản bao gồm : Quản lý sản phẩm, Quản lý hóa đơn, Quản trị, Trang người dùng.....

1.3. Giới hạn và phạm vi của đề tài

1.3.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu: Các nhà cung cấp hải sản chất lượng cao có nhu cầu bán.

Người có nhu cầu tìm hiểu, mua hải sản.

Các loại hải sản như là: Tôm, cua, cá,...

1.3.2 Phạm vi nghiên cứu

- Giới hạn của đề tài: Ứng dụng cửa hàng bán hải sản.
- Phạm vi của đề tài: Cửa hàng bán hải sản.
- Thực tiễn của đề tài : Có thể áp dụng vào các cửa hàng hay các doanh nghiệp vừa và nhỏ.

1.4. Nội dung thực hiện

- Đăng ký , đăng nhập tài khoản người dùng
- Đăng nhập tài khoản quản trị
- Quản lý sản phẩm:
 - Thêm sản phẩm
 - Sửa thông tin sản phẩm
 - Xóa thông tin về sản phẩm
 - Tìm kiếm tên sản phẩm
 - Thống kê sản phẩm: theo tình trạng.....
- Quản lý loại sản phẩm:
 - Thêm loại sản phẩm
 - Sửa thông tin loại sản phẩm
 - Xóa thông tin về loại sản phẩm
 - Tìm kiếm tên loại sản phẩm
- Quản lý quản trị
 - Thêm quản trị
 - Sửa thông tin quản trị

- Xóa thông tin về quản trị
- Tìm kiếm quản trị
- Quản lý hóa đơn
 - Thêm hóa đơn
 - Sửa thông tin hóa đơn
 - Xóa thông tin về hóa đơn
 - Tìm kiếm hóa đơn
 - Thống kê hóa đơn
- Phân hệ người dùng
 - Thêm khách hàng
 - Sửa thông tin khách hàng
 - Tìm kiếm

1.5. Phương pháp tiếp cận

- Phỏng vấn nhu cầu của khách hàng
- Tham khảo các trang mạng lớn
- Sử dụng các phương pháp nghiên cứu:
- Phương pháp đọc tài liệu.
- Phương pháp nghiên cứu bằng cách tiếp cận trực tiếp với người sử dụng thông qua việc khảo sát hệ thống cũ, đề xuất lựa chọn hệ thống mới.
- Lựa chọn công cụ lập trình và tiến hành xây dựng web thiết kế.
- Microsoft Visual Studio Code 2019: dùng để thiết kế web
- Visual Studio 2022
- API được xây dựng trên nền tảng web, sử dụng công nghệ ASP.NET(.NETframework application).
- Ngôn ngữ lập trình: C#
- Cơ sở dữ liệu: SQL server

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Quy trình phát triển phần mềm

Quy trình xây dựng phần mềm bao gồm tập hợp các thao tác và kết quả tương quan sử dụng trong việc phát triển để sản xuất ra một sản phẩm phần mềm. Mỗi một phần mềm lại có đặc điểm và yêu cầu khác nhau, tuy nhiên tất cả đều cần phải trải qua các bước sau:

Bước 1: Phân tích yêu cầu

Phân tích yêu cầu là công việc bao gồm các tác vụ xác định yêu cầu cho một hệ thống mới hoặc được thay đổi dựa trên cơ sở là các nhu cầu trong quá trình sử dụng. Việc phân tích yêu cầu có ý nghĩa quan trọng đối với thành công của một dự án. Các yêu cầu phải có tính đo được, kiểm thử được, có liên quan đến các nhu cầu hoặc cơ hội doanh nghiệp đã được xác định, và phải được định nghĩa ở mức độ chi tiết đủ cho việc thiết kế hệ thống.

Bước 2: Thiết kế phần mềm

Là một quá trình giải quyết vấn đề và lập kế hoạch cho một giải pháp phần mềm. Sau khi các mục đích và các đặc điểm kỹ thuật của phần mềm được giải quyết, lập trình viên sẽ thiết kế hoặc thuê người thiết kế để phát triển một kế hoạch cho giải pháp phần mềm. Nó bao gồm các thành phần cấp thấp, các vấn đề thuật toán cũng như một khung nhìn kiến trúc. Thiết kế chức năng, cơ sở dữ liệu và giao diện.

Bước 3: Lập trình máy tính

Lập trình máy tính (gọi tắt là lập trình) là kỹ thuật cài đặt một hoặc nhiều thuật toán trừu tượng có liên quan với nhau bằng một hoặc nhiều ngôn ngữ lập trình để tạo ra một chương trình máy tính có các thành tố nghệ thuật, khoa học, toán học, kỹ nghệ. Các ngôn ngữ lập trình khác nhau hỗ trợ các phong cách lập trình khác nhau. Một phần của công việc lập trình là việc lựa chọn một trong những ngôn ngữ phù hợp nhất với các bài toán cần giải quyết. Các ngôn ngữ lập trình khác nhau đòi hỏi lập trình viên phải xử lý các chi tiết ở mức độ khác nhau khi cài đặt các thuật

toán. Sự thống nhất trong các cách xử lý sẽ tạo thuận lợi cho việc lập trình và hiệu quả của chương trình.

Bước 4: Kiểm thử phần mềm

Kiểm thử phần mềm là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử. Kiểm thử có thể cung cấp cho doanh nghiệp một quan điểm, một cách nhìn độc đáo về phần mềm để từ đó đánh giá và thấu hiểu được những rủi ro trong quá trình triển khai phần mềm. Tùy thuộc vào từng phương pháp, việc kiểm thử có thể được thực hiện bất cứ lúc nào trong quá trình phát triển phần mềm. Theo truyền thống thì các nỗ lực kiểm thử được tiến hành sau khi các yêu cầu được xác định và việc lập trình được hoàn tất trong phương pháp phát triển “Agile” thì việc kiểm thử được tiến hành liên tục trong suốt quá trình xây dựng phần mềm. Như vậy, mỗi một phương pháp kiểm thử bị chi phối theo một quy trình phát triển phần mềm nhất định.

Bước 5: Triển khai phần mềm

Sau khi phần mềm được kiểm thử và khắc phục những sai sót sẽ được triển khai đưa vào sử dụng trong thực tế. Đối với những phần mềm thiết kế theo thỏa thuận với khách hàng, việc triển khai đơn giản chỉ là hướng dẫn cho khách hàng cách sử dụng đạt hiệu quả cao. Với những phần mềm mang tính thông dụng, việc triển khai còn qua các chương trình giới thiệu và đưa sản phẩm ra thị trường. Trong quá trình triển khai cũng luôn đánh giá hiệu quả sử dụng của phần mềm, xem xét những nhược điểm để lên kế hoạch thiết kế phần mềm hiệu quả hơn.

Bước 6: Bảo trì phần mềm

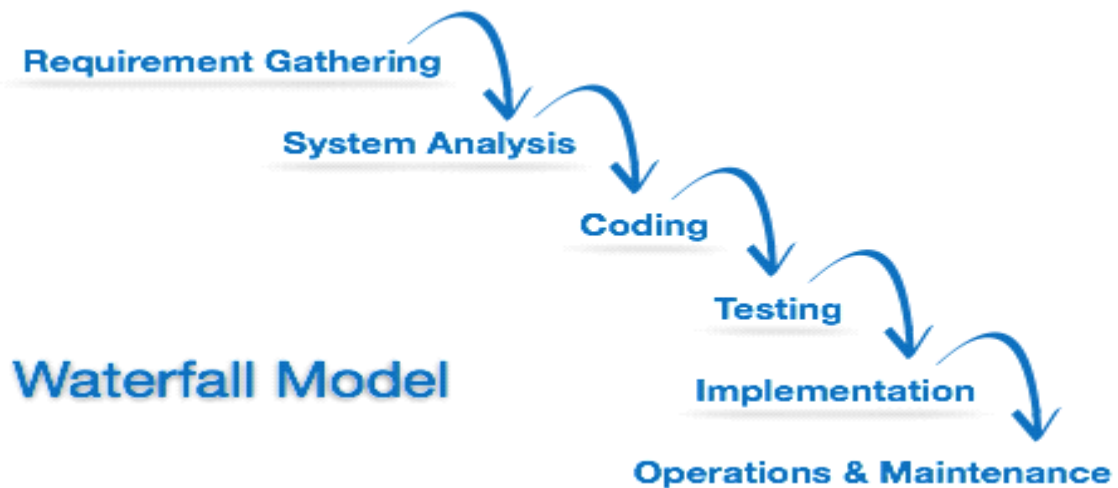
Bảo trì phần mềm bao gồm điều chỉnh các lỗi mà chưa được phát hiện trong các giai đoạn trước chu kỳ sống của phần mềm, nâng cấp tính năng sử dụng và an toàn vận hành của phần mềm. Bảo trì phần mềm có thể chiếm đến 65%-75% công sức chu kỳ sống của một phần mềm. Quá trình phát triển phần mềm bao gồm rất nhiều giai đoạn: thu thập yêu cầu, phân tích, xây dựng, kiểm tra, triển khai và bảo

trì phần mềm. Nhiệm vụ của giai đoạn bảo trì phần mềm là giữ cho phần mềm được cập nhật khi môi trường thay đổi và yêu cầu người sử dụng thay đổi.

Mỗi một giai đoạn xây dựng phần mềm lại đòi hỏi các kỹ năng phân tích và ứng dụng kiến thức công nghệ khác nhau. Để xây dựng phần mềm thiết thực, mang lại hiệu quả kinh tế cao đòi hỏi con người lập trình viên phải tuân thủ các yêu cầu trong từng giai đoạn thiết kế.

- Các mô hình phát triển phần mềm:

2.1.1 Waterfall model – Mô hình thác nước:

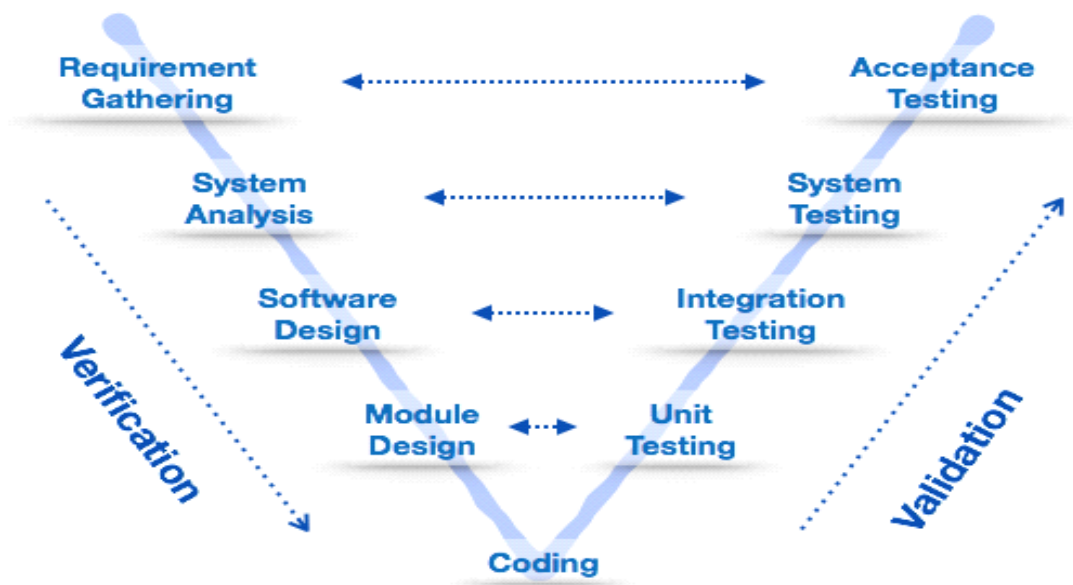


Hình 2.1 Mô hình thác nước

- Mô tả:
 - Mô hình thác nước là mô hình áp dụng theo tính tuần tự của các giai đoạn phát triển phần mềm.
 - Có nghĩa là: giai đoạn sau chỉ được phép thực hiện tiếp khi giai đoạn trước đã kết thúc.
 - Không được quay lại giai đoạn trước để xử lý các thay đổi trong yêu cầu.
 - Đây được coi là mô hình phát triển phần mềm đầu tiên.

- Áp dụng: Thường được áp dụng cho các dự án không thường xuyên thay đổi về yêu cầu.
- Đặc điểm:
 - Ưu điểm:
 - Dễ sử dụng, dễ tiếp cận.
 - Các giai đoạn và hoạt động được xác định rõ ràng.
 - Xác nhận ở từng giai đoạn, đảm bảo phát hiện sớm lỗi.
 - Nhược điểm:
 - Rất khó để quay lại giai đoạn nào đó khi nó đã kết thúc.
 - Ít tính linh hoạt và phạm vi điều chỉnh của nó khá là khó khăn, tốn kém.

2.1.2 V-Shaped Model – Mô hình chữ V:

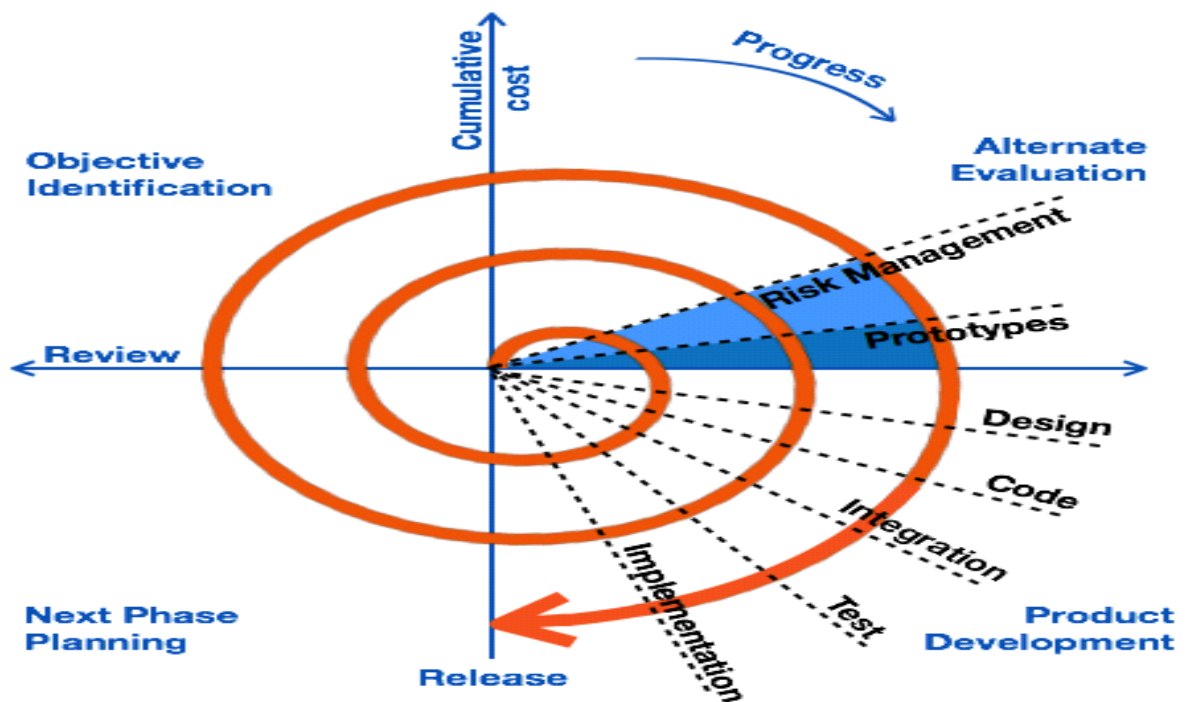


Hình 2.2 Mô hình chữ V

- Mô tả:
 - Đây là mô hình mở rộng từ mô hình thác nước

- Thay thì di chuyển xuống theo tuần tự các bước thì quy trình sẽ đi theo hình chữ V.
- Áp dụng:
 - Yêu cầu phần mềm phải xác định rõ ràng.
 - Công nghệ phần mềm và các công cụ phải được tìm hiểu kỹ.
- Đặc điểm:
 - Ưu điểm:
 - Đơn giản, dễ sử dụng.
 - Phân phối cụ thể theo mỗi giai đoạn.
 - Thực hiện verification và validation sớm trong mỗi giai đoạn phát triển.
 - Nhược điểm: Phạm vi điều chỉnh khá là khó khăn và tốn kém.

2.1.3 Spiral Model – Mô hình xoắn ốc:



Hình 2.3 Mô hình xoắn ốc.

- Mô tả:
 - Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.
 - Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.
 - Mô hình này sử dụng nhiều những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, ...
- Áp dụng: Thường được sử dụng cho các ứng dụng lớn và các hệ thống được xây dựng theo các giai đoạn nhỏ hoặc theo các phân đoạn
- Đặc điểm:
 - Ưu điểm:
 - Estimates (i.e. budget, schedule, etc.) trở nên thực tế hơn như là một quy trình làm việc, bởi vì những vấn đề quan trọng đã được phát hiện sớm hơn.
 - Có sự tham gia sớm của developers
 - Quản lý rủi ro và phát triển hệ thống theo phase
 - Nhược điểm:
 - Chi phí cao và thời gian dài để có sản phẩm cuối cùng.
 - Phải có kỹ năng tốt để đánh giá rủi ro và giả định.

2.2. Thiết kế giao diện web với HTML, CSS

2.2.1. HTML (*Hypertext Markup Language*)

❖ Ngôn ngữ HTML:

HTML (viết tắt của từ Hypertext Markup Language, hay là “Ngôn ngữ Đánh dấu Siêu văn bản”) là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Cùng với CSS và JavaScript, HTML là một trong

những ngôn ngữ quan trọng trong lĩnh vực thiết kế website. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn mực của Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, phiên bản mới nhất của ngôn ngữ này là HTML5.

❖ **Lưu ý: HTML không phải là ngôn ngữ lập trình.**

Bằng cách dùng HTML động hoặc Ajax, lập trình viên có thể được tạo ra và xử lý bởi số lượng lớn các công cụ, từ một chương trình soạn thảo văn bản đơn giản có thể gõ vào ngay từ những dòng đầu tiên cho đến những công cụ xuất bản WYSIWYG phức tạp. Hypertext là cách mà các trang Web (được thiết kế bằng HTML) được kết nối với nhau. Và như thế, đường link có trên trang Web được gọi là Hypertext. Như tên gọi đã nói, HTML là ngôn ngữ đánh dấu bằng thẻ (Markup Language), nghĩa là bạn sử dụng HTML để đánh dấu một tài liệu text bằng các thẻ (tag) để nói cho trình duyệt Web cách để cấu trúc nó để hiển thị.

❖ **Ưu, nhược điểm**

- **Ưu điểm:**

- + Có thể tích hợp được với các ngôn ngữ khác như CSS.
- + Ngôn ngữ được sử dụng rộng lớn này có rất nhiều nguồn tài nguyên hỗ trợ và cộng đồng sử dụng cực lớn.
- + Sử dụng mượt mà trên hầu hết mọi trình duyệt.
- + Có quá trình học đơn giản và trực tiếp.
- + Mã nguồn mở và hoàn toàn miễn phí.
- + Chuẩn chính của web được vận hành bởi World Wide Web Consortium.

- **Nhược điểm:**

- + Được dùng chủ yếu cho web tĩnh. Đối với các tính năng động cần sử dụng
- + JavaScript hoặc ngôn ngữ backend bên thứ 3 như PHP.
- + Một lượng lớn code phải được viết để tạo một trang web đơn giản.
- + Một số trình duyệt chậm hỗ trợ tính năng mới.
- + Tính năng bảo mật không tốt.

❖ ***Có bốn loại phân tử đánh dấu trong HTML:***

Đánh dấu Có cấu trúc miêu tả mục đích của phần văn bản

Đánh dấu trình bày miêu tả phần hiện hình trực quan của phần văn bản bất kể chức năng của nó là gì (ví dụ, `boldface` sẽ hiển thị đoạn văn bản boldface) (Chú ý là cách dùng đánh dấu trình bày này bây giờ không còn được khuyến dùng mà nó được thay thế bằng cách dùng CSS),

Đánh dấu liên kết ngoài chứa phần liên kết từ trang này đến trang kia cụ thể Các phân tử thành phần điều khiển giúp tạo ra các đối tượng (ví dụ, các nút và các danh sách).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  Nội dung trang web
</body>
</html>
```

Hình 2.4 Hình ảnh ví dụ HTML

Một số thẻ trong HTML:

Bảng 2.1 Một số thẻ trong HTML

Tag	Giải thích
<!DOCTYPE...>	Còn gọi là thẻ khai báo một tài liệu HTML. Thẻ này xác định loại tài liệu phiên bản HTML.
<html>	Thẻ này chứa đựng các tài liệu HTML đầy đủ. Ở đầu trang sẽ xuất hiện các thẻ <head>, </head> và thân tài liệu là các thẻ <body>, </body>.
<head>	Thẻ này đại diện cho đầu trang tài liệu mà có thể giữ các thẻ HTML như <title>, <link> ...
<title>	Thẻ <title> được sử dụng trong thẻ <head> chỉ tiêu đề tài liệu.
<body>	Thẻ này đại diện cho thân tài liệu và giữ các thẻ như <h1>, <div>, <p> ...
<h1>	Thẻ tag này đại diện cho các tiêu đề trang.
<p>	Thẻ tag này đại diện cho định dạng các đoạn văn trong trang web.

2.2.2. CSS (Cascading Style Sheet language)

❖ Ngôn ngữ CSS:

CSS (Cascading Style Sheets) là mã bạn sử dụng để tạo kiểu cho trang web của mình, Giống như HTML, CSS không thực sự là một ngôn ngữ lập trình. Nó không phải là một ngôn ngữ đánh dấu - đó là một ngôn ngữ định kiểu. Điều này có nghĩa là nó cho phép bạn áp dụng kiểu có chọn lọc cho các phần tử trong tài liệu HTML.

❖ **Cơ cấu bộ quy tắc CSS:**

- **Selector:** Tên phần tử HTML bắt đầu của bộ quy tắc. Nó chọn (các) phần tử được tạo kiểu (trong trường hợp này là phần tử p). Để tạo kiểu cho một phần tử khác, chỉ cần thay đổi bộ chọn.
- **Declaration:** Một quy tắc duy nhất như: color: red; xác định thuộc tính của phần tử nào bạn muốn tạo kiểu.
- **Properties:** Những cách mà bạn có thể tạo kiểu cho một phần tử HTML. (Trong trường hợp này, color là một thuộc tính của phần tử). Trong CSS, bạn chọn thuộc tính nào bạn muốn tác động trong quy tắc của mình.
- **Giá trị thuộc tính:** Ở bên phải của thuộc tính sau dấu hai chấm(:), chúng ta có giá trị thuộc tính, mà chọn một trong số nhiều lần xuất hiện có thể cho một thuộc tính cụ thể (color có rất nhiều giá trị ngoài red).

❖ **Chọn nhiều phần tử:**

Bạn cũng có thể chọn nhiều kiểu phần tử và áp dụng một quy tắc duy nhất được đặt cho tất cả các yếu tố đó. Bao gồm nhiều bộ chọn được phân biệt bởi dấu phẩy (,).

Các bộ :

Bảng 2.2 Các bộ của CSS

Tên bộ chọn	Nó chọn gì	Ví dụ
Bộ chọn phần tử (đôi khi được gọi là thẻ hoặc loại bộ chọn)	Tất cả (các) thành phần HTML của loại được chỉ định.	p Chọn <p>

Bộ chọn ID	Phần tử trên trang có ID được chỉ định. (Trên một trang HTML nhất định, bạn chỉ được phép chọn một phần tử cho mỗi ID và tất nhiên là mỗi ID cho mỗi phần tử).	#my-id Chọn <p id="my-id"> or
Bộ chọn Class	(Các) thành phần trên trang có lớp (class) được chỉ định (lớp có thể đề xuất hiện nhiều lần trên một trang).	.my-class Chọn <p class="my-class"> và
Bộ chọn thuộc tính	(Các) thành phần trên trang có thuộc tính được chọn	img[src] Chọn nhưng không phải
Bộ chọn Pseudo-class	(Các) thành phần được chỉ định, nhưng chỉ khi ở trạng thái được chỉ định, ví dụ: rê chuột.	a:hover Chọn <a>, nhưng chỉ khi con trỏ chuột đang di chuột qua liên kết.

❖ **Tác dụng của CSS:**

- Hạn chế tối thiểu việc làm rối mã HTML của trang Web bằng các thẻ quy định kiểu dáng (chữ đậm, chữ in nghiêng, chữ có gạch chân, chữ màu), khiến mã nguồn của trang Web được gọn gàng hơn, tách nội dung của trang Web và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung.

- Tạo ra các kiểu dáng có thể áp dụng cho nhiều trang Web, giúp tránh phải lặp lại việc định dạng cho các trang Web giống nhau.

❖ ***Có 3 cách để sử dụng CSS.***

- “Inline CSS”: Áp dụng trực tiếp trên một đối tượng nhất định bằng thuộc tính style.
- “Internal CSS”: Đặt CSS ở đầu trang Web để áp dụng kiểu dáng cho toàn bộ trang ấy, khi đó chỉ cần đặt đoạn CSS vào trong cặp thẻ <style> rồi đặt vào trong phần header của Web (giữa <head> và </head>).
- “External CSS”: Đặt các thuộc tính CSS vào một tệp tin riêng biệt (*.css), khi đó có thể tham chiếu đến từ nhiều trang Web khác nhau.

❖ ***Tham chiếu tới tệp tin CSS trên từ trang Web bằng đoạn mã:***

<link rel=”stylesheet” type=”text/css” href=”.../style.css”/>

2.3. Lập trình phía front-end

❖ **Ngôn ngữ JavaScript:**

JavaScript là ngôn ngữ lập trình được phát triển cho web với độ tương tác cao hơn và dần phổ biến trên nhiều lĩnh vực khác. Nó triển khai các tập lệnh phía máy khách để tương tác với người dùng trên website, tạo được các trang web động, thường được gọi là lập trình hướng đối tượng.

JavaScript được Netscape giới thiệu với tên ban đầu là LiveScript. Nhưng vào những năm 1990, sự phát triển mạnh mẽ và tác động của Java nên đơn vị này chuyển thành tên JavaScript phổ biến như ngày nay.

Hiện lập trình với JavaScript đã mở rộng ra trên nhiều lĩnh vực, từ lập trình máy tính đến lập trình thiết bị của NASA. Kể từ khi Javascript được tạo ra, nền tảng Node.js đã tạo điều kiện thuận lợi cho JavaScript phát triển với nhiều mục đích. Hãy cùng khám phá JavaScript có thể sử dụng để làm được những gì dưới đây nhé.

❖ **Tác dụng của JavaScript:**

- Phát triển website:

Công dụng phổ biến nhất của JavaScript đó là phát triển web và hiện là một trong những thành phần không thể thiếu của các trang web hoạt động hiện nay. Các nhà lập trình sử dụng JavaScript trong quá trình phát triển web để thêm tính tương tác và các tính năng nhằm cải thiện trải nghiệm người dùng, giúp trang web giàu tính tương tác hơn.

Theo thống kê từ W3techs, hơn 90% tất cả các trang web đều sử dụng JavaScript. Từ đó có thể xem Javascript là một trong những mã nguồn không thể thiếu của các web. Cải thiện khả năng tương tác với người dùng cuối (khách hàng) được nâng cao, dễ dàng thao tác, truy cập và điều hướng.

JavaScript có thể sử dụng để tạo ra các loại phần mềm khác nhau như: trò chơi, chương trình máy tính, ứng dụng web và thậm chí cả các công nghệ chuỗi khối (Blockchain) mà thời gian gần đây bạn thường nghe tới.

- Trò chơi trực tuyến ngay trên website - Games on website:

Javascript là mã nguồn không thể thiếu trong các trò chơi trực tuyến. Chắc hẳn bạn đã từng biết hay chơi thử game như: Gunny, Võ lâm truyền kỳ... đều có sử dụng công nghệ này để phát triển.

Khi HTML5 được phát hành, các nhà lập trình đã kết hợp với JavaScript để tăng thêm những trải nghiệm trong game, tăng tính tương tác và thực sự hoàn hảo với việc phát triển các trò chơi trên web.

- Phát triển các ứng dụng web - Web applications:

Nhờ có JavaScript, các nhà lập trình có thể phát triển trình ứng dụng web giàu tính tương tác với kết nối trên một sever từ xa. Một trong những ứng dụng phổ biến đó là Google Maps khi cung cấp cho người dùng nhiều thông tin trên một giao diện web đơn giản. Nhờ các framework nổi bật như VueJS, Angular hỗ trợ JavaScript trong khả năng xây dựng giao diện người dùng và kết hợp hoàn hảo với Node.js để xây dựng phần back-end (phần quản trị máy chủ, dữ liệu...).

- Ứng dụng điện thoại - Mobile applications:

Sự phát triển của các ứng dụng trên điện thoại cho cả Android và iOS cũng có sự đóng góp lớn từ JavaScript. Các nhà phát triển ứng dụng trên điện thoại sử dụng JS để xây dựng các kịch bản khác nhau, những tình huống và kết quả khác nhau từ thao tác của người dùng.

- Kết nối vạn vật - Internet of Things (IoT):

Kết nối vạn vật IoT có thể hiểu đơn giản là kết nối nhiều thiết bị công nghệ và JavaScript có khả năng tạo ra những phần mã để phục vụ mục đích này. Với sự kết hợp như Arduino cho phép viết được các mã nguồn hỗ trợ cả phần cứng lẫn phần mềm như: định vị GPS, công tắc,.....

❖ **Ưu điểm của JavaScript:**

- Ít tương tác với máy chủ hơn: Bạn có thể xác thực đầu vào của người dùng trước khi gửi trang đến máy chủ. Điều này tiết kiệm lưu lượng máy chủ, có nghĩa là tải ít hơn trên máy chủ của bạn.
- Khả năng phản hồi nhanh chóng: Khách truy cập trang web không phải chờ đợi quá lâu để tải lại trang vì có sự hỗ trợ của JavaScript.
- Tạo trang web giàu tính tương tác: Các giao diện bao gồm HTML và CSS chỉ cho người dùng những tính năng cơ bản. Khi được kết hợp với JavaScript, người dùng sẽ được trải nghiệm những phản ứng, kịch bản đã được chuẩn bị từ trước.
- Giao diện phát triển phong phú: Với hơn 90% trang web sử dụng JavaScript, các thư viện mã nguồn hỗ trợ cho JS cũng được xây dựng đa dạng. Một số thư viện mã nguồn JS có thể kể đến:
 - Algolia Places: Thư viện phát triển để hỗ trợ các form mẫu: tên, địa chỉ, ngày sinh, nghề nghiệp, học vấn...
 - AOS – Animate On Scroll: Hỗ trợ trải nghiệm chuyển động cuộn mượt mà hơn.

- Chart.js: Cung cấp các mẫu biểu đồ được viết bằng JS sẵn, người dùng chỉ cần tích hợp vào hệ thống của mình.
- ReactJS: Là một trong những thư viện quan trọng và được nhiều lập trình viên sử dụng để xây dựng giao diện người dùng dựa trên các UI đơn lẻ.

❖ Hạn chế của JavaScript:

Tuy có nhiều ưu điểm, nhưng JavaScript cũng có những nhược điểm sau:

- Dễ bị khai thác, chèn mã độc: Dễ bị khai thác, tính an ninh không cao khi phía máy người dùng truy cập không cho đọc và ghi tệp. Chính điều này gây ảnh hưởng khi đối tượng xấu có thể thực thi gán mã độc trên máy người dùng.
- Hạn chế trên một số trình duyệt: Tuy được hỗ trợ trên đa số các trình duyệt phổ biến hiện nay nhưng trong quá trình hoạt động cũng có một số trình duyệt không hỗ trợ JavaScript hoặc trên một số trang web sử dụng JS để ngăn chặn vài thao tác từ người dùng, gây khó khăn trong việc duyệt web.
- Hiện thị khác nhau trên các thiết bị khác nhau: Cùng một mã nguồn JS được viết nhưng khi trên các trình duyệt khác nhau lại tạo ra những hiển thị khác nhau, khiến các nhà lập trình cần tối ưu cho từng thiết bị và trình duyệt. Điều này có thể tạo sự không đồng nhất khi phát triển website.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Phát biểu bài toán

Xây dựng ứng dụng cho phép giới thiệu và đặt hàng online cho cửa hàng bán hải sản. Ứng dụng cho phép khách hàng có thể xem và đặt hàng online, nhân viên của cửa hàng có thể quản lý thông tin của ứng dụng và quản lý hoạt động đặt và giao hàng.

Các yêu cầu của hệ thống được mô tả như sau:

1. Cửa hàng bán các loại hải sản với các hãng khác nhau mỗi loại có thông tin (Mã loại, tên loại, nội dung). Khi cửa hàng bán một loại hải sản mới nhân viên cửa hàng có nhiệm vụ cập nhật thông tin về loại hải sản vào trong hệ thống, nếu thông tin có sai sót nhân viên có thể thực hiện sửa hoặc xóa thông tin về loại hải sản. Nhân viên có thể tìm kiếm thông tin loại hàng mà mình muốn.
2. Một loại hải sản sẽ có nhiều sản phẩm khác nhau. Mỗi sản phẩm trong cửa hàng gồm các thông tin (Mã sản phẩm, tên sản phẩm, mã loại, đơn vị tính, mô tả, hình ảnh, giá...). Khi cửa hàng bán một sản phẩm mới, nhân viên thực hiện nhập thông tin về sản phẩm vào trong hệ thống, nếu thông tin có sai sót nhân viên có thể thực hiện thêm sửa hoặc xóa thông tin của sản phẩm. Nhân viên có thể tìm kiếm thông tin sản phẩm cần tìm.
3. Giá bán của các mặt hàng được điều chỉnh theo giá bán của thị trường, khi các giá của mặt hàng thay đổi thì nhân viên có nhiệm vụ cập nhật lại giá bán, nếu giá bán có sai sót thì có thể thực hiện sửa giá bán. Giá bán của mỗi sản phẩm gồm các thông tin (mã sản phẩm, giá bán, ngày hiệu lực, ngày hết hiệu lực).
4. Cửa hàng có nhập sản phẩm của các nhà cung cấp khác nhau thông tin về nhà cung cấp gồm (Mã nhà cung cấp, tên nhà cung cấp, địa chỉ, email, số điện thoại). Khi có một nhà cung cấp mới nhân viên thực hiện nhập thông tin nhà cung cấp vào hệ thống, nếu thông tin nhà cung cấp có sai sót thì nhân

viên có thể sửa hoặc xoá thông tin. Một nhà cung cấp có thể cung cấp nhiều sản phẩm khác nhau.

5. Khi nhập hàng về cửa hàng, nhân viên có nhiệm vụ nhập sản phẩm vào cửa hàng và lưu thông tin về hoá đơn nhập hàng vào hệ thống hoá đơn nhập hàng gồm thông tin (Mã hoá đơn nhập hàng, mã nhà cung cấp, mã nhân viên, ngày nhập, thành tiền và các thông tin về sản phẩm (mã sản phẩm, số lượng, giá nhập,...)); khi thông tin về hoá đơn nhập có sai sót, hệ thống cho phép nhân viên kho có thể sửa, xoá thông tin hoá đơn nhập.
6. Khi khách hàng muốn mua sản phẩm, khách hàng vào website xem thông tin của các sản phẩm (thông tin về các sản phẩm có thể được hiển thị theo nhóm như sản phẩm mới nhập, sản phẩm bán chạy nhất, sản phẩm khuyến mại, sản phẩm theo loại sản phẩm). Khi khách hàng quan tâm đến sản phẩm nào khách hàng có thể xem thông tin chi tiết về sản phẩm đó. Trong quá trình xem thông tin sản phẩm trên website, nếu khách hàng mua sản phẩm nào sẽ chọn sản phẩm đó đưa vào giỏ hàng.
7. Sau khi chọn mua các sản phẩm trong giỏ hàng, khách hàng có thể xem thông tin các sản phẩm trong giỏ hàng, sửa số lượng sản phẩm mua hoặc xóa sản phẩm chọn mua trong giỏ hàng. Khi khách hàng muốn đặt mua, khách hàng sẽ thực hiện đặt hàng, thông tin đơn đặt hàng được lưu vào hệ thống. Thông tin của đơn hàng gồm các thông tin (Mã đơn hàng, thông tin của khách hàng, ngày đặt hàng, thành tiền, địa chỉ giao hàng (nếu có), số điện thoại nhận hàng (nếu có), trạng thái đơn hàng và các thông tin về các sản phẩm đặt mua gồm (Mã sản phẩm, số lượng, đơn giá).
8. Khi có một khách hàng mới đặt hàng lần đầu trong hệ thống, thông tin của khách hàng sẽ được lưu lại sau khi thực hiện xác minh được thông tin của khách. Thông tin của khách hàng gồm (mã khách hàng, tên khách hàng, số điện thoại, email, địa chỉ). Khách hàng có thể xem lại thông tin khách hàng, nếu có sai sót khách hàng có thể thực hiện sửa thông tin.

9. Khách hàng có thể xem lại danh sách các đơn hàng đã đặt và thông tin chi tiết từng đơn hàng đã đặt. Với các đơn hàng chưa được giao, khách hàng có thể sửa số lượng các mặt hàng đã đặt mua hoặc xoá bớt các mặt hàng đã đặt hoặc huỷ đơn đặt hàng (Để thực hiện các thao tác trên khách hàng cần đăng nhập).
10. Tất cả nhân viên thực hiện quản lý thông tin hệ thống phải đăng nhập trước khi thực hiện, thông tin nhân viên gồm (mã nhân viên, mật khẩu, vai trò).
11. Nhà vận chuyển: Xác nhận đơn hàng từ nhà vận chuyển

3.2 Đặc tả yêu cầu phần mềm

3.2.1 Các yêu cầu chức năng

a) Chức năng của phân hệ quản trị nội dung

- **Chức năng:**

- Quản lý khách hàng: Mỗi khách hàng được quản lý với các thông tin: Họ tên, Địa chỉ, Số điện thoại, Email, Tài khoản, mật khẩu,.....
- Quản lý sản phẩm: Mã sản phẩm, Tên sản phẩm, Đơn giá, Hình ảnh, Chi tiết, Mã loại....
- Quản lý loại sản phẩm: Mã loại, Tên loại, Nội dung.
- Quản lý đơn hàng: Mã đơn hàng, Mã khách hàng, Ngày giao dịch, Tổng hóa đơn. Với thông tin chi tiết: Mã chi tiết, Mã sản phẩm, Số lượng, Đơn giá.....
- Quản lí quản trị: ID quản trị, Họ tên, Địa chỉ, Tên tài khoản, mật khẩu, loại tài khoản.....

- **Biểu đồ Usecase tổng quát trang quản trị**

- ✧ *Quản lý khách hàng*

khách hàng	tại: Hiển thị thông tin, nếu không tồn tại: nhập lại hoặc thoát
11.Quản lý yêu cầu sửa thông tin khách hàng	12.HT ktra mã: nếu tồn tại→Sửa: HT tiếp tục ktra sự hợp lệ của DL sửa→Hợp lệ: Cập nhật ttin mới, mã không tồn tại: Nhập lại/Thoát
13.Quản lý yêu cầu xóa thông tin khách hàng	14.HT ktra mã đã tồn tại chưa: nếu tồn tại→Xóa các thông tin liên quan, mã không tồn tại: Nhập lại/Thoát

- **Dòng sự kiện phụ:**

Dòng phụ UC001-A1: Kiểm tra mã trùng

Bảng 3.2 Dòng sự kiện phụ(Kiểm tra mã trùng)

Hành động của tác nhân	Phản ứng của hệ thống
1. Quản lí nhập thông tin	2. Kiểm tra sự tồn tại thông tin khách hàng trong hệ thống

Dòng phụ UC001-A2: Kiểm tra tính hợp lệ của dữ liệu

Bảng 3.3 Dòng sự kiện phụ(Kiểm tra tính hợp lệ của dữ liệu)

Hành động của tác nhân	Phản ứng của hệ thống
3. Quản lí nhấn lưu thông tin	4. Hệ thống tiến hành kiểm tra các ràng buộc: sâu rồng, số lượng,... để check xem DL nhập có thỏa mãn hay không, nếu thỏa mãn thì tiến hành lưu, nếu ko thỏa mãn thì hệ thống báo lỗi

✧ Quản lý sản phẩm



Hình 3.2 Usecase quản lý sản phẩm

- Dòng sự kiện chính

Bảng 3.4 Dòng sự kiện chính

Hành động của tác nhân	Phản ứng của hệ thống
1. Quản lý yêu cầu chức năng quản lý sản phẩm	2. Hiện thị chức năng quản lý sản phẩm
3. Quản lý yêu cầu chọn chức năng	4. Hiện thị chức năng: thêm, sửa, xóa, tìm kiếm
5. Quản lý nhập thông tin sản phẩm	6. Hệ thống kiểm tra sự hợp lệ của dữ liệu, nếu hợp lệ thì tiến hành lưu
7. Quản lý nhấn enter để lưu thông tin	8. Lưu thông tin sản phẩm

9. Quản lý yêu cầu tìm kiếm thông tin sản phẩm	10. Hệ thống kiểm tra mã(tên, giá) nếu tồn tại: Hiển thị thông tin,nếu không tồn tại: nhập lại hoặc thoát
11. Quản lý yêu cầu sửa thông tin sản phẩm	12. Hệ thống ktra mã: nếu tồn tại→Sửa: HT tiếp tục ktra sự hợp lệ của DL sửa→Hợp lệ: Cập nhật ttin mới, mã không tồn tại: Nhập lại/Thoát
13. Quản lý yêu cầu xóa thông tin sản phẩm	14. Hệ thống ktra mã đã tồn tại chưa: nếu tồn tại→Xóa các thông tin liên quan, , mã không tồn tại: Nhập lại/Thoát

- ***Dòng sự kiện phụ***

Dòng phụ UC002-A1: Kiểm tra mã trùng

Bảng 3.5 Dòng sự kiện phụ(Kiểm tra mã trùng)

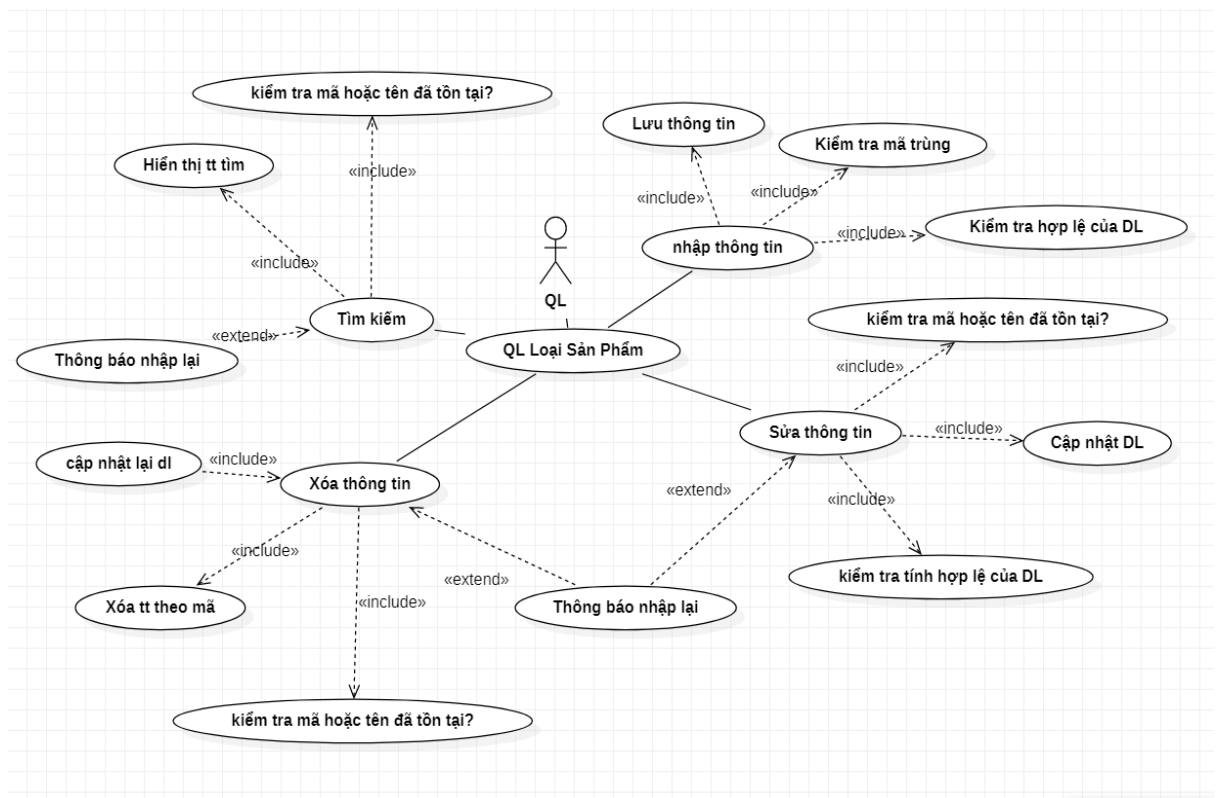
Hành động của tác nhân	Phản ứng của hệ thống
1. Quản lý nhập thông tin	2. Kiểm tra sự tồn tại thông tin sản phẩm trong hệ thống

Dòng phụ UC002-A2: Kiểm tra tính hợp lệ của dữ liệu

Bảng 3.6 Dòng sự kiện phụ(Kiểm tra tính hợp lệ của dữ liệu)

Hành động của tác nhân	Phản ứng của hệ thống
3. Quản lý nhấn lưu thông tin	4. HT tiến hành kiểm tra các ràng buộc: xâu rỗng, số lượng,... để check xem DL nhập có thỏa mãn hay không, nếu thỏa mãn thì tiến hành lưu, nếu ko thỏa mãn thì HT báo lỗi

✧ *Quản lý loại sản phẩm*



Hình 3.3 Usecase quản lý loại sản phẩm

- *Dòng sự kiện chính*

Bảng 3.7 Dòng sự kiện chính

Hành động của tác nhân	Phản ứng của hệ thống
1. Quản lý yêu cầu chức năng quản lý loại sản phẩm	2. Hiện thị chức năng quản lý loại sản phẩm
3. Quản lý yêu cầu chọn chức năng	4. Hiện thị chức năng: thêm, sửa, xóa, tìm kiếm
5. Quản lý nhập thông tin loại sản phẩm	6. Hệ thống kiểm tra sự hợp lệ của dữ liệu, nếu hợp lệ thì tiến hành
7. Quản lý nhấn enter để lưu thông tin	8. Lưu thông tin loại sản phẩm
9. Quản lý yêu cầu tìm kiếm thông tin loại sản phẩm	10. Hệ thống kiểm tra mã(tên, giá) nếu tồn tại: Hiện thị thông tin,nếu không tồn tại: nhập lại hoặc thoát
11. Quản lý yêu cầu sửa thông tin loại sản phẩm	12. Hệ thống ktra mã: nếu tồn tại→Sửa: HT tiếp tục ktra sự hợp lệ của DL sửa→Hợp lệ: Cập nhật ttin mới, mã không tồn tại: Nhập
13. Quản lý yêu cầu xóa thông tin loại sản phẩm	14. Hệ thống ktra mã đã tồn tại chưa: nếu tồn tại→Xóa các thông tin liên quan, , mã không tồn tại: Nhập lại/Thoát

- Dòng sự kiện phụ

Dòng phụ UC003-A1: Kiểm tra mã trùng

Bảng 3.8 Dòng sự kiện phụ(Kiểm tra mã trùng)

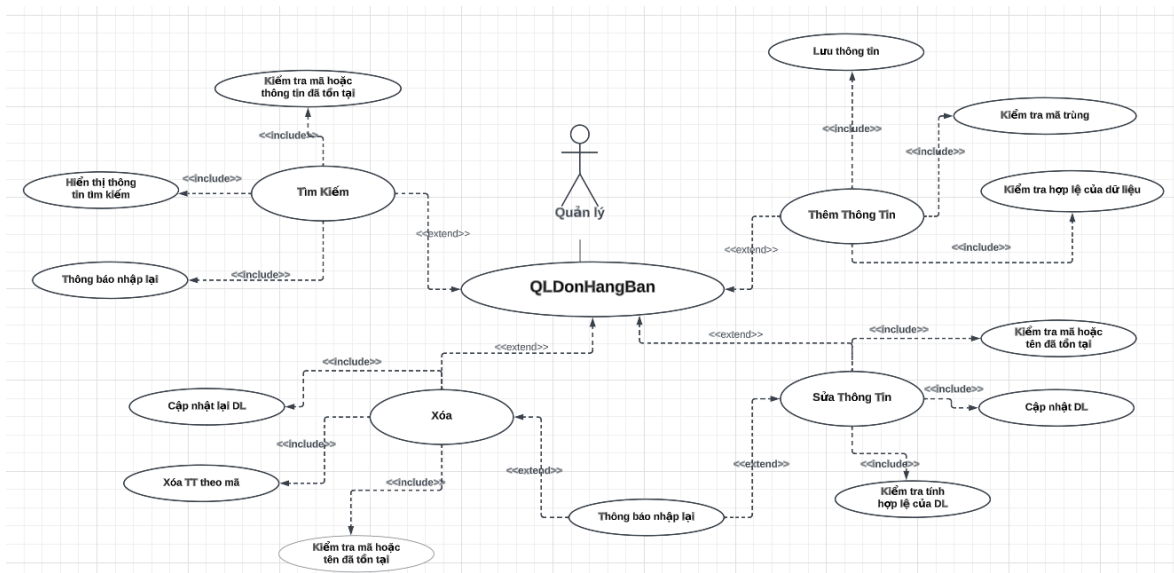
Hành động của tác nhân	Phản ứng của hệ thống
1. Quản lý nhập thông tin	2. Kiểm tra sự tồn tại thông tin loại sản phẩm trong hệ thống

Dòng phụ UC003-A2: Kiểm tra tính hợp lệ của dữ liệu

Bảng 3.9 Dòng sự kiện phụ(Kiểm tra tính hợp lệ của dữ liệu)

Hành động của tác nhân	Phản ứng của hệ thống
3. Quản lý nhân lưu thông tin	4. HT tiến hành kiểm tra các ràng buộc: xâu rỗng, số lượng,... để check xem DL nhập có thỏa mãn hay không, nếu thỏa mãn thì tiến hành lưu, nếu ko thỏa mãn thì HT báo lỗi

✧ Usecase QL Đơn hàng



Hình 3.4 Usecase quản lý đơn hàng

- Dòng sự kiện chính

Bảng 3.10 Dòng sự kiện chính

Hành động của tác nhân	Phản ứng của hệ thống
1. Quản lí yêu cầu chức năng quản lý đơn hàng bán	2. Hiện thị chức năng quản lý đơn hàng bán
3. Quản lí yêu cầu chọn chức năng	4. Hiện thị chức năng: thêm, sửa, xóa, tìm kiếm
5. Quản lí nhập thông tin đơn hàng	6. Hệ thống kiểm tra sự hợp lệ của dữ liệu, nếu hợp lệ thì tiến hành lưu
7. Quản lí nhấn enter để lưu thông tin	8. Lưu thông tin hóa đơn bán
9. Quản lí yêu cầu tìm kiếm thông tin đơn hàng bán	10. Hệ thống kiểm tra mã nếu tồn tại: Hiện thị thông tin, nếu không tồn tại: nhập lại hoặc thoát
11. Quản lí yêu cầu sửa thông tin đơn hàng bán	12. HT ktra mã: nếu tồn tại → Sửa: HT tiếp tục ktra sự hợp lệ của DL sửa → Hợp lệ: Cập nhật tin mới, mã không tồn tại: Nhập lại/Thoát
13. Quản lí yêu cầu xóa thông tin hàng	14. HT ktra mã đã tồn tại chưa: nếu tồn tại → Xóa các thông tin liên quan, mã không tồn tại: Nhập lại/Thoát

- Dòng sự kiện phụ

Dòng phụ UC005-A1: Kiểm tra mã trùng

Bảng 3.11 Dòng sự kiện phụ(Kiểm tra mã trùng)

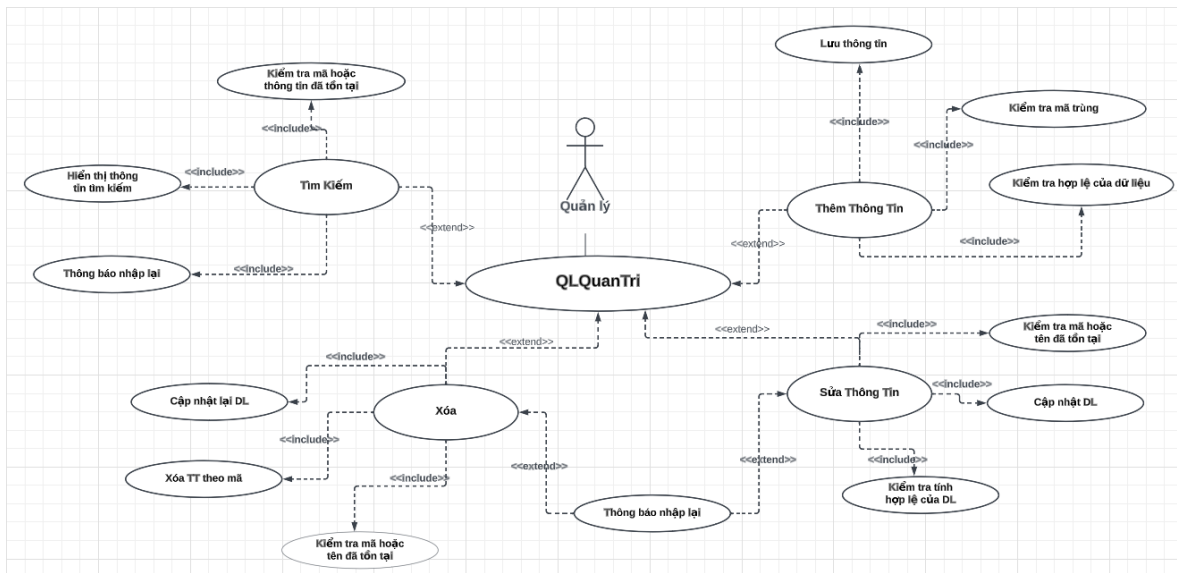
Hành động của tác nhân	Phản ứng của hệ thống
1. Quản lí nhập thông tin	2. Kiểm tra sự tồn tại thông tin hóa đơn trong hệ thống

Dòng phụ UC005-A2: Kiểm tra sự hợp lệ của dữ liệu

Bảng 3.12 Dòng sự kiện phụ(Kiểm tra sự hợp lệ của dữ liệu)

Hành động của tác nhân	Phản ứng của hệ thống
3. Quản lí nhân lưu thông tin	4. HT tiến hành kiểm tra các ràng buộc: xâu rỗng, số lượng,... để check xem DL nhập có thỏa mãn hay không, nếu thỏa mãn thì tiến hành lưu, nếu ko thỏa mãn thì HT báo lỗi

✧ *Usecase QL Tài Khoản:*



Hình 3.5 Usecase QLQuanTri

- **Dòng sự kiện chính**

Bảng 3.13 Dòng sự kiện chính

Hành động của tác nhân	Phản ứng của hệ thống
1.Quản lý yêu cầu chức năng quản lý quản trị	2.Hiển thị chức năng quản lý quản trị
3.Quản lý yêu cầu chọn chức năng	4.Hiển thị chức năng: thêm, sửa, xóa, tìm kiếm
5.Quản lý nhập thông tin quản trị	6.Hệ thống kiểm tra sự hợp lệ của dữ liệu, nếu hợp lệ thì tiến hành lưu
7.Quản lý nhấn enter để lưu thông tin	8.Lưu thông tin tài khoản
9.Quản lý yêu cầu tìm kiếm thông tin quản trị	10.Hệ thống kiểm tra mã nếu tồn tại: Hiển thị thông tin,nếu không tồn tại: nhập lại hoặc thoát
11.Quản lý yêu cầu sửa thông tin quản trị	12.HT ktra mã: nếu tồn tại→Sửa: HT tiếp tục ktra sự hợp lệ của DL sửa→Hợp lệ: Cập nhật tin mới, mã không tồn tại: Nhập lại/Thoát
13.Quản lý yêu cầu xóa thông tin quản trị	14.HT ktra mã đã tồn tại chưa: nếu tồn tại→Xóa các thông tin liên quan, , mã không tồn tại: Nhập lại/Thoát

- Dòng sự kiện phụ

Dòng phụ UC007-A1: Kiểm tra mã trùng

Bảng 3.14 Dòng sự kiện phụ(Kiểm tra mã trùng)

Hành động của tác nhân	Phản ứng của hệ thống
1.Quản lí nhập thông tin	2.Kiểm tra sự tồn tại thông tin quản trị trong hệ thống

Dòng phụ UC007-A2: Kiểm tra sự hợp lệ của dữ liệu

Bảng 3.15 Dòng sự kiện phụ(Kiểm tra sự hợp lệ của dữ liệu)

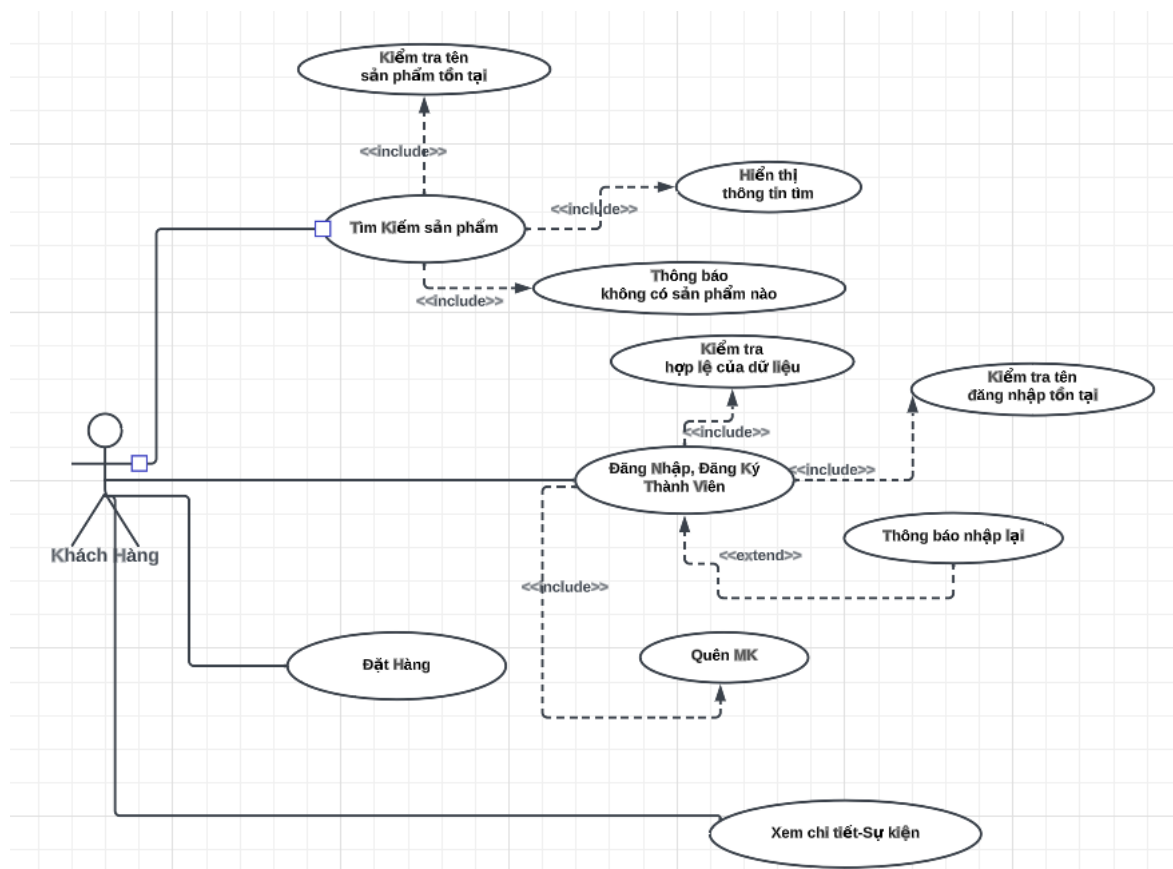
Hành động của tác nhân	Phản ứng của hệ thống
3.Quản lí nhân lưu thông tin	4.HT tiến hành kiểm tra các ràng buộc: xâu rỗng, số lượng,... để check xem DL nhập có thỏa mãn hay không, nếu thỏa mãn thì tiến hành lưu, nếu ko thỏa mãn thì HT báo lỗi

b) Chức năng của phân hệ người dùng (nếu có)

➤ **Chức năng**

- Tìm kiếm sản phẩm
- Đăng nhập, đăng ký thành viên
- Đặt hàng
- Xem chi tiết- Sự kiện

➤ **Biểu đồ Usecase phân hệ người dùng**



Hình 3.6 Usecase phân hệ người dùng

- Dòng sự kiện chính:

Bảng 3.16 Dòng sự kiện chính phân hệ người dùng

Hành động của tác nhân	Phản ứng của hệ thống
1.Khách hàng yêu cầu chức năng tìm kiếm sản phẩm	2.Hiển thị chức năng tìm kiếm sản phẩm. Hệ thống kiểm tra mã nếu tồn tại: Hiển thị thông tin,nếu không tồn tại: nhập lại hoặc thoát
3.Khách hàng yêu cầu đăng nhập/ đăng ký thành viên	4.Hiển thị chức năng đăng nhập/ Đăng ký thành viên

5. Khách hàng nhập thông tin tài khoản	6. Hệ thống kiểm tra sự hợp lệ của dữ liệu, nếu hợp lệ thì tiến hành lưu và đăng nhập hoặc đăng kí
7. Khách hàng nhấn enter để lưu thông tin/đăng nhập	8. Lưu thông tin tài khoản/Đăng nhập vào hệ thống

3.2.2 Biểu đồ lớp thực thể

Quản Trị

Danh sách thuộc tính

Bảng 3.17 Danh sách thuộc tính bảng *Quản Trị*

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	QuanTriID	String	Khác rỗng	Khóa chính
2	HoTen	String	Khác rỗng	Họ và tên
3	DiaChi	String	Khác rỗng	Địa chỉ
4	GioiTinh	String	Khác rỗng	Giới tính
5	Email	String	Khác rỗng	Địa Chỉ Email
6	VaiTro	String	Khác rỗng	Vai trò
7	TaiKhoan	String	Khác rỗng	Tài Khoản
8	MatKhau	String	Khác Rỗng	Mật Khẩu

Danh sách phương thức

Bảng 3.18 Danh sách phương thức bảng Quản Trị

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemQT()	Void	Public	Chức năng của quản lý. Thêm quản trị
2	SuaQT()	Void	Public	Chức năng của quản lý. Sửa quản trị
3	XoaQT()	Void	Public	Chức năng của quản lý. Xóa quản trị
4	TimKiem()	Void	Public	Chức năng của quản lý. Tìm kiếm tài khoản
5	DangNhap(taikhoan, Matkhau)	Void	Public	Đăng nhập hệ thống

Khách hàng

Danh sách thuộc tính

Bảng 3.19 Danh sách thuộc tính bảng Khách hàng

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	KhachHangID	String	Khác rỗng	Mã Khách Hàng
2	HoTen	String	Khác rỗng	Tên khách hàng đăng ký
3	NgaySinh	Date	Khác null	Ngày sinh của khách hàng đăng ký
4	GioiTinh	Bool	Khác null	Giới tính khách hàng
5	ĐịaChi	String	Khác rỗng	Địa chỉ khách hàng
6	SoDT	String	Khác rỗng	Số điện thoại khách hàng đăng ký
7	Email	String	Khác rỗng	Địa chỉ Email của KH
8	TaiKhoan	String	Khác rỗng	Tài khoản Khách hàng

9	MatKhau	String	Khác Rỗng	Mật khẩu Khách hàng
----------	---------	--------	-----------	---------------------

Danh sách phương thức

Bảng 3.20 Danh sách phương thức bảng Khách Hàng

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemKH()	Void	Public	Thêm khách hàng
2	SignUp(username, password)	Void	Public	Đăng nhập
4	SuaKH()	Void	Public	Sửa khách hàng

Sản phẩm

Danh sách thuộc tính

Bảng 3.21 Danh sách thuộc tính bảng Sản phẩm

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	IDHangHoa	String	Khác rỗng	Mã sản phẩm
2	TenSP	String	Khác rỗng	Tên của sản phẩm
3	Gia	Decimal	>0	Giá của sản phẩm
4	AnhSP	String	Khác rỗng	Đường dẫn hình ảnh
5	MoTa	String	Khác rỗng	Thông tin chi tiết
6	LoaiSPID	String	Khác rỗng	Mã loại sản phẩm

Danh sách phương thức

Bảng 3.22 Danh sách phương thức bảng Sản phẩm

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemSP()	Void	Public	Chức năng của quản lý. Thêm sản phẩm

2	SuaSP()	Void	Public	Chức năng của quản lý. Sửa sản phẩm
3	XoaSP()	Void	Public	Chức năng của quản lý. Xóa sản phẩm
4	TimKiem()	Void	Public	Chức năng của quản lý. Tìm kiếm sản phẩm

Chi tiết sản phẩm

Danh sách thuộc tính

Bảng 3.23 Danh sách thuộc tính bảng Chi tiết sản phẩm

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	ChiTietSPID	String	Khác rỗng	Mã Chi tiết
2	IDHangHoa	String	Khác rỗng	Mã sản phẩm
3	AnhCT	String	Khác rỗng	Đường dẫn hình ảnh
5	MoTa	String	Khác rỗng	Thông tin chi tiết

Loại sản phẩm

Danh sách thuộc tính

Bảng 3.24 Danh sách thuộc tính bảng Loại sản phẩm

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	LoaiSPID	String	Khác rỗng	Mã loại sản phẩm
2	TenLoaiSP	String	Khác rỗng	Tên của loại sản phẩm
3	NoiDung	String	Khác rỗng	Nội dung laoih sản phẩm

Danh sách phương thức

Bảng 3.25 Danh sách phương thức bảng Loại sản phẩm

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemLSP()	Void	Public	Chức năng của quản lý. Thêm loại sản phẩm
2	SuaLSP()	Void	Public	Chức năng của quản lý. Sửa loại sản phẩm
3	XoaLSP()	Void	Public	Chức năng của quản lý. Xóa loại sản phẩm
4	TimKiem()	Void	Public	Chức năng của quản lý. Tìm kiếm loại sản phẩm

Hóa Đơn

Danh sách thuộc tính

Bảng 3.26 Danh sách thuộc tính bảng Hóa Đơn

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	HoaDonID	String	Khác rỗng, Len < 10	Mã Hóa đơn
2	TenKH	String	Khác rỗng	Tên khách hàng đặt mua
3	DiaChi	String	Khác rỗng	Địa chỉ khách hàng đặt hàng
3	NgayTao	Datetime	Khác rỗng	Ngày giao dịch
4	TongGia	Double	> 0	Tổng đơn hàng

Danh sách phương thức

Bảng 3.27 Danh sách phương thức bảng Hóa Đơn

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemHD()	Void	Public	Chức năng của quản lý. Thêm hóa đơn bán
2	XoaHDB()	Void	Public	Chức năng của quản lý. Xóa hóa đơn bán
3	SuaHDB()	Void	Public	Chức năng của quản lý. Sửa hóa đơn bán
4	TimKiem()	Void	Public	Chức năng tìm kiếm hóa đơn bán

Chi Tiết Hóa Đơn

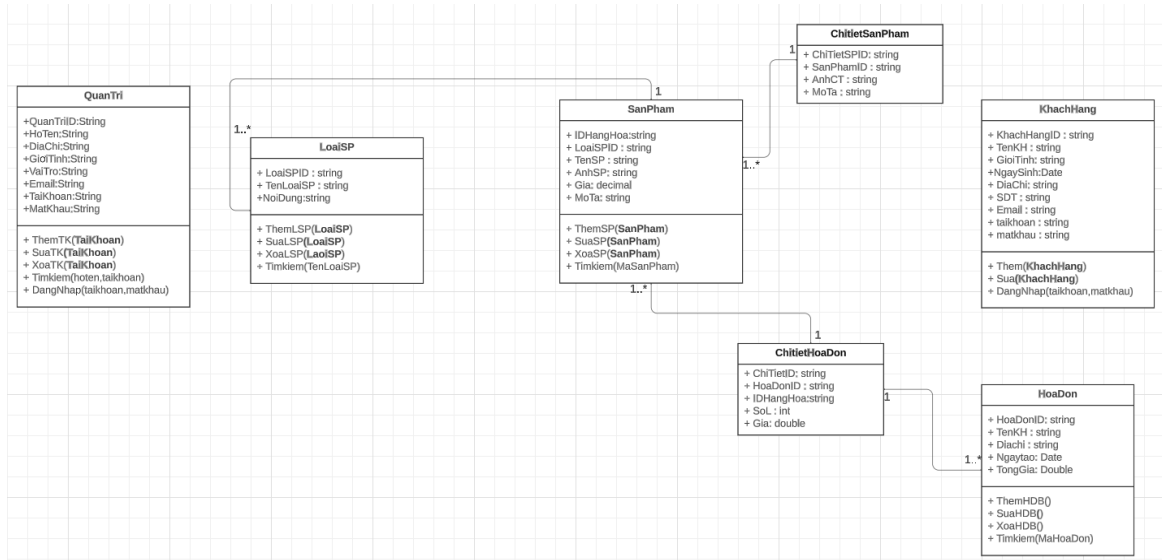
Danh sách thuộc tính

Bảng 3.28 Danh sách thuộc tính Chi tiết hóa đơn

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	ChiTietID	String	Khác rỗng, Len < 10	Mã chi tiết hóa đơn bán
2	HoaDonID	String	Khác rỗng	Mã hóa đơn bán
3	SanPhamID	String	Khác rỗng, Len < 10	Mã sản phẩm
4	SoL	Int	> 0	Số lượng

5	Gia	Double	> 0	Đơn giá
---	-----	--------	-----	---------

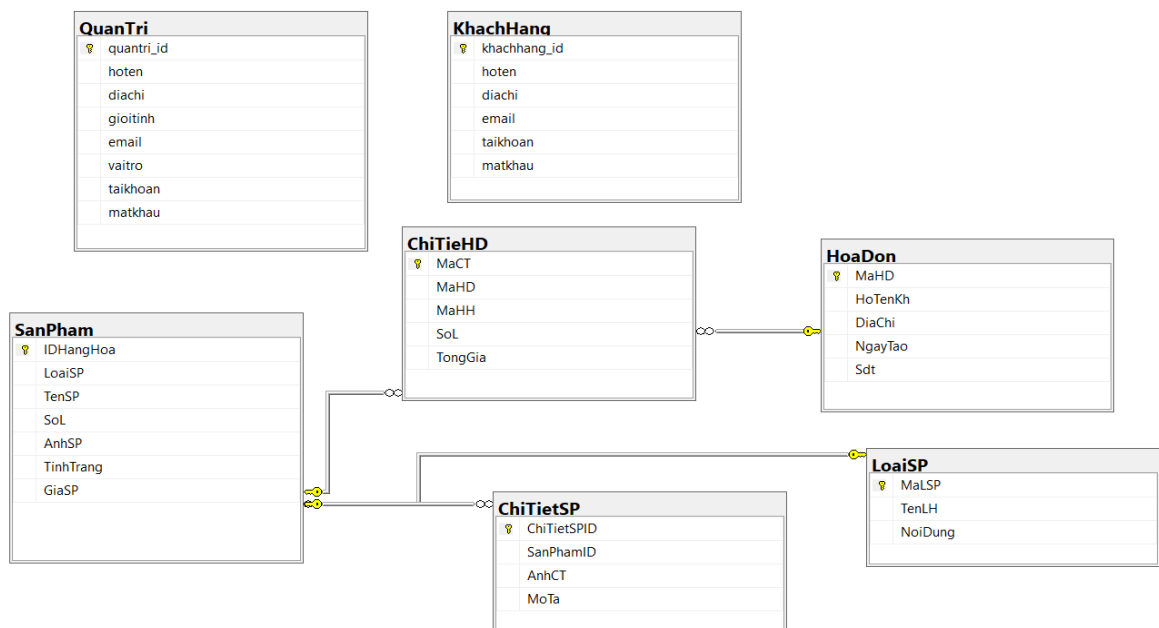
❖ Biểu đồ lớp:



Hình 3.7 Biểu đồ lớp

3.2.3 Cơ sở dữ liệu

❖ Lược đồ CSDL:



Hình 3.8 Cơ sở dữ liệu

❖ Mô tả chi tiết từng bảng trong CSDL:

Bảng 3.29 Bảng LoaiSP

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaLSP	nvarchar(10)	Khóa chính	Mã loại sản phẩm
2	TenLH	nvarchar(max)	Khác rỗng	Tên loại sản phẩm
3	NoiDung	nvarchar(500)	Khác null	Nội dung loại sản phẩm

Bảng 3.30 Bảng SanPham

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	IDHangHoa	nvarchar(10)	Khóa chính	Mã sản phẩm
2	LoaiSP	nvarchar(50)	Khóa ngoại	Mã loại sản phẩm
3	TenSP	Nvarchar(max)	Khá rỗng	Tên sản phẩm
3	SoL	Int	>0	Số lượng
4	AnhSP	nvarchar(max)	Khác rỗng	Đường dẫn hình ảnh
6	GiaSP	decimal(18,0)	Khác rỗng	Giá sản phẩm

Bảng 3.31 Bảng ChiTietSP

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	ChiTietSPID	Nvarchar(50)	Khóa chính	Mã Chi tiết
2	SanPhamID	Nvarchar(10)	Khóa ngoại	Mã sản phẩm
3	AnhCT	Nvarchar(max)	Khác rỗng	Đường dẫn hình ảnh
5	MoTa	Nvarchar(max)	Khác rỗng	Thông tin chi tiết

Bảng 3.32 Bảng HoaDon

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaHD	Nvarchar(10)	Khóa chính	Mã Hóa đơn
2	HoTenKh	Nvarchar(50)	Khác rỗng	Tên khách hàng đặt mua
3	DiaChi	Nvarchar(50)	Khác rỗng	Địa chỉ khách hàng đặt hàng
3	NgayTao	Datetime	Khác rỗng	Ngày giao dịch
4	Sdt	Nvarchar(50)	Khác rỗng	Số điện thoại khách hàng

Bảng 3.33 Bảng ChiTietHD

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaCT	Nvarchar(50)	Khóa chính	Mã chi tiết hóa đơn bán
2	MaHD	Nvarchar(10)	Khóa ngoại	Mã hóa đơn bán
3	SanPhamID	Nvarchar(10)	Khóa ngoại	Mã sản phẩm
4	SoL	Int	> 0	Số lượng

5	TongGia	Decimal	> 0	Đơn giá
---	---------	---------	-----	---------

Bảng 3.34 Bảng QuanTri

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	quantri_id	Varchar(50)	Khóa chính	Mã quản trị
2	hoten	Nvarchar(max)	Khác rỗng	Họ và tên
3	daichi	Nvarchar(250)	Khác rỗng	Địa chỉ
4	gioitinh	Nvarchar(30)	Khác rỗng	Giới tính
5	email	Nvarchar(max)	Khác rỗng	Địa Chỉ Email
6	vaitro	Nvarchar(250)	Khác rỗng	Vai trò
7	taikhoan	Nvarchar(max)	Khác rỗng	Tài Khoản
8	matkhau	Nvarchar(max)	Khác Rỗng	Mật Khẩu

Bảng 3.35 Bảng KháchHang

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	khachhang_id	nvarchar(50)	Khóa chính	Mã khách hàng
2	hoten	Nvarchar(max)	Khác rỗng	Họ và tên
3	daichi	Nvarchar(250)	Khác rỗng	Địa chỉ
4	gioitinh	Nvarchar(30)	Khác rỗng	Giới tính
5	email	Nvarchar(max)	Khác rỗng	Địa Chỉ Email
6	taikhoan	Nvarchar(max)	Khác rỗng	Tài Khoản

7	matkhou	Nvarchar(max)	Khác Rỗng	Mật Khẩu
---	---------	---------------	-----------	----------

3.2.4 Các yêu cầu phi chức năng

Yêu cầu về hiệu năng:

- Độ tin cậy
 - Khi khách hàng có yêu cầu thanh toán hóa đơn, NV thu ngân tổng hợp lại thông tin hóa đơn- tính giá trị hóa đơn chính xác, đảm bảo cung cấp thông tin chính xác đến khách hàng.
- Thời gian thực thi
 - Khi có yêu cầu thanh toán của khách hàng, kết quả thanh toán đưa ra cho khách hàng là nhanh. Có thể thực hiện tính giá trị hóa đơn cho tối đa 500 lượt khách vào trong 24 giờ
 - MUST: 100% lần thực hiện thì thời gian không quá 1 giây.
- Tính dễ sử dụng
 - Ngôn ngữ, giao diện dễ hiểu, các biểu tượng mang ý nghĩa nhất quán
 - MUST: 100% lần thực hiện đều đạt.
- Tính ổn định
 - Hệ thống luôn sẵn sàng phục vụ, hoạt động tốt 24/24
 - MUST: 100% lần thực hiện đều đạt.
- Tính bảo mật
 - Người sử dụng được đăng nhập với tài khoản duy nhất, và không thể đăng nhập với tài khoản khác.
 - Phân quyền cho người sử dụng đến từng chức năng.
 - Đảm bảo khả năng lưu trữ dữ liệu và phục hồi hệ thống.
 - MUST: 100% lần thực hiện đều đạt.

- Bảo mật: Thông tin người dùng phải được bảo mật và được mã hóa bằng MD5 Hash

Yêu cầu về sự logic của cơ sở dữ liệu:

- Các bảng dữ liệu phải có các thuộc tính của đối tượng đó, bao gồm mã, tên,... Các mã phải là khóa chính, không trùng lặp các khóa chính giữa các bảng với nhau, tên đối tượng không được để trống do người dùng sẽ để ý đến tên hơn là mã.

Các ràng buộc thiết kế:

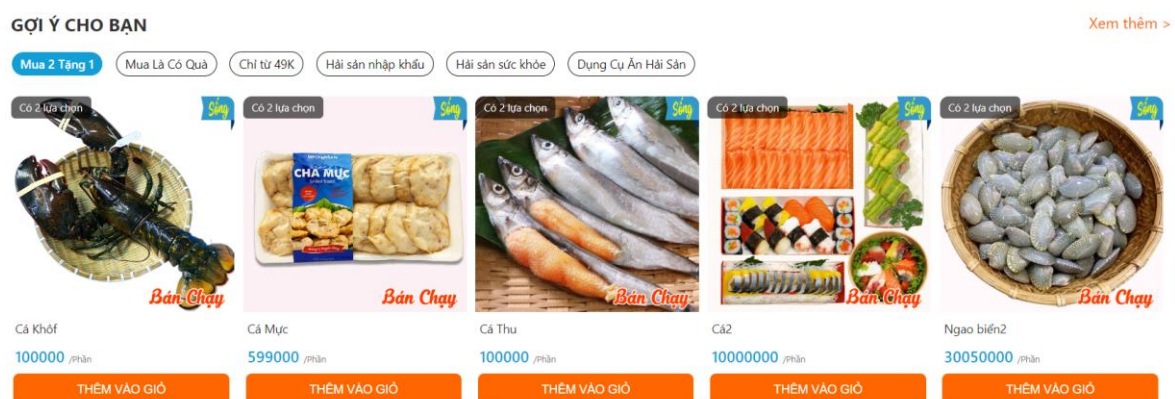
- Ngôn ngữ lập trình: HTML, CSS, Javascript, C#.....

Yêu cầu quá trình thiết kế phần mềm: Báo cáo nghiên cứu khả thi, Kế hoạch dự án, Đặc tả yêu cầu phần mềm, Tài liệu thiết kế, tài liệu kiểm thử và kết quả thử nghiệm.


- Công cụ hỗ trợ phát triển : Visual Studio Code.







Thiết kế kiến trúc và hạn chế: một tài liệu để kiến trúc và tài liệu thiết kế cần phải được gửi đến người hướng dẫn. Họ cần được rõ ràng và dễ thực hiện, duy trì hệ thống.

3.3 Thiết kế giao diện



Hình 3.9 Giao diện phần người dùng gợi ý cho bạn





Chia sẻ 116

Tôm Càng Xanh Sống

Mã SP: 05

Quy cách: : Size 12 - 13 con/kg

Tình trạng: Còn, Giao sống nước oxi hoặc đóng thùng kèm đá

Xuất xứ: Tỉnh Bến Tre - Việt Nam

Món ngon: Hấp thái, sốt trứng muối, sốt phô mai xào, rang muối Hồng Kong, sốt bơ tỏi, nấu lẩu hải sản, lẩu toyum...

Tôm Càng Xanh nhập mới mỗi ngày

329000 -23%

Quy Cách: 1Kg

Chọn

Số lượng:

🎁 Khuyến Mãi

1 📄 Hoá đơn > 1.000.000đ TẶNG Voucher **Giảm 100K Dịch Vụ Vệ Sinh Thợ Kiến** (Xem thêm tại: thokien.com)

2 📄 Hoá đơn > 1.500.000đ TẶNG 1 Phần **Verm NewZealand Sốt Bơ Tỏi**

3 📄 Hoá đơn > 3.000.000đ TẶNG 1 Phần **Sò Điệp Nhật Sốt Phô Mai Nướng**

Hình 3.10 Giao diện phân người dùng chi tiết sản phẩm



Đăng nhập

Tài khoản

Mật Khẩu

Hình 3.11 Giao diện đăng nhập quản trị

Ứng dụng cửa hàng bán hải sản

ADMIN [Đăng Xuất](#)

HỆ THỐNG QUẢN TRỊ SẢN PHẨM

Trang Chủ
Bán Hàng
Nhập Hàng
Danh Mục

Quản lý sản phẩm

Mã sản phẩm

Loại sản phẩm

Tên sản phẩm

Số lượng

Giá

Tình Trạng

Không có tệp nào được chọn

Ảnh sản phẩm

Mã sản phẩm	Tên sản phẩm	Mã Loại sản phẩm	Số Lượng	Tình Trạng	Giá	Ảnh sản phẩm	
09	Cá Khôf	02	30	Còn	100000		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
10	Cá Mực	03	60	Hết	599000		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
01	Cá Thu	03	50	Còn	100000		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
02	Cá2	03	10	Còn	10000000		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
04	Ngao biển2	01	80	Còn	30050000		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
05	Tôm Càng Xanh Sống	04	50	Còn, Giao sống nước oxi hoặc đóng thùng kèm đá	329000		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
03	Tôm giáp	04	20	Còn	30000000		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Hình 3.12 Giao diện quản lý sản phẩm

CHƯƠNG 4: TRIỂN KHAI WEBSITE

4.1 Triển khai các chức năng cho phân hệ người dùng

4.1.1 Trang chủ: Gợi ý cho bạn (Ví dụ)

a) Phía font end

➤ Xây dựng bố cục trang Home phần gợi ý cho bạn bằng các thẻ HTML

```
<div id="goiychoban " >
  <h2>
    <span>GỢI Ý CHO BẠN</span>
    <a href="#">Xem thêm ></a>
  </h2>
  <div class="tabs-nav tabs-nav-collection">
    <ul>
      <li class="is-active" data-handle="mua-2-tang-1"><span>Mua 2 Tặng 1</span></li>
      <li data-handle="mua-la-co-qua"><span>Mua Là Có Quà</span></li>
      <li data-handle="bua-ngon-tien-loi"><span>Chỉ từ 49K</span></li>
      <li data-handle="hai-san-nhap-khau"><span>Hải sản nhập khẩu</span></li>
      <li data-handle="hai-san-khoe-suc"><span>Hải sản sức khỏe</span></li>
      <li data-handle="dung-cu-an-hai-san"><span>Dụng Cụ Ăn Hải Sản</span></li>
    </ul>
  </div>
  <br>
  <div class="cards " >
    <div ng-repeat="x in listItem" class="card" style="margin-left: 0px; margin-right: 10px;" >
      <div class="product_item--image" >
        <a href="chitiet-san-pham.html"></a>
        <div class="product_item--variant--size">Có 2 lựa chọn</div>
        <div class="tag_icon"></div>
        <div class="tag_icon icon_tag_1 ratiobox">
        </div>
      </div>
      <div class="card_content">
        <h3 class="card-title" style="font-size: 14px; min-height: 2.14286em; white-space: break-spaces; font-weight: n
          text-overflow: ellipsis; margin: 0;"><a href="#">{{x.tenSP}} </a></h3>
        <div class="product_item--price">
          <span class="product_item--price--normal">{{x.giaSP}} <span
            class="product_item--variant">/Phần</span></span>
          <del></del>
        </div>
      </div>
    </div>
  </div>
```

Hình 4.1 Bố cục trang Home phần gợi ý cho bạn bằng các thẻ HTML

➤ Kỹ thuật định dạng bằng CSS

```
.product__item--action button.product__item--soldout {
  background-color: #949494;
  padding: 0.64286em 0.71429em;
  border-radius: 0.28571em;
  -moz-border-radius: 0.28571em;
  -webkit-border-radius: 0.28571em;
  color: #ffffff;
  align-items: center;
  display: inline-flex;
  justify-content: center;
}
.tag_icon.icon_tag_5, .tag_icon.icon_tag_4 {
  background: #ff7200;
  color: #fff;
  padding: 0.25em 0.5em;
  font-size: 0.85714em;
  top: auto;
  right: 0;
}
.tag_icon.icon_tag_8, .tag_icon.gift-icon {
  bottom: 4px;
  left: 0;
  top: auto;
  right: auto;
}

#goiychoban{
  height: 500px;
  width: 1345px;
}

#goiychoban>.cards>card{
  animation:translate3d(0, 0, 0);
  animation: moveSlideshow 20s linear infinite;
}
```

Hình 4.2 Định dạng bằng CSS cho trang Home phần gợi ý cho bạn

- Sử dụng Javascript, JQuery, AngularJS để lập trình các chức năng

```
var app = angular.module('AppBanHang', []);
app.controller('HomeCtrl', function ($scope, $http) {

  $scope.listItem;
  $scope.GetGoiY= function () {
    $http({
      method: 'POST',
      data: { page: 1, pageSize: 5},
      url: current_url + '/api/SanPham/SearchTenSP',
    }).then(function (response) {
      $scope.listItem = response.data.data;
    });
  };
  $scope.GetGoiY();
});
```

Hình 4.3 Javascript cho trang Home phần gợi ý cho bạn

b) Phía backend

➤ Triển khai các lớp tầng DataAccess

```
2 references
public List<ThongKeSanPhamModel> SearchTenSP(int pageIndex, int pageSize, out long total, string TenSP);
2 references
public List<ThongKeSanPhamModel> SearchTenSP(int pageIndex, int pageSize, out long total, string TenSP)
{
    string msgError = "";
    total = 0;
    try
    {
        var dt = _dbHelper.ExecuteSPProcedureReturnDataTable(out msgError, "sp_sanpham_search",
            "@page_index", pageIndex,
            "@page_size", pageSize,
            "@TenSP", TenSP);

        if (!string.IsNullOrEmpty(msgError))
            throw new Exception(msgError);
        if (dt.Rows.Count > 0) total = (long)dt.Rows[0]["RecordCount"];
        return dt.ConvertTo<ThongKeSanPhamModel>().ToList();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

Hình 4.4 Tầng DAL cho phần gợi ý cho bạn

➤ Triển khai lớp tầng Bussiness

```
public List<ThongKeSanPhamModel> SearchTenSP(int pageIndex, int pageSize, out long total, string TenSP);
2 references
public List<ThongKeSanPhamModel> SearchTenSP(int pageIndex, int pageSize, out long total, string TenSP)
{
    return _res.SearchTenSP(pageIndex, pageSize, out total, TenSP);
}
```

Hình 4.5 Tầng BUS cho phần gợi ý cho bạn

➤ Triển khai lớp tầng Controller

```
[Route("SearchTenSP")]
[HttpPost]
public IActionResult SearchTenSP([FromBody] Dictionary<string, object> formData)
{
    try
    {
        var page = int.Parse(formData["page"].ToString());
        var pageSize = int.Parse(formData["pageSize"].ToString());
        string TenSP = "";
        if (formData.Keys.Contains("TenSP") && !string.IsNullOrEmpty(Convert.ToString(formData["TenSP"]))) { TenSP = Convert.ToString(formData["TenSP"]); }

        long total = 0;
        var data = _sanphamBusiness.SearchTenSP(page, pageSize, out total, TenSP);
        return Ok(
            new
            {
                TotalItems = total,
                Data = data,
                Page = page,
                PageSize = pageSize
            }
        );
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}
```

Hình 4.6 Tầng Controller cho phần gọi ý cho bạn

4.1.2 Trang Xem Chi Tiết Sản Phẩm (Ví dụ)

a) Phía font end

➤ Xây dựng bố cục trang Xem Chi Tiết Sản Phẩm bằng các thẻ HTML

```
<div class="row">
    <div class="row_img">
        <div class="imgBox">
            
        </div>
        <ul class="thumb">
            <li ng-repeat="x in sanpham.list_json_chitietsanpham" style="margin-right: 10px;">
                <a href="{{x.anhCT}}" target="imgBox">
                    </script>
        <script src="chitietsanpham.js"></script>
        <div class="social">...
        </div>
    </div>
    <div class="row_content">
        <!-- Product Title -->
        <h1 class="product_detail--title">{{sanpham.tenSP}}</h1>
        <p class="product_detail--sku"><strong>Mã SP:</strong> {{sanpham.idHangHoa}}</p>
        <!-- Short Description -->
        <div class="product_detail--short-des">
            <p><strong>Quy cách:</strong> Size 12 - 13 con/kg</p>
            <p><strong>Tình trạng:</strong>{{sanpham.tinhTrang}}&nbsp;</p>
            <p><strong>Xuất xứ:</strong> Tỉnh Bến Tre - Việt Nam</p>
            <p><strong>Món ngon:</strong> Hấp thái, sốt trứng muối, sốt phô mai xào, rang muối Hồng Kong, sốt bơ tỏi, nấu lẩu hải sản, lẩu toyum...</p>
            <p><strong>Tôm Càng Xanh</strong>nhập mới mỗi ngày</p>
        </div>
        <!-- Price -->
        <div class="product_detail--price" id="price-preview">
            <span class="price has-discount">{{sanpham.giaSP}}</span>
            <del></del>
            <span class="product_detail--price--sale">-23%</span>
        </div>
    </div>
</div>
```

Hình 4.7 Bố cục trang xem chi tiết sản phẩm bằng các thẻ HTML

➤ Kỹ thuật định dạng bằng CSS

```
.product__item {
  padding: 0.35714em;
  margin-bottom: 15px;
  background: #fff;
}
.product__item--image {
  position: relative;
  padding-bottom: 100%;
}
.product__item--image a img {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  display: flex;
}
.tag_icon {
  position: absolute;
  top: 0;
  right: -0.57143em;
  z-index: 10;
}
.product__item--info {
  padding-top: 0.71429em;
}
.product__item--title {
  overflow: hidden;
  min-height: 2.14286em;
  font-size: 14px;
  white-space: break-spaces;
  font-weight: normal;
  display: block;
  display: -webkit-box;
  -webkit-line-clamp: 2;
  -webkit-box-orient: vertical;
}
```

Hình 4.8 Định dạng bằng CSS cho trang xem chi tiết sản phẩm

➤ Sử dụng Javascript, JQuery, AngularJS để lập trình các chức năng

```
var app = angular.module('AppBanHang', []);
app.controller("ChiTietCtrl", function ($scope, $http) {
  $scope.sanpham;
  $scope.GetSanPham= function () {
    $http({
      method: 'GET',
      url: current_url + '/api/SanPham/get-by-id/05',
    }).then(function (response) {
      debugger;
      $scope.sanpham = response.data;
    });
  };
  $scope.GetSanPham();
});
```

Hình 4.9 Javascript cho trang xem chi tiết sản phẩm

b) Phía backend

- Triển khai các lớp tầng DataAccess

```

2 references
SanPhamModel GetDatabyID(string id);
2 references

public SanPhamModel GetDatabyID(string id)
{
    string msgError = "";
    try
    {
        var dt = _dbHelper.ExecuteSPProcedureReturnDataTable(out msgError, "sp_sanpham_get_by_id",
            "@IDHangHoa", id);
        if (!string.IsNullOrEmpty(msgError))
            throw new Exception(msgError);
        return dt.ConvertTo<SanPhamModel>().FirstOrDefault();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Hình 4.10 Tầng DAL cho trang xem chi tiết sản phẩm

- Triển khai lớp tầng Bussiness

```

2 references
SanPhamModel GetDatabyID(string id);
2 references

public SanPhamModel GetDatabyID(string id)
{
    return _res.GetDatabyID(id);
}

```

Hình 4.11 Tầng BUS cho trang xem chi tiết sản phẩm

- Triển khai lớp tầng Controller

```

[Route("get-by-id/{id}")]
[HttpGet]
0 references
public SanPhamModel GetDatabyID(string id)
{
    return _sanphamBusiness.GetDatabyID(id);
}

```

Hình 4.12 Tầng Controller cho trang xem chi tiết sản phẩm

4.2 Triển khai các chức năng cho phân hệ quản trị nội dung (nếu có)

4.2.1 Đăng nhập quản trị(Ví dụ)

a) Phía font end

- Xây dựng bố cục trang Home phần gợi ý cho bạn bằng các thẻ HTML

```
<form >
  <div class="anh">
    
  </div>
  <div class="content">
    <h1>Đăng nhập</h1>
    <label >Tài khoản</label><br>
    <input type="text" id="username"><br>
    <lable>Mật Khẩu</lable><br>
    <input type="password" id="password"><br><br>
    <!-- <input type="button" onclick="Login()" value="Đăng nhập"></input> -->
    <button type="button" onclick="Login()">Đăng nhập</button>
  </div>
</form>
```

Hình 4.13 *Bố cục trang đăng nhập quản trị bằng các thẻ HTML*

- Kỹ thuật định dạng bằng CSS

```
form{
  display: flex;
  margin-top: 100px;
  background-color: blue;
  padding: 20px;
  width: 700px;
  margin-left: 250px;
}
.anh img{
  height: 250px;
  width: 250px;
}
.content{
  padding-left: 50px;
  color: white;
  line-height: 1.5;
}
h1{
  margin: 0px !important;
}
input{
  width: 350px;
  outline: 0;
  height: 30px;
}
button{
  width: 100px;
  height: 30px;
  border: 1px solid white;
}
button:hover{
  background-color: orange;
  color: white;
  border: 1px solid orange;
}
```

Hình 4.14 Định dạng bằng CSS cho trang Đăng nhập quản trị

- Sử dụng Javascript, JQuery, AngularJS để lập trình các chức năng

```
<script>
function Login() {
    var item = {};
    item.Username = $("#username").val();
    item.Password = $("#password").val();
    $.ajax({
        type: "POST",
        url: "http://localhost:19711/api/QuanTri/login_quantri",
        dataType: "json",
        contentType: 'application/json',
        data: JSON.stringify(item)
    }).done(function (data) {
        debugger;
        if (data != null && data.message != null && data.message != 'undefined') {
            alert(data.message);
        } else {
            localStorage.setItem("user", JSON.stringify(data));
            window.location.href = "QuanTriSanPham.html";
        }
    }) .fail(function() {
        alert('Tài khoản hoặc mật khẩu không đúng');
    });
};
</script>
```

Hình 4.15 Javascript cho trang Đăng nhập quản trị

b) Phía backend

- Triển khai các lớp tầng DataAccess

```
QuanTriModel Login(string taikhoan, string matkhau);

public QuanTriModel Login(string taikhoan, string matkhau)
{
    string msgError = "";
    try
    {
        var dt = _dbHelper.ExecuteSPProcedureReturnDataTable(out msgError, "sp_quantrilogin",
            "@taikhoan", taikhoan,
            "@matkhau", matkhau
        );
        if (!string.IsNullOrEmpty(msgError))
            throw new Exception(msgError);
        return dt.ConvertTo<QuanTriModel>().FirstOrDefault();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

Hình 4.16 Tầng DAL cho trang Đăng nhập quản trị

➤ Triển khai lớp tầng Bussiness

```
QuanTriModel Login(string taikhoan, string matkhau);

public QuanTriModel Login(string taikhoan, string matkhau)
{
    var user = _res.Login(taikhoan, matkhau);
    if (user == null)
        return null;
    var tokenHandler = new JwtSecurityTokenHandler();
    var key = Encoding.ASCII.GetBytes(secret);
    var tokenDescriptor = new SecurityTokenDescriptor
    {
        Subject = new ClaimsIdentity(new Claim[]
        {
            new Claim(ClaimTypes.Name, user.taikhoan.ToString()),
            new Claim(ClaimTypes.Email, user.email)
        }),
        Expires = DateTime.UtcNow.AddDays(7),
        SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(key), SecurityAlgorithms.Aes128CbcHmacSha256)
    };
    var token = tokenHandler.CreateToken(tokenDescriptor);
    user.token = tokenHandler.WriteToken(token);
    return user;
}
```

Hình 4.17 Tầng BUS cho trang đăng nhập quản trị

➤ Triển khai lớp tầng Controller

```
[AllowAnonymous]
[HttpPost("login_quantri")]
0 references
public IActionResult Login([FromBody] AuthenticateModel model)
{
    var user = _quantriBusiness.Login(model.Username, model.Password);
    if (user == null)
        return BadRequest(new { message = "Tài khoản hoặc mật khẩu không đúng!" });
    return Ok(new { taikhoan = user.taikhoan, email = user.email, token = user.token });
}
```

Hình 4.18 Tầng Controller cho trang đăng nhập quản trị

4.2.2 Quản lý sản phẩm(Ví dụ)

a) Phía front end

➤ Xây dựng bố cục trang Home phần gợi ý cho bạn bằng các thẻ HTML

```
<body class="no-skin" ng-app="AppBanHang" ng-controller="SanPhamCtrl">
  <div id="contain">
    <div id="header">...
    </div>
    <div id="main">
      <div class="pad-main">
        <div class="menu">...
        </div>
        <div class="content">
          <div class="title_cnt">
            <i class="fa fa-arrow-circle-o-right" style="font-size: 40px; color: blue;"></i>
            <h2>Quản lý sản phẩm</h2>
          </div>
          <div class="row">
            <form class="form_main" role="form">
              <table>
                <tr>
                  <td>Mã sản phẩm</td>
                  <td class="tdthongtin"><input ng-model="IDHangHoa" type="text" id="masanpham"></td>
                </tr>
                <tr>
                  <td>Loại sản phẩm</td>
                  <td class="tdthongtin">
                    <select ng-model="MaLSP" id="loaisanpham">
                      <option ng-repeat="x in listLoaiSP" value="{{x.maLSP}}">{{x.maLSP}}</option>
                    </select>
                  </td>
                </tr>
                <tr>
                  <td>Tên sản phẩm</td>
                  <td class="tdthongtin"><input ng-model="TenSP" type="text" id="tensanpham"></td>
                </tr>
                <tr>
                  <td>Số lượng</td>
                  <td class="tdthongtin"><input ng-model="Sol" type="number" id="SoLuong"></td>
                </tr>
              </table>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
```

Hình 4.19 *Bố cục trang Quản lý sản phẩm*

➤ Kỹ thuật định dạng bằng CSS

```
.form_main{
    height:auto;
    width: 98%;
    margin: auto;
    line-height: 55px;
}

#masanpham{
    width: 550px;
    height: 40px;
}
#loaisanpham{
    width: 550px;
    height: 40px;
}
#tensanpham{
    width: 550px;
    height: 40px;
}
#SoLuong{
    width: 550px;
    height: 40px;
}
```

Hình 4.20 Định dạng bằng CSS cho trang Quản lý sản phẩm

- Sử dụng Javascript, JQuery, AngularJS để lập trình các chức năng

```
$scope.LoadSanPham = function () {
    $http({
        method: 'POST',
        headers: { "Authorization": 'Bearer ' + _user.token },
        data: { page: $scope.page, pageSize: $scope.pageSize },
        url: current_url + '/api/SanPham/SearchTenSP',
    }).then(function (response) {
        $scope.listSanPham = response.data.data;
    });
};

$scope.LoadLoaiSP = function () {
    $http({
        method: 'GET',
        headers: { "Authorization": 'Bearer ' + _user.token },
        url: current_url + '/api/LoaiSanPham/getallsp',
    }).then(function (response) {
        $scope.listLoaiSP = response.data;
    });
};
```

Hình 4.21 Javascript cho trang quản lý sản phẩm

```

$scope.Save = function () {
    let obj = {};
    obj.IDHangHoa = $scope.IDHangHoa;
    obj.LoaiSP = $scope.MaLSP;
    obj.TenSP = $scope.TenSP;
    obj.SoL = Number($scope.SoL);
    obj.GiaSP = Number($scope.GiaSP);
    obj.TinhTrang = $scope.TinhTrang;
    obj.list_json_chitietsanpham = [];
    var file = document.getElementById('file').files[0];
    if (file) {
        var src = urlSaveImg + file.name;
        obj.AnhSP = src;
    } else {
        obj.AnhSP = $scope.AnhSP;
    }
    if ($scope.submit == "Thêm mới") {
        $http({
            method: 'POST',
            headers: { "Authorization": 'Bearer ' + _user.token },
            data: obj,
            url: current_url + '/api/SanPham/create-sanpham',
        }).then(function (response) {
            $scope.LoadSanPham();
            alert('Thêm sản phẩm thành công!');
        });
    } else {
        $http({
            method: 'POST',
            headers: { "Authorization": 'Bearer ' + _user.token },
            data: obj,
            url: current_url + '/api/SanPham/update-sanpham',
        }).then(function (response) {
            $scope.LoadSanPham();
            alert('Cập nhật sản phẩm thành công!');
        });
    }
}

$scope.Xoa = function (id) {
    var result = confirm("Bạn có thực sự muốn xóa không?");
    if (result) {
        $http({
            method: 'GET',
            headers: { "Authorization": 'Bearer ' + _user.token },
            url: current_url + '/api/SanPham/delete-sanpham/' + id,
        }).then(function (response) {
            $scope.LoadSanPham();
            alert('Xóa thành công!');
        });
    }
}
};

```

Hình 4.22 Javascript cho trang quản lý sản phẩm chức năng thêm, sửa, xóa

b) Phía backend

➤ Triển khai các lớp tầng DataAccess

```

2 references
bool Create(SanPhamModel model);
2 references
bool Update(SanPhamModel model);
2 references
SanPhamModel Delete(string id);

```

```

public bool Create(SanPhamModel model)
{
    string msgError = "";
    try
    {
    }
}
2 references
public bool Update(SanPhamModel model)
{
    string msgError = "";
    try
    {
        var result = _dbHelper.ExecuteScalarSPProcedureWithTransaction(out msgError, "sp_sanpham_update",
            "@IDHangHoa", model.IDHangHoa,
            "@LoaiSP", model.LoaiSP,
            "@TenSP", model.TenSP,
            "@SoL", model.SoL,
            "@AnhSP", model.AnhSP,
            "@TinhTrang", model.TinhTrang,
            "@GiaSP", model.GiaSP,
            "@list_json_chitietsanpham", model.list_json_chitietsanpham != null ? MessageConvert.SerializeObject(model.list_json_chitietsanpham)
            if ((result != null && !string.IsNullOrEmpty(result.ToString())) || !string.IsNullOrEmpty(msgError))
            {
                throw new Exception(Convert.ToString(result) + msgError);
            }
        return true;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
2 references
public SanPhamModel Delete(string id)
{
    string msgError = "";
    try
    {
        var dt = _dbHelper.ExecuteSPProcedureReturnDataTable(out msgError, "sp_sanpham_delete",
            "@IDHangHoa", id);
        if (!string.IsNullOrEmpty(msgError))
            throw new Exception(msgError);
        return dt.ConvertTo<SanPhamModel>().FirstOrDefault();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Hình 4.23 Tầng DAL cho trang quản lý sản phẩm

➤ Triển khai lớp tầng Bussiness


```

bool Create(SanPhamModel model);
2 references
bool Update(SanPhamModel model);
2 references
SanPhamModel Delete(string id);
2 references

public bool Create(SanPhamModel model)
{
    return _res.Create(model);
}
2 references
public bool Update(SanPhamModel model)
{
    return _res.Update(model);
}
2 references
public SanPhamModel Delete(string id)
{
    return _res.Delete(id);
}

```

Hình 4.24 Tầng BUS cho trang quản lý sản phẩm

➤ Triển khai lớp tầng Controller

```

[Route("create-sanpham")]
[HttpPost]
0 references
public SanPhamModel CreateItem([FromBody] SanPhamModel model)
{
    _sanphamBusiness.Create(model);
    return model;
}

[Route("update-sanpham")]
[HttpPost]
0 references
public SanPhamModel Update([FromBody] SanPhamModel model)
{
    _sanphamBusiness.Update(model);
    return model;
}

[Route("delete-sanpham/{id}")]
[HttpGet]
0 references
public SanPhamModel Delete(string id)
{
    return _sanphamBusiness.Delete(id);
}

```

Hình 4.25 Tầng Controller cho trang quản lý sản phẩm

4.3 Kiểm thử và triển khai ứng dụng

4.3.1 Kiểm thử

- Để mua người dùng sẽ chọn sản phẩm và điền đầy đủ các thông tin cần thiết để mua hàng.
- Khi chọn sản phẩm người dùng sẽ đc giới thiệu qua hình ảnh, các tiêu chí đánh giá và sau đó người dùng sẽ cân nhắc mua hàng.
- Khi quản trị thêm sản phẩm sẽ phải nhập các mục yêu cầu.....

4.3.2 Đóng gói ứng dụng

- Nén project source (.rar)
- Xuất file databale (.sql)

4.3.3 Triển khai ứng dụng

- Hệ điều hành Window XP hoặc cao hơn
- NetFramework: sản phẩm cài đặt .NET Framework 2.0 trở lên
- Công cụ phát triển: Visual Studio 2010 trở lên

KẾT LUẬN

Kết quả đạt được:

- Trong thời gian thực hiện bài tập lớn, em đã hoàn thành được ứng dụng cửa hàng bán hải sản với các chức năng: Quản lý sản phẩm, quản lý hóa đơn, quản lý loại sản phẩm, Đăng nhập...
- Thể hiện được các sản phẩm của cửa hàng trên chương trình.....
- Hiểu được nghiệp vụ quản lý của ứng dụng bán hàng

Hạn chế của đề tài

- Chương trình chưa tối ưu
- Kiểm tra dữ liệu chưa chặt chẽ
- Do thời gian có hạn và lượng kiến thức còn hạn hẹp nên em chưa hoàn thiện một cách tốt nhất và còn nhiều sai sót những chức năng trong sản phẩm của mình.
- Một số chức năng hoạt động chưa khoa học.
- Quy mô của đề tài còn nhỏ hẹp, chỉ áp dụng được với một số cửa hàng bán hải sản như kiểu gia đình.
- Chương trình còn chưa được đẹp mắt, tính thẩm mỹ chưa cao
- Còn hạn chế về thời gian

Hướng phát triển của đề tài

- Phát triển ứng dụng bằng việc mở rộng thêm các chức năng đáp ứng nhu cầu của người dùng
- Tối ưu hóa các chức năng của hệ thống để đáp ứng tốt hơn nhu cầu của người sử dụng.

- Cần bổ sung thêm các chức năng, giao diện thân thiện và sát với yêu cầu hơn.
- Kiểm tra các dữ liệu một cách chặt chẽ, khoa học và có tính đúng đắn cao.

TÀI LIỆU THAM KHẢO

- [1] K. CNTT, Công nghệ phần mềm, Hưng Yên: Đại Học SPKT Hưng Yên, 2022.
- [2] Elisabeth Robson, Eric Freeman (2012), Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages, O'Reilly Media, Neri XB, Newton, Massachusetts, Hoa Kỳ
- [3] J.D Gauchat, “HTML5 for Masterminds,” 3rd Edition, Kindle Edition, 2018
Trang web: <https://www.w3schools.com/>
Trang web: [Search | Font Awesome](#)
Trang web: [Browse Fonts - Google Fonts](#)
Trang web: <https://www.codingsnow.com/2022/02/responsive-website-landing-page-with.html>
Trang web: <https://fptshop.com.vn/tin-tuc/danh-gia/javascript-la-gi-151984>
- [4] Khoa Công nghệ thông tin, CĐ.Công nghệ.Net, Đại học SPKT Hưng Yên.
- [5] Phạm Hữu Khang (2000), “Lập trình cơ sở dữ liệu,” NXB LĐTĐ và XH