

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN CÔNG NGHỆ PHẦN MỀM



CAO NGỌC THIỆN : 15110320

PHẠM LƯƠNG MỸ : 15110253

Đề tài:

PHÁT TRIỂN ỨNG DỤNG WEB BÁN HÀNG TRÊN CÔNG NGHỆ .NET CORE

KHÓA LUẬN TỐT NGHIỆP KỸ SƯ CNTT

GIÁO VIÊN HƯỚNG DẪN

TS. LÊ VĂN VINH

KHÓA 2015 - 2019

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN CÔNG NGHỆ PHẦN MỀM



CAO NGỌC THIỆN : 15110320

PHẠM LƯƠNG MỸ : 15110253

Đề tài:

PHÁT TRIỂN ỨNG DỤNG WEB BÁN HÀNG TRÊN CÔNG NGHỆ .NET CORE

KHÓA LUẬN TỐT NGHIỆP KỸ SƯ CNTT

GIÁO VIÊN HƯỚNG DẪN

TS. LÊ VĂN VINH

KHÓA 2015 - 2019

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên: Cao Ngọc Thiện

MSSV: 15110320

Họ và tên sinh viên: Phạm Lương Mỹ

MSSV: 15110253

Ngành: Công nghệ thông tin

Tên đề tài: Phát triển ứng dụng web bán hàng trên công nghệ .NET Core

Họ và tên Giáo viên hướng dẫn: TS. Lê Văn Vinh

NHẬN XÉT:

1. Về nội dung đề tài và khối lượng thực hiện:

.....
.....
.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....
.....
.....

3. Khuyết điểm:

.....
.....
.....
.....

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

Tp.Hồ Chí Minh, ngày ... tháng ... năm 201

Giáo viên hướng dẫn

(Ký và ghi rõ họ tên)

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên sinh viên: Cao Ngọc Thiện

MSSV: 15110320

Họ và tên sinh viên: Phạm Lương Mỹ

MSSV: 15110253

Ngành: Công nghệ thông tin

Tên đề tài: Phát triển ứng dụng web bán hàng trên công nghệ .NET Core

Họ và tên Giáo viên phản biện: ThS. Nguyễn Hữu Trung

NHẬN XÉT:

1. Về nội dung đề tài và khối lượng thực hiện:

.....
.....
.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....
.....
.....

3. Khuyết điểm:

.....
.....
.....
.....

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

Tp.Hồ Chí Minh, ngày ... tháng ... năm 201

Giáo viên phản biện

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của bất kỳ ai khác. Trong suốt thời gian học tập ở trường, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè.

Trước hết, chúng em xin kính gửi đến thầy Lê Văn Vinh lời chúc sức khỏe. Chúng em xin chúc thầy luôn thành công trong sự nghiệp giáo dục đào tạo cũng như mọi lĩnh vực trong cuộc sống. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình, chu đáo của thầy, đến nay chúng em đã hoàn thành được bài báo cáo luận văn tốt nghiệp đề tài: “Tìm hiểu .NET Core và phát triển ứng dụng website bán hàng”.

Chân thành cảm ơn tập thể Giáo viên Khoa Công nghệ thông tin, trường Đại học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh về những bài giảng trong suốt khóa học, về những kiến thức nền tảng cũng như những kiến thức về chuyên ngành đã giúp cho chúng em hoàn thành tốt đề tài này. Chúng em xin chúc các Thầy Cô luôn thành công trong sự nghiệp giảng dạy cũng như mọi lĩnh vực trong cuộc sống.

Chúng em xin chân thành gửi lời cảm ơn đến Khoa Công nghệ thông tin, trường Đại học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh đã tạo điều kiện cho chúng em thực hiện đề tài này.

Xin chân thành cảm ơn các anh chị và bạn bè đã ủng hộ, giúp đỡ, động viên chúng em trong thời gian học tập và nghiên cứu.

Trong phạm vi khả năng cho phép, chúng em đã rất cố gắng để hoàn thành đề tài một cách tốt nhất. Song, chắc chắn sẽ không tránh khỏi những thiếu sót. Chúng em kính mong nhận được sự cảm thông và những ý kiến đóng góp quý báu của Thầy Cô để em học hỏi thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn trong tương lai.

Một lần nữa chúng em xin chân thành cảm ơn!

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ CƯƠNG LUẬN VĂN TỐT NGHIỆP

Họ và tên sinh viên thực hiện: Cao Ngọc Thiện MSSV: 15110320

Họ và tên sinh viên thực hiện: Phạm Lương Mỹ MSSV: 15110253

Thời gian làm luận văn từ 15/03/2019 đến 10/07/2019

Chuyên ngành: Công nghệ phần mềm

Tên luận văn: Phát triển ứng dụng web bán hàng trên công nghệ .NET Core.

Giáo viên hướng dẫn: TS. Lê Văn Vinh

Nội dung thực hiện:

Lý thuyết:

- Tìm hiểu ASP.NET Core, ASP.NET Core MVC.
- Tìm hiểu Entity Framework Core (EF Core)
- Tìm hiểu IdentityUser trong quản lý người dùng.

Thực hành: Áp dụng xây dựng Website bán hàng sử dụng .NET Core.

- Xây dựng giao diện dựa trên Razor View Engine của ASP.NET Core.
- Áp dụng IdentityUser trong quản lý người dùng.
- Sử dụng Entity Framework Core trong việc thao tác với cơ sở dữ liệu.
- Áp dụng SignalR để xây dựng web real-time.

Kế hoạch thực hiện

STT	Thời gian	Công việc
1	Tuần 1 (15/03/2019)	Phân tích yêu cầu, nhận diện tác nhân và chức năng trong sơ đồ Usecase
2	Tuần 2 (22/03/2019)	Đặc tả Usecase, lược đồ sequence, thiết kế database

3	Tuần 3 (29/03/2019)	Thiết kế cơ sở dữ liệu, thiết kế luồng xử lý các chức năng, lập danh sách các màn hình
4	Tuần 4 (05/04/2019)	Mô tả chi tiết các màn hình, tìm hiểu design pattern
5	Tuần 5 (12/04/2019)	Chạy giao diện đã thiết kế, nhận góp ý về giao diện. Demo luồng hoạt động
6	Tuần 6 (19/04/2019)	Nghiệp vụ Đăng nhập - Trang chủ
7	Tuần 7 (26/04/2019)	Nghiệp vụ Tài khoản - Sản phẩm (Quản lý)
8	Tuần 8 (04/05/2019)	Merge code và Kiểm thử 4 nghiệp vụ trên
9	Tuần 9 (11/05/2019)	Triển khai thực hiện Trang chủ
10	Tuần 10 (18/05/2019)	Triển khai thực hiện Chi tiết sản phẩm
11	Tuần 11 (27/05/2019)	Triển khai thực hiện Giỏ hàng
12	Tuần 12 (03/06/2019)	Triển khai thực hiện Thanh toán (Mua hàng)
13	Tuần 13 (07/06/2019)	Merge code và Kiểm thử
14	Tuần 14 (14/06/2019)	Triển khai thực hiện Duyệt sản phẩm
15	Tuần 15 (21/06/2019)	Triển khai thực hiện Thông kê (quản lý)
16	Tuần 16 (28/06/2019)	Triển khai thực hiện SignalR để cập nhật số lượng sản phẩm
17	Tuần 17 (05/07/2019)	Merge code và Kiểm thử Tiến hành viết báo cáo
18	Tuần 18 (10/07/2019)	Hoàn thành báo cáo và hoàn chỉnh phần mềm

Ngày ... tháng ... năm 2019

Ý kiến của Giáo viên hướng dẫn

(Ký và ghi rõ họ tên)

Người viết đề cương

MỤC LỤC

LỜI CẢM ƠN.....	1
ĐỀ CƯƠNG LUẬN VĂN TỐT NGHIỆP	2
DANH MỤC CÁC BẢNG	1
DANH MỤC CÁC HÌNH	3
PHẦN MỞ ĐẦU	5
1.TÍNH CẤP THIẾT CỦA ĐỀ TÀI	5
2.MỤC TIÊU CỦA ĐỀ TÀI	5
3.CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU	5
3.1. Đối tượng nghiên cứu.....	5
3.2. Phạm vi nghiên cứu.....	6
3.3. Phương pháp nghiên cứu	6
4.PHÂN TÍCH CÁC ỨNG DỤNG CÓ LIÊN QUAN.....	6
5.KẾT QUẢ DỰ KIÉN ĐẠT ĐƯỢC.....	6
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	7
1.1.ASP.NET CORE.....	7
1.1.1. Hoàn cảnh ra đời của ASP.NET Core	7
1.1.2. Xây dựng web UI và web API sửng dụng ASP.NET Core MVC.....	8
1.1.3. Đặc điểm riêng biệt	8
1.1.4. .NET Framework , .NET Core, Mono	10
1.2.ENTITY FRAMEWORK CORE	15
1.2.1. Expression, IMetadata (Metadata): Linh hồn của Entity Framework.....	15
1.2.2. Entity Framework Core.....	17
1.2.3. Nhược điểm	18
1.2.4. Ưu điểm.....	19
1.3.ASP.NET MVC	19
1.3.1. Tổng quan.....	19
1.3.2. Ưu điểm.....	21
1.3.3. Tính năng	21
1.4.RAZOR VIEW ENGINE	22
1.5.DEPENDENCY INJECTION	23

1.5.1.	SOLID – Nguyên lý SOLID.....	23
1.5.2.	Dependency Injection	25
1.6.	REPOSITORY & UNIT OF WORK PATTERNNS.....	27
1.6.1.	Tổng quan.....	27
1.6.2.	DbContext của Entity Framework Core	28
1.6.3.	Repository	28
1.6.4.	Unit of Work.....	29
1.7.	ASP.NET IDENTITY	30
1.7.1.	ASP.NET Membership	30
1.7.2.	ASP.NET Simple Membership	30
1.7.3.	ASP.NET Identity.....	30
1.8.	SIGNALR.....	30
1.8.1.	SignalR là gì?	30
1.8.2.	Connections và Hubs	32
1.8.3.	Chọn một chế độ giao tiếp	32
	CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU	33
2.1.	THAM KHẢO CÁC WEBSITE LIÊN QUAN	33
2.1.1.	Shopee.....	33
2.1.2.	Chợ tốt.....	34
2.1.3.	Tiki.....	35
2.1.4.	Lazada	36
2.1.5.	Amazon	37
2.2.	PHÂN TÍCH HIỆN TRẠNG	38
2.3.	MỤC TIÊU	38
2.4.	PHÂN TÍCH YÊU CẦU	39
2.4.1.	Yêu cầu chức năng.....	39
2.4.2.	Yêu cầu phi chức năng.....	40
2.5.	MÔ HÌNH HÓA YÊU CẦU.....	41
2.5.1.	Nhận diện tác nhân và chức năng trong sơ đồ use case	41
2.5.2.	Sơ đồ Usecase.....	42
2.5.3.	Đặc tả Usecase.....	46

CHƯƠNG 3: THIẾT KẾ PHẦN MỀM	60
3.1.THIẾT KẾ HỆ THỐNG.....	60
3.1.1. Lược đồ lớp	60
3.1.2. Lược đồ Sequence	61
3.2.THIẾT KẾ CƠ SỞ DỮ LIỆU.....	66
3.2.1. Mô hình cơ sở dữ liệu	66
3.2.2. Mô tả cơ sở dữ liệu	66
3.3.THIẾT KẾ GIAO DIỆN.....	73
3.3.1. Danh sách màn hình và sơ đồ chuyển đổi.....	74
3.3.2. Mô tả chi tiết các màn hình	74
CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ.....	91
4.1.CÀI ĐẶT	91
4.1.1. ASP.Net Core	91
4.1.2. Entity Framework Core.....	92
4.1.3. Repository	94
4.1.4. Unit Of Work.....	95
4.1.5. Dependency Injection	96
4.2.KIỂM THỬ.....	96
4.2.1. Kế hoạch kiểm thử	96
4.2.2. Quy trình kiểm thử.....	97
PHẦN KẾT LUẬN.....	99
<u>1. KẾT QUẢ ĐẠT ĐƯỢC</u>	99
<u>2. ƯU ĐIỂM</u>	99
<u>3. NHƯỢC ĐIỂM</u>	99
<u>4. HƯỚNG PHÁT TRIỂN</u>	99
TÀI LIỆU THAM KHẢO	101

DANH MỤC CÁC BẢNG

Bảng 2. 1: Bảng yêu cầu phi chức năng	40
Bảng 2. 2: Nhận diện tác nhân và chức năng trong sơ đồ use case.....	41
Bảng 2. 3: Mô tả Usecase Đăng nhập	46
Bảng 2. 4: Mô tả Usecase Thêm sản phẩm.....	47
Bảng 2. 5: Mô tả Usecase Chính sửa sản phẩm.....	47
Bảng 2. 6: Mô tả Usecase Xóa sản phẩm	48
Bảng 2. 7: Mô tả Usecase Thêm sản phẩm vào giỏ hàng	49
Bảng 2. 8: Mô tả Usecase Chính sửa giỏ hàng	50
Bảng 2. 9: Mô tả Usecase Xóa sản phẩm trong giỏ hàng.....	50
Bảng 2. 10: Mô tả Usecase Mua hàng.....	51
Bảng 2. 11: Mô tả Usecase Xem thông tin cá nhân.....	52
Bảng 2. 12: Mô tả Usecase Chính sửa thông tin cá nhân.....	52
Bảng 2. 13: Mô tả Usecase Đánh giá, bình luận các sản phẩm	53
Bảng 2. 14: Mô tả Usecase Tìm kiếm sản phẩm	53
Bảng 2. 15: Mô tả Usecase Thêm người dùng	54
Bảng 2. 16: Mô tả Usecase Xóa người dùng.....	55
Bảng 2. 17: Mô tả Usecase Tìm kiếm người dùng.....	56
Bảng 2. 18: Mô tả Usecase Thêm danh mục	56
Bảng 2. 19: Mô tả Usecase Chính sửa danh mục.....	57
Bảng 2. 20: Mô tả Usecase Xóa danh mục	58
Bảng 2. 21: Mô tả Usecase Thông kê	59
Bảng 3. 1: Chi tiết bảng AspNetUsers	66
Bảng 3. 2: Chi tiết bảng AspNetRoles.....	67
Bảng 3. 3: Chi tiết bảng AspNetUserRoles	67
Bảng 3. 4: Chi tiết bảng Cart	68
Bảng 3. 5: Chi tiết bảng CartDetail.....	68
Bảng 3. 6: Chi tiết bảng Catalog.....	68
Bảng 3. 7: Chi tiết bảng Comment	69
Bảng 3. 8: Chi tiết bảng Image.....	69
Bảng 3. 9: Chi tiết bảng Order.....	70

<i>Bảng 3. 10: Chi tiết bảng OrderDetail</i>	70
<i>Bảng 3. 11: Chi tiết bảng Product.....</i>	71
<i>Bảng 3. 12: Chi tiết bảng ProductDetail</i>	71
<i>Bảng 3. 13: Chi tiết bảng Rate</i>	72
<i>Bảng 3. 14: Chi tiết bảng Answer.....</i>	73
<i>Bảng 3. 15: Các đối tượng màn hình Trang chủ.....</i>	75
<i>Bảng 3. 16: Các đối tượng màn hình Đăng ký.....</i>	76
<i>Bảng 3. 17: Các đối tượng màn hình Đăng nhập.....</i>	77
<i>Bảng 3. 18: Các đối tượng màn hình Thông tin tài khoản.....</i>	78
<i>Bảng 3. 19: Các đối tượng màn hình Chi tiết sản phẩm.....</i>	79
<i>Bảng 3. 20: Các đối tượng màn hình Quản lý sản phẩm.....</i>	81
<i>Bảng 3. 21: Các đối tượng màn hình Thêm sản phẩm.....</i>	83
<i>Bảng 3. 22: Các đối tượng màn hình Quản lý người dùng.....</i>	84
<i>Bảng 3. 23: Các đối tượng màn hình Thêm người dùng.....</i>	85
<i>Bảng 3. 24: Các đối tượng màn hình Quản lý danh mục.....</i>	Error! Bookmark not defined.
<i>Bảng 3. 25: Các đối tượng màn hình Thêm danh mục</i>	Error! Bookmark not defined.
<i>Bảng 3. 26: Các đối tượng màn hình Giỏ hàng</i>	86
<i>Bảng 3. 27: Các đối tượng màn hình Địa chỉ giao hàng</i>	87
<i>Bảng 3. 28: Các đối tượng màn hình Hoàn tất đơn đặt hàng</i>	88
<i>Bảng 3. 29: Các đối tượng màn hình Thống kê.....</i>	90
<i>Bảng 4. 1: Cài đặt ASP.NET Core.....</i>	91
<i>Bảng 4. 2: Cú pháp Migration.....</i>	94

DANH MỤC CÁC HÌNH

<i>Hình 2. 1: Giao diện Shopee</i>	<i>33</i>
<i>Hình 2. 2: Giao diện Chợ Tốt.....</i>	<i>34</i>
<i>Hình 2. 3: Giao diện Tiki</i>	<i>35</i>
<i>Hình 2. 4: Giao diện Lazada</i>	<i>36</i>
<i>Hình 2. 5: Giao diện Amazon.....</i>	<i>37</i>
<i>Hình 2. 6: Usecase Người dùng.....</i>	<i>43</i>
<i>Hình 2. 7: Usecase Quản trị viên</i>	<i>43</i>
<i>Hình 2. 8: Usecase Quản lý giỏ hàng</i>	<i>44</i>
<i>Hình 2. 9: Usecase Quản lý sản phẩm.....</i>	<i>44</i>
<i>Hình 2. 10: Usecase Quản lý người dùng</i>	<i>45</i>
<i>Hình 2. 11: Usecase Thông kê.....</i>	<i>45</i>
<i>Hình 2. 12: Usecase Quản lý danh mục.....</i>	<i>46</i>
<i>Hình 3. 1: Lược đồ lớp.....</i>	<i>60</i>
<i>Hình 3. 2: Sequence Đăng nhập.....</i>	<i>61</i>
<i>Hình 3. 3: Sequence Đăng ký</i>	<i>61</i>
<i>Hình 3. 4: Sequence Thêm người dùng</i>	<i>62</i>
<i>Hình 3. 5: Sequence Thêm sản phẩm.....</i>	<i>62</i>
<i>Hình 3. 6: Sequence Mua hàng</i>	<i>63</i>
<i>Hình 3. 7: Sequence Tìm kiếm sản phẩm</i>	<i>63</i>
<i>Hình 3. 8: Sequence Đánh giá sản phẩm.....</i>	<i>64</i>
<i>Hình 3. 9: Sequence Thêm sản phẩm vào giỏ hàng.....</i>	<i>64</i>
<i>Hình 3. 10: Sequence Thêm danh mục mới.....</i>	<i>65</i>
<i>Hình 3. 11: Sequence Tìm kiếm người dùng</i>	<i>65</i>
<i>Hình 3. 12: Mô hình cơ sở dữ liệu.....</i>	<i>66</i>
<i>Hình 3. 13: Danh sách màn hình và sơ đồ chuyển đổi</i>	<i>74</i>
<i>Hình 3. 14: Màn hình trang chủ.....</i>	<i>75</i>
<i>Hình 3. 15: Màn hình trang chủ.....</i>	<i>75</i>
<i>Hình 3. 16: Màn hình Đăng ký.....</i>	<i>76</i>
<i>Hình 3. 17: Màn hình Đăng nhập.....</i>	<i>77</i>
<i>Hình 3. 18: Màn hình Thông tin tài khoản.....</i>	<i>78</i>

<i>Hình 3. 19: Màn hình Chi tiết sản phẩm.....</i>	79
<i>Hình 3. 20: Màn hình Chi tiết sản phẩm.....</i>	79
<i>Hình 3. 21: Màn hình Quản lý sản phẩm.....</i>	80
<i>Hình 3. 22: Màn hình Thêm sản phẩm.....</i>	82
<i>Hình 3. 23: Màn hình Thêm sản phẩm.....</i>	83
<i>Hình 3. 24: Màn hình Quản lý người dùng</i>	84
<i>Hình 3. 25: Màn hình Thêm người dùng.....</i>	85
<i>Hình 3. 28: Màn hình Giỏ hàng</i>	86
<i>Hình 3. 29: Màn hình Địa chỉ giao hàng.....</i>	87
<i>Hình 3. 30: Màn hình Địa chỉ giao hàng.....</i>	87
<i>Hình 3. 31: Màn hình Hoàn tất đơn đặt hàng</i>	88
<i>Hình 3. 32: Màn hình Thống kê</i>	89
<i>Hình 3. 33: Màn hình Thống kê</i>	89
<i>Hình 4. 1: Khởi tạo .NET Core.....</i>	91
<i>Hình 4. 2: NuGet Packages.....</i>	92
<i>Hình 4. 3: Microsoft.entityframeworkcore.....</i>	93
<i>Hình 4. 4: Cài đặt Entity Framework Core thành công</i>	93
<i>Hình 4. 5: Tạo Migration thành công.....</i>	94
<i>Hình 4. 6: Repository</i>	95
<i>Hình 4. 7: Unit Of Work.....</i>	95
<i>Hình 4. 8: Dependency Injection.....</i>	96
<i>Hình 4. 9: Quy trình kiểm thử.....</i>	97

PHẦN MỞ ĐẦU

1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Trong thời đại phát triển khoa học công nghệ trên toàn thế giới, cuộc sống của mọi người đang được dần cải thiện. Mọi thứ đều có thể trực tuyến “online”, và thứ được mọi người quan tâm nhất chính là mua và bán sản phẩm online. Đây cũng là một hình thức kinh doanh mới của xã hội hiện nay. Hình thức này vừa dễ làm, ít vốn và cực kỳ tiện lợi, đặc biệt là rất phù hợp với những người muốn khởi nghiệp trong kinh doanh.

Để đáp ứng nhu cầu thị trường ở trên, và các thành viên trong nhóm cũng có hứng thú với nội dung này. Nên nhóm quyết định chọn Website bán hàng để làm nội dung triển khai công nghệ .Net Core.

2. MỤC TIÊU CỦA ĐỀ TÀI

- Tìm hiểu .NET Core, MVC, Entity Framework.
- Phân tích, thiết kế và xây dựng website bán hàng với các chức năng cơ bản :
 - o Đăng nhập.
 - o Đăng xuất.
 - o Đăng ký tài khoản.
 - o Xem danh sách sản phẩm.
 - o Xem chi tiết sản phẩm.
 - o Giỏ hàng.
 - o Quản lý sản phẩm.
 - o Quản lý người dùng
 - o Quản lý danh mục.
 - o Tìm kiếm.

3. CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

3.1. Đối tượng nghiên cứu

Đối với đề tài này, đối tượng nhóm nghiên cứu dựa trên nhu cầu mua sắm của con người tại các trang web bán hàng trực tuyến. Đồng thời kèm theo đó là các công nghệ áp dụng tạo ra sản phẩm. Cụ thể như sau:

- Tập trung nghiên cứu công nghệ .NET Core là nền tảng để xây dựng nền hệ thống trang web bán hàng trực tuyến.
- Áp dụng Entity Framework Core làm kết nối đến hệ quản trị cơ sở dữ liệu SQL Server.
- Sử dụng Kiến trúc N-Tier là kiến trúc tổng thể & Mô hình MVC để xây dựng hệ thống.

3.2. Phạm vi nghiên cứu

Đề tài này nhóm chủ yếu đi tập trung vào việc xử lý các nghiệp vụ cơ bản của website bán hàng như: Xem sản phẩm, mua hàng, quản lý sản phẩm, quản lý người dùng,... và tính đúng đắn của dữ liệu trong quá trình sử dụng.

3.3. Phương pháp nghiên cứu

- Phương pháp tổng hợp lý thuyết: Nghiên cứu và tìm hiểu các tài liệu, ứng dụng liên quan đến các công nghệ, mô hình, lý thuyết cần thiết.
- Phương pháp chuyên gia: Tham khảo ý kiến của Thầy hướng dẫn cùng các anh, chị và bạn bè có kinh nghiệm liên quan đến các vấn đề, nghiệp vụ của đề tài để tăng tính chính xác và đúng đắn.
- Phương pháp mô hình hóa: Mô phỏng website từ bước thiết kế cài đặt cho đến kết quả thành phẩm của website.

4. PHÂN TÍCH CÁC ỨNG DỤNG CÓ LIÊN QUAN

- Các website bán hàng phải có giao diện dễ nhìn, bắt mắt, đẹp, dễ sử dụng.
- Các sản phẩm được thể hiện theo danh mục sản phẩm, từ đó khách hàng dễ dàng tìm kiếm sản phẩm mình cần.
- Đều có giỏ hàng và mua hàng.

5. KẾT QUẢ ĐỰ KIẾN ĐẠT ĐƯỢC

- Xây dựng website bán hàng đơn giản có thể hiển thị sản phẩm, dễ dàng sử dụng, và quản lý được sản phẩm, người dùng.
- Thông kê được các dữ liệu, doanh số.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. ASP.NET CORE

ASP.NET Core là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.¹

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework.

Nó được thiết kế để cung cấp và tối ưu development framework cho những ứng dụng cái mà được triển khai trên đám mây (cloud) hoặc chạy on-promise. Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm dẻo trong việc xây dựng giải pháp của bạn.

Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux. Đồng thời nó đã trở thành một mã nguồn mở.

Đây là một thay đổi rất lớn và theo mình là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.

1.1.1. Hoàn cảnh ra đời của ASP.NET Core

Bản phát hành đầu tiên của ASP.NET đã xuất hiện cách đây 15 năm trước, nó là một phần của .NET Framework. Từ đó, hàng triệu lập trình viên đã sử dụng nó để xây dựng những ứng dụng web tuyệt vời, và trên những năm đó Microsoft đã phát triển thêm nhiều tính năng mới.

ASP.NET Core có một số thay đổi kiến trúc lớn, đó là kết quả của việc học hỏi rất nhiều từ các framework module hóa khác. ASP.NET Core không còn dựa trên System.Web.dll. Nó được dựa trên một tập hợp các gói, các module hay cũng được gọi là các Nuget packages. Điều này cho phép bạn tối ưu ứng dụng của bạn để chỉ bao gồm

¹ : [Asp.net Core những thay đổi lớn cho lập trình viên .NET](#)

những packages nào cần thiết. Lợi ích của nó là giúp cho ứng dụng nhỏ hơn, bảo mật chặt chẽ hơn, giảm sự phức tạp, tối ưu hiệu suất hoạt động và giảm chi phí, thời gian cho việc phát triển.

Với ASP.NET Core bạn đạt được những nền tảng cải tiến dưới đây:

- Hợp nhất việc xây dựng web UI và web APIs.
- Tích hợp những client-side frameworks hiện đại và những luồng phát triển.
- Hệ thống cấu hình dựa trên môi trường đám mây thật sự.
- Dependency Injection (DI) được xây dựng sẵn.
- HTTP request được tối ưu nhẹ hơn.
- Có thể host trên IIS hoặc self-host trong process của riêng bạn.
- Được xây dựng trên .NET Core, hỗ trợ thực sự app versioning.
- Chuyển các thực thể, thành phần, module như những NuGet packages.
- Những công cụ mới để đơn giản hóa quá trình phát triển web hiện đại.
- Xây dựng và chạy đa nền tảng(Windows, Mac và Linux).
- Mã nguồn mở và tập trung vào cộng đồng.

1.1.2. Xây dựng web UI và web API sửng dụng ASP.NET Core MVC

- Bạn có thể tạo ra các ứng dụng web có thể testing theo mô hình MVC(Model-View-Controller)
- Bạn có thể xây dựng HTTP services cái mà hỗ trợ nhiều định dạng và có đầy đủ hỗ trợ cho nội dung của dữ liệu trả về
- Razor cung cấp một ngôn ngữ hiệu quả để tạo Views
- Tag Helper cho phép code server side tham gia vào việc tạo và render phần tử HTML
- Model Binding tự động ánh xạ dữ liệu từ HTTP request tới tham số của method action
- Model Validation tự động thực hiện validate client và server

1.1.3. Đặc điểm riêng biệt

1.1.3.1. Hỗ trợ đa nền tảng

Ngày nay bạn có thể phát triển và chạy ASP.NET trên cả Windows, Mac, và Linux. Và nếu trên Windows bạn có thể sử dụng công cụ tốt nhất Visual Studio 2015 để

tạo, quản lý và gỡ lỗi các ứng dụng ASP.NET Core của bạn, thì nay trên bất kỳ nền tảng nào bạn có thể sử dụng Visual Studio Code. Visual Studio Code là một trình soạn thảo với các plugin có hỗ trợ để chỉnh sửa các ứng dụng ASP.NET Core của bạn.²

1.1.3.2. Nguồn mở

Ngày nay mã nguồn và tài liệu ASP.NET đã được Microsoft mở tất cả. Các mã nguồn giờ đã có sẵn trong Github bạn giờ có thể tải hay thay đổi bất kỳ mã nào mà bạn thích. Nếu bạn có bất kỳ một góp ý cải tiến gì đó, bạn có thể gửi một yêu cầu đến cho microsoft để xem xét và kết hợp.

Tương tự như vậy, tất cả các tài liệu cũng là mã nguồn mở và có sẵn trong bài viết docs.asp.net. Mỗi trang trên đó điều có chức năng “chỉnh sửa trang này” ở phía trên và bạn có thể chỉnh sửa các tài liệu từ Microsoft.³

1.1.3.3. Hỗ trợ đầy đủ cho framework

Một thông tin hữu ích là asp.net core đã trở thành mã nguồn mở nhưng Microsoft vẫn sẽ cung cấp hỗ trợ trong 3 năm cho mỗi bản phát hành lớn nhỏ của họ.

1.1.3.4. Hiệu suất

Microsoft giới thiệu máy chủ mới web Kestrel chạy trong host IIS của bạn hoặc chạy sau một host process khác. Kestrel hiện tại là máy chủ .NET chạy nhanh nhất hiện nay.

1.1.3.5. Hỗ trợ xây dựng bằng Dependency Injection

ASP.NET Core đã được xây dựng trong Dependency Injection. Dependency Injection là một mẫu thiết kế cho phép các phụ thuộc của một class được injected như các đối tượng được yêu cầu trong ứng dụng của bạn. Với ASP.NET Core, Microsoft đã cung cấp một Dependency Injection mà bạn có thể sử dụng để xác định sự phụ thuộc được đưa vào Controller, View của bạn, hoặc bất kỳ lớp học khác mà framework sẽ tạo

² : [Asp.net core là gì? Giới thiệu ASP.NET Core?](#)

³ : [About .NET Core, Microsoft](#)

ra cho bạn. Bạn có thể bắt đầu việc cấu hình thông qua phương thức ConfigureServices trong tập tin Startup.cs.

1.1.3.6. Một Framework duy nhất

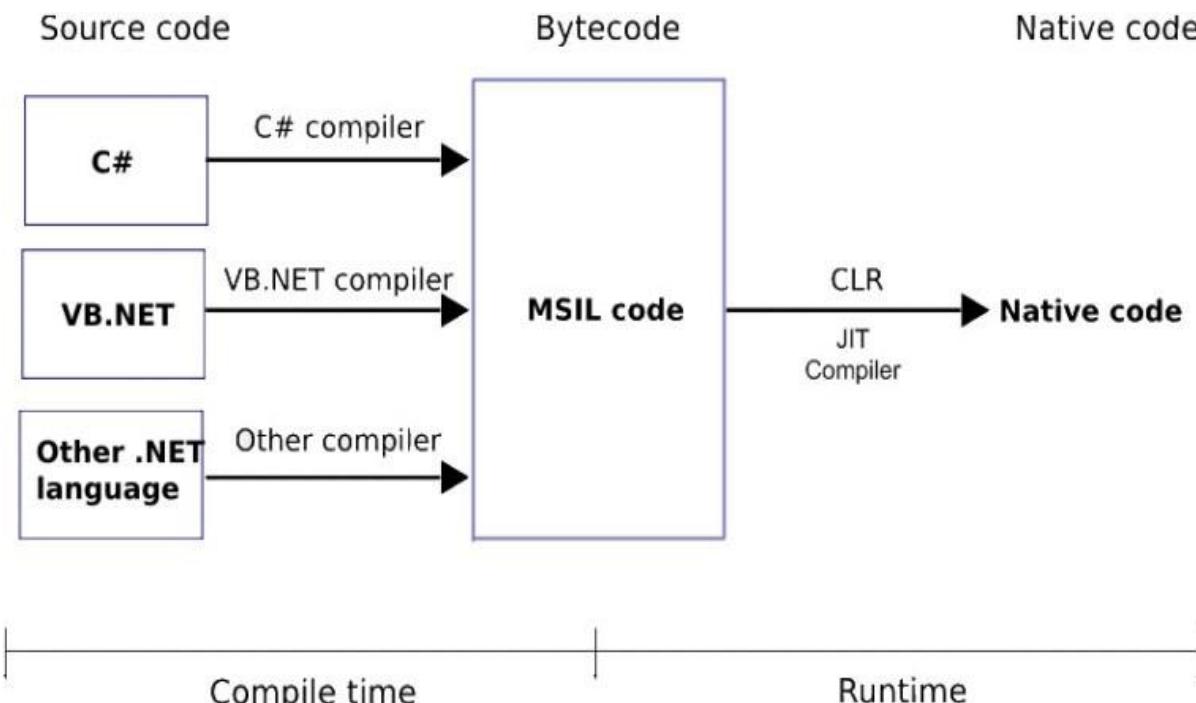
Trong ASP.NET Core, Microsoft đã đưa tất cả các framework vào một framework duy nhất vừa nhẹ hơn và vừa có những tính năng của MVC và WebAPI. Với việc sáp nhập này của MVC và Web API, mọi thứ đơn giản hơn nhiều khi bạn không cần phải cân nhắc định tuyến khác nhau, an toàn, hoặc các bộ lọc cho một ApiController so với MVC Controller. Tất cả các lớp Controller hiện giờ có thể xử lý các yêu cầu sử dụng API Web hoặc cách tiếp cận MVC.

1.1.3.7. MVC Helpers Tag

Với ASP.NET Core, Microsoft đã giới thiệu tag helpers để tạo ra mã phía client từ .NET và làm cho nó dễ dàng hơn để tái sử dụng trong Razor markup. Nó được tham chiếu trong đánh dấu phía máy chủ của bạn như thẻ họ là một tag HTML mà bạn được sử dụng. Công cụ Razor sẽ nhận ra thẻ và thực thi các mã .NET có liên quan tương ứng với nó.

1.1.4. .NET Framework , .NET Core, Mono

1.1.4.1. Môi trường hoạt động

**Hình 1. 1: Quy trình biên soạn code trong .NET⁴**

Trong hình trên, phần ‘Compile time’ là quá trình ‘Build’, và ‘Runtime’ là quá trình chạy (tức là tính từ khi bạn khởi động ứng dụng của mình).

Về cơ bản, khi bạn thực hiện lệnh build (trong Visual Studio, hay ‘dotnet build’ bằng dòng lệnh) thì source code của bạn được chuyển hóa thành một dạng ngôn ngữ trung gian có tên là MSIL (Microsoft Intermediate Language). Khi ứng dụng được khởi chạy, thành phần Runtime-hay tên gọi riêng biệt trong .NET là CLR(common language runtime) sẽ tiến hành dịch mã MSIL thành mã máy (Native code) để cho máy tính có thể thực thi. Quá trình này gọi là JIT (just-in-time) compilation. Cách biên soạn và vận hành của .NET khá giống với Java.

⁴ : [\[Giới thiệu .NET core và ASP.NET core\]Phần 2:Phân biệt .NET Framework, .NET Core và Mono](#)

Để hiểu rõ hơn về JIT, chúng ta hãy thử làm một thí nghiệm. Copy đoạn code sau vào file Program.cs trong project của bạn:

```
namespace ConsoleApplication {
    public class Program {
        public static void DoSomeCalculation() {
            string input = "0";
            for (int i = 0; i < 10; i++) {
                var converted = Convert.ToInt32(input);
                converted++;
                input = converted.ToString();
            }
        }
        public static void Main(string[] args) {
            Stopwatch sw = new Stopwatch();
            //first time
            sw.Start();
            DoSomeCalculation();
            sw.Stop();
            Console.WriteLine($"First calculation: {sw.ElapsedTicks} ticks");
            //second time
            sw.Restart();
            DoSomeCalculation();
            sw.Stop();
            Console.WriteLine($"Second calculation: {sw.ElapsedTicks} ticks");
        }
    }
}
```

Chúng ta sẽ tiến hành gọi cùng 1 hàm 2 lần và đo xem thời gian chạy của 2 lần gọi là bao nhiêu. (10000 ticks = 1 ms). Tiến hành chạy ‘dotnet run’ và xem kết quả.

```

goaty@goaty:~/projects/mydotnetcore$ dotnet run
Project mydotnetcore (.NETCoreApp, Version=v1.0) will be compiled because inputs
were modified
Compiling mydotnetcore for .NETCoreApp, Version=v1.0

Compilation succeeded.
  0 Warning(s)
  0 Error(s)

Time elapsed 00:00:01.2887865

First calculation: 977753 ticks
Second calculation: 9148 ticks
goaty@goaty:~/projects/mydotnetcore$ █

```

Hình 1. 2: Kết quả khi chạy ‘dotnet run’⁵

Bạn có thể thấy rằng lần chạy đầu tiên lâu hơn lần thứ 2 rất nhiều. Đó là do khi DoSomeCalculation() được gọi lần đầu, CLR tiến hành biên soạn (JIT) hàm này thành ngôn ngữ máy. Ở những lần chạy kế tiếp, DoSomeCalculation() không cần ‘JIT’ lại (do đã được ‘JIT’ xong), nên thời gian thực thi nhanh hơn nhiều. (Điều này cũng lý giải tại sao trong một số phần mềm, thời gian thực hiện một chức năng nào đó lần đầu tiên thường chậm hơn so với các lần kế tiếp).

Ngoài việc biên dịch, môi trường hoạt động (Runtime) còn có những công dụng như:

- Tự động quản lý bộ nhớ. Khi làm việc với những ngôn ngữ bậc cao như C# hay Java, bạn không cần giải phóng bộ nhớ bằng cách gọi free() như khi làm việc với C/C++. CLR bao gồm một công cụ dọn rác (Garbage collector -GC) sẽ tự động giải phóng những phần bộ nhớ không được sử dụng.
- Strong typings: CLR quản lý thông tin về các kiểu dữ liệu mà bạn sử dụng. Điều này giúp cho bạn có thể phân biệt được các định dạng thông tin của từng biến khác nhau (class, structure...).

1.1.4.2. Hệ thống thư viện (Libraries) và công cụ (Toolings)

Khi bạn làm việc với .NET, code của bạn sẽ tương tác với rất nhiều các class khác nhau. Ví dụ: Class được sử dụng nhiều nhất trong .NET là System.String. Tất cả những

⁵ : [Giới thiệu .NET core và ASP.NET core, Phần 2](#)

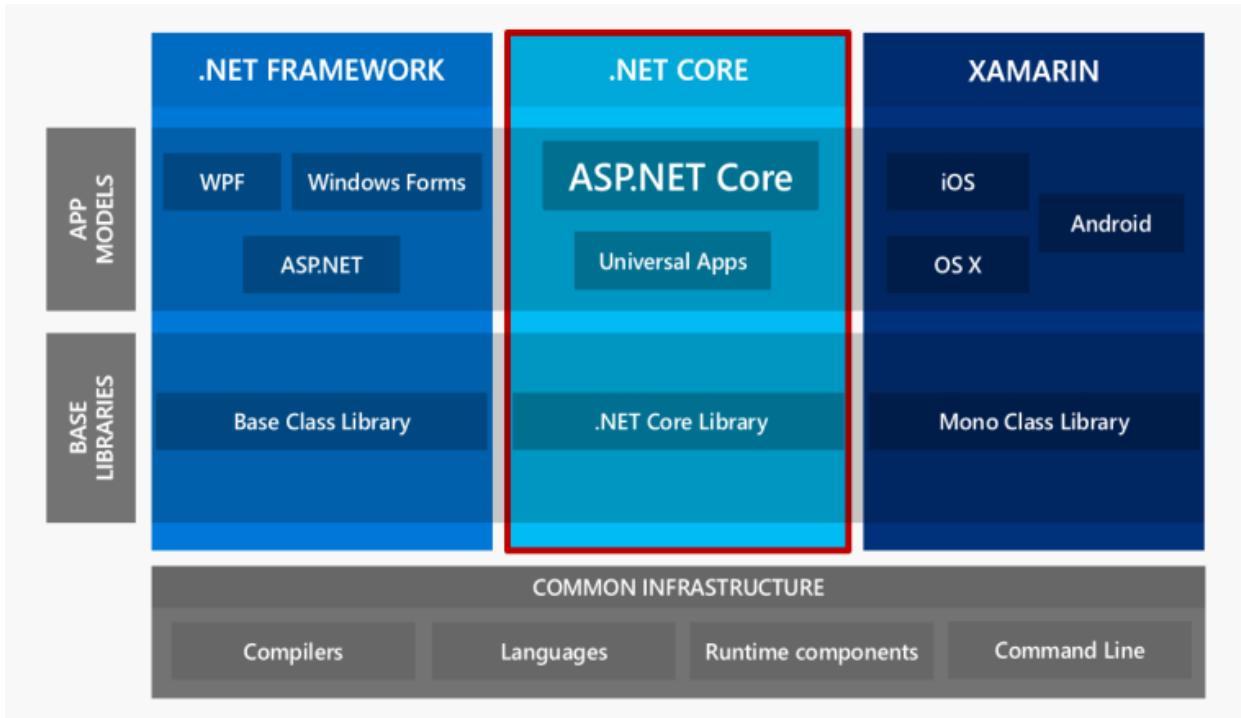
class này được định nghĩa trong hệ thống thư viện cơ bản của .NET mà người ta hay gọi tắt là BCL (Base class libraries).

Các công cụ (toolings) của .NET bao gồm compiler và Visual Studio .NET sử dụng hệ thống build của Microsoft gọi là MSBuild. Đối với nền tảng .NET core mới thì chúng ta còn có thêm công cụ dòng lệnh (dotnet cli).

Phân biệt .NET Framework, .NET Core, và Mono: Tại sao cần phải phân biệt chúng? Vì bạn cần phải hiểu rõ mình đang làm gì. VD: Nếu bạn có ý định chạy một Web server trên Linux thì tuyệt đối không nên sử dụng Mono.

Đối với những người mới làm quen với .NET hay kể cả một số người đã làm việc với .NET lâu năm, những cái tên như .NET Framework, Mono hay gần đây nhất là .NET core vẫn hay gây ra những hiểu nhầm. Tuy nhiên những khái niệm căn bản về .NET nói trên giúp chúng ta phân biệt khá dễ dàng. Về cơ bản, .NET Framework, .NET core và Mono là ba phiên bản .NET khác nhau (có nghĩa là mỗi phiên bản có Runtime, Libraries và Toolings riêng). Vậy tại sao lại có đến 3 phiên bản khác nhau?

- .NET Framework được Microsoft đưa ra chính thức từ năm 2002. .NET Framework chỉ hoạt động trên Windows. Những nền tảng ứng dụng như WPF, Winforms, ASP.NET(1-4) hoạt động dựa trên .NET Framework.
- Mono là phiên bản cộng đồng nhằm mang .NET đến những nền tảng ngoài Windows. Mono được phát triển chủ yếu nhằm xây dựng những ứng dụng với giao diện người dùng và được sử dụng rất rộng rãi: Unity Game, Xamarin...
- Cho đến năm 2013, Microsoft định hướng đi đa nền tảng và phát triển .NET core. .NET core hiện được sử dụng trong các ứng dụng Universal Windows platform và ASP.NET Core.



Hình 1. 3: Phân biệt .NET Framework, .NET Core, Mono⁶

1.2. ENTITY FRAMEWORK CORE

1.2.1. Expression, IModel (Metadata): Linh hồn của Entity Framework.

Trước khi tìm hiểu EF, chúng ta phải hiểu về vai trò và chức năng của Expression và Metadata trong EF.

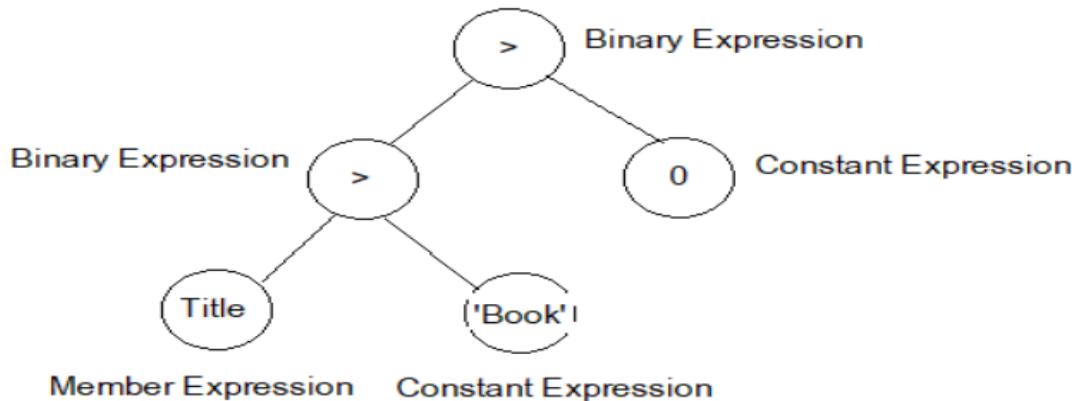
Quay trở lại thời điểm trước khi ORM ra đời, ứng dụng của chúng ta có những class để mô hình hóa các đối tượng và các table để đặc tả dữ liệu lưu trữ trong cơ sở dữ liệu. Khi đó, chúng ta phải viết các phần trung gian (ADO.NET chẳng hạn) để chuyển đổi giữa đối tượng này với nhau. Câu hỏi đặt ra là có cách nào để viết các câu lệnh truy vấn trên các class và sau đó chuyển đổi thành các câu lệnh truy vấn dưới CSDL? Câu hỏi trên chứa đựng hai ý chính:

- Câu truy vấn (Expression)?
- Chuyển đổi Expression thành các câu lệnh SQL?

⁶ : [Giới thiệu .NET core và ASP.NET core, Phần 2](#)

Để hiện thực ý thứ nhất, .NET sử dụng **Expression Tree** (abstract syntax tree, Expression Tree, hay chúng thường gọi là Linq).

Ví dụ: *Product.Where(x => x.Title.CompareTo('Book') > 0)*. Sẽ được phân rã thành Expression Tree như sau:



Hình 1. 4: Cây nhị phân trong Entity Framework⁷

Để có thể chuyển đổi Expression thành các câu truy vấn dưới CSDL, .NET kết hợp Expression với Metadata. Thực chất Metadata là tập hợp những đặc tả, mô tả mối quan hệ giữa các class với các table, giữa các property với column và data type trong C# với data type trong CSDL,... Sở dĩ chúng ta cần Metadata là vì tên bảng và column trong CSDL có thể không trùng với tên class và property trên .NET, và tất nhiên tên các data type không giống nhau giữa CSDL và .NET.

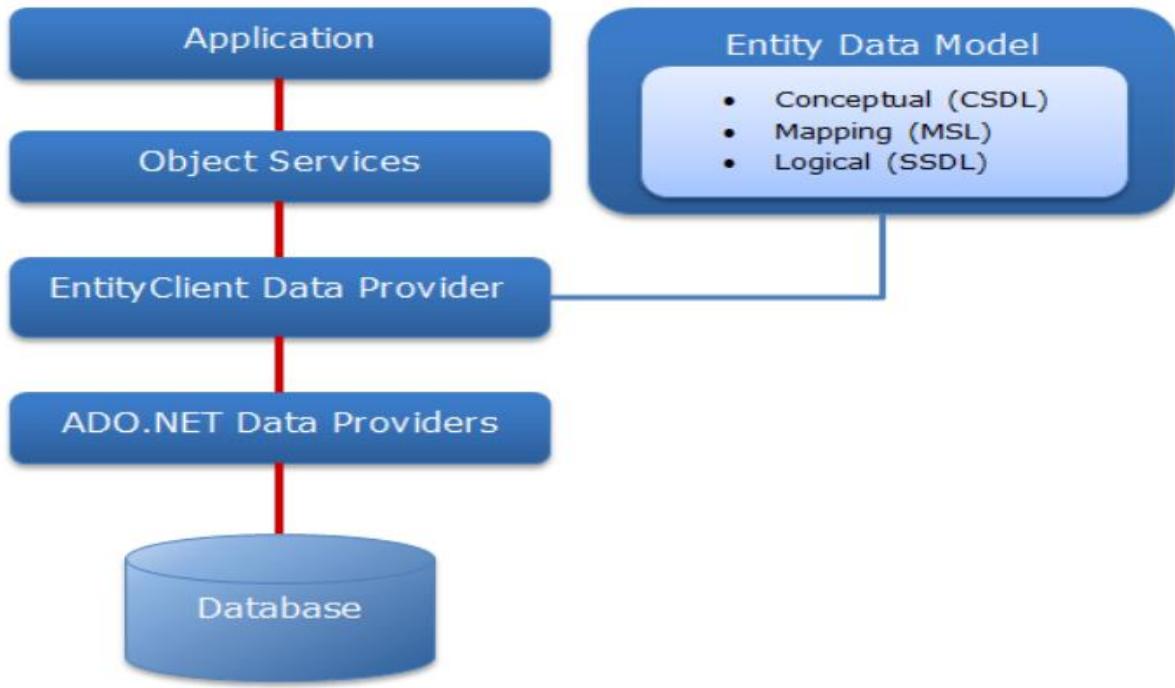
Ngoài ra, thì nó còn giúp cho EF có thể được áp dụng được với nhiều hệ CSDL khác nhau như MySQL, SQL Server,...

Về cơ bản **Metadata(Entity Data Model)** chứa ba thành phần chính:

- CSDL: lưu trữ đặc tả entity class, property.
- MSL: mô tả mối quan hệ giữa các đối tượng trong CSDL với SSDL.
- SSDL: lưu trữ thông tin về database, table, column, datatype của database.

⁷ : [Entity Framework Core 2.0: Những điều bên trong bộ mã nguồn mở EF Core 2.0](#)

- Trong EF6, chúng có thể được gọi là DataSpace, tương ứng với: C-Space, CS-Space và S-Space.



Hình 1. 5: Kiến trúc Entity Framework⁸

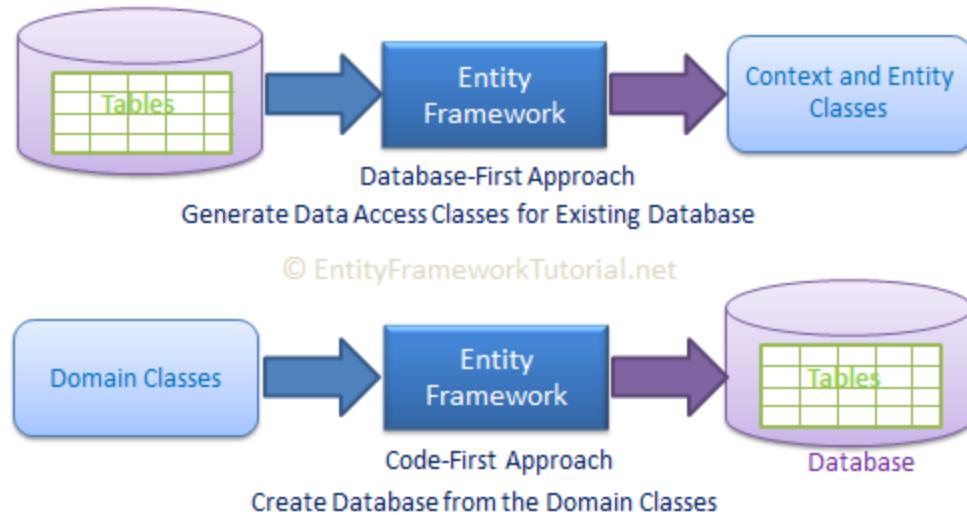
Như vậy, chúng ta có thể thấy Expression và Metadata không thể thiếu trong Entity Framework, nó được coi là linh hồn của Entity Framework. Mục tiêu chính của EF là thực thi các Expression. Và Metadata là gia vị, là thông tin hỗ trợ cần thiết cho quá trình chuyển đổi Expression thành câu lệnh SQL.

1.2.2. Entity Framework Core

Entity Framework Core là phiên bản mới của Entity Framework sau EF 6.x. Nó là mã nguồn mở, nhẹ, có thể mở rộng và là phiên bản đa nền tảng của công nghệ truy cập dữ liệu Entity Framework.

EF Core được sử dụng với các ứng dụng .NET Core. Tuy nhiên, nó cũng có thể được sử dụng với các ứng dụng dựa trên Framework 4.5.

⁸ : [Entity Framework Core 2.0: Những điều bên trong bộ mã nguồn mở EF Core 2.0](#)

**Hình 1. 6: Hai cách tiếp cận của EF Core⁹**

1.2.3. Nhược điểm

EF Core bao gồm toàn bộ những đặc trưng ở EF6. Tuy nhiên, những đặc trưng sau ở EF6 sẽ không được hỗ trợ trên EF Core:

- EDMX/ Graphical Visualization of Model
- Entity Data Model Wizard (for DB-First approach)
- ObjectContext API
- Querying using Entity SQL.
- Automated Migration
- Inheritance: Table per type (TPT)
- Inheritance: Table per concrete class (TPC)
- Many-to-Many without join entity
- Entity Splitting
- Spatial Data
- Lazy loading of related data
- Stored procedure mapping with DbContext for CUD operation
- Seed data
- Automatic migration

⁹ : [Entity Framework Tutorial: Entity Framework Core](#)

1.2.4. Ưu điểm

EF Core được nâng cấp với những thứ sau:

- Easy relationship configuration
- Batch INSERT, UPDATE, and DELETE operations
- In-memory provider for testing
- Support for IoC (Inversion of Control)
- Unique constraints
- Shadow properties
- Alternate keys
- Global query filter
- Field mapping
- DbContext pooling
- Better patterns for handling disconnected entity graphs

1.3. ASP.NET MVC

1.3.1. Tổng quan

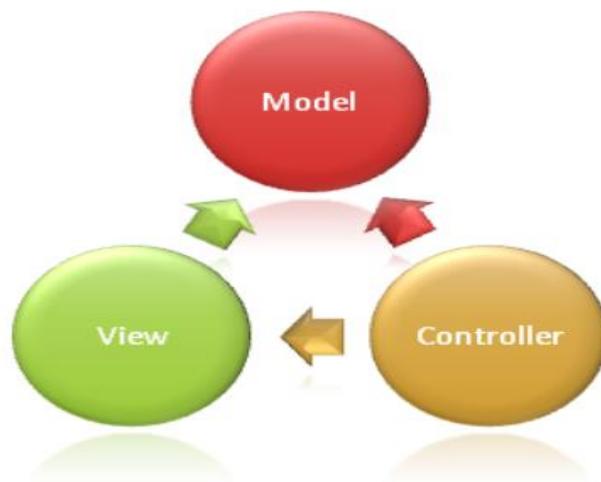
ASP.NET MVC là một framework tuyệt vời hỗ trợ pattern MVC cho ASP.NET. Nếu bạn muốn hiểu ASP.NET MVC làm việc như thế nào, bạn cần phải có một sự hiểu biết rõ ràng về mô hình MVC. MVC là cụm từ viết tắt của Model-View-Controller, nó phân chia pattern của ứng dụng thành 3 phần - model, controller và view.¹⁰

- Model giúp lưu trữ dữ liệu của ứng dụng và trạng thái của nó. Nó là một cơ sở dữ liệu hoặc cũng có thể chỉ là một tập tin XML.
- View được coi là một giao diện người dùng được sử dụng bởi khách truy cập trang web của bạn để nhìn thấy các dữ liệu. Các trang ASPX thường được sử dụng để hiển thị view trong các ứng dụng ASP.NET MVC.

¹⁰ : [ASP .NET MVC - Phần 1: Tao ứng dụng ASP .NET MVC đầu tiên](#)

- Controller chịu trách nhiệm xử lý các tương tác của người dùng với trang web. Nó được sử dụng để xác định loại view nào cần phải được hiển thị. Controller cũng được sử dụng cho mục đích giao tiếp với model.

Framework này là khá nhẹ và cung cấp khả năng kiểm thử, trong đó tích hợp với các tính năng hiện có của ASP.NET như xác thực (authentication) dựa trên membership và cả các master page.



Hình 1. 7: Mô hình Model-View-Controller¹¹

- Models: Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.
- Views: Views là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các checkbox dựa trên trạng thái hiện tại của một đối tượng Product.

¹¹ : [Tổng quan về ASP.NET MVC](#)

- Controllers: Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Ví dụ, controller sẽ quản lý các dữ liệu người dùng gửi lên (query-string values) và gửi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.

MVC giúp bạn tạo được các ứng dụng mà chúng phân tách rạch rời các khía cạnh của ứng dụng (logic về nhập liệu, logic xử lý tác vụ và logic về giao diện). MVC chỉ ra mỗi loại logic kể trên nên được thiếp lập ở đâu trên ứng dụng. Logic giao diện (UI logic) thuộc về views. Logic nhập liệu (input logic) thuộc về controller. Và logic tác vụ (Business logic – là logic xử lý thông tin, mục đích chính của ứng dụng) thuộc về model. Sự phân chia này giúp bạn giảm bớt được sự phức tạp của ứng dụng và chỉ tập trung vào mỗi khía cạnh cần được cài đặt ở mỗi thời điểm. Ví dụ như bạn chỉ cần tập trung vào giao diện (views) mà không phải quan tâm đến logic xử lý thông tin của ứng dụng.

1.3.2. Ưu điểm

- Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller.
- Nó không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
- Nó sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Nhờ đó bạn có thể thiết kế một hạ tầng quản lý định tuyến.
- Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)
- Hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng

1.3.3. Tính năng

- Tách bạch các tác vụ của ứng dụng (logic nhập liệu, business logic, và logic giao diện), dễ dàng kiểm thử và mặc định áp dụng hướng phát triển TDD. Tất cả các tính

năng chính của mô hình MVC được cài đặt dựa trên interface và được kiểm thử bằng cách sử dụng các đối tượng mocks, mock object là các đối tượng mô phỏng các tính năng của những đối tượng thực sự trong ứng dụng. Bạn có thể kiểm thử unit-test cho ứng dụng mà không cần chạy controller trong tiến trình ASP.NET, và điều đó giúp unit test được áp dụng nhanh chóng và tiện dụng. Bạn có thể sử dụng bất kỳ nền tảng unit-testing nào tương thích với nền tảng .NET.

- MVC là một nền tảng khả mở rộng (*extensible*) & khả nhúng (*pluggable*). Các thành phần của ASP.NET MVC được thiết kế để chúng có thể được thay thế một cách dễ dàng hoặc dễ dàng tùy chỉnh. Bạn có thể nhúng thêm view engine, cơ chế định tuyến cho URL, cách kết xuất tham số của action-method và các thành phần khác. ASP.NET MVC cũng hỗ trợ việc sử dụng *Dependency Injection* (DI) và *Inversion of Control* (IoC). DI cho phép bạn gắn các đối tượng vào một lớp cho lớp đó sử dụng thay vì buộc lớp đó phải tự mình khởi tạo các đối tượng. IoC quy định rằng, nếu một đối tượng yêu cầu một đối tượng khác, đối tượng đầu sẽ lấy đối tượng thứ hai từ một nguồn bên ngoài, ví dụ như từ tập tin cấu hình. Và nhờ vậy, việc sử dụng DI và IoC sẽ giúp kiểm thử dễ dàng hơn.
- ASP.NET MVC có thành phần ánh xạ URL mạnh mẽ cho phép bạn xây dựng những ứng dụng có các địa chỉ URL xúc tích và dễ tìm kiếm. Các địa chỉ URL không cần phải có phần mở rộng của tên tập tin và được thiết kế để hỗ trợ các mẫu định dạng tên phù hợp với việc tối ưu hóa tìm kiếm (URL) và phù hợp với lập địa chỉ theo kiểu REST.
- Hỗ trợ các tính năng sẵn có của ASP.NET như cơ chế xác thực người dùng, quản lý thành viên, quyền, output caching và data caching, session và profile, quản lý tình trạng ứng dụng, hệ thống cấu hình...
- ASP.NET MVC 3 còn bổ sung một view engine mới là *Razor View Engine* cho phép thiết lập các view nhanh chóng, dễ dàng và tốn ít công sức hơn so với việc sử dụng Web Forms view engine.

1.4. RAZOR VIEW ENGINE

Theo ScottGu¹², Razor là một view engine:

- Compact, Expressive, and Fluid: Razor hạn chế số lượng kí tự trong một tập tin, việc gõ code sẽ trở nên nhanh chóng, linh hoạt và rõ ràng. Không giống hầu hết các loại khuôn mẫu cú pháp thường thấy, bạn không cần phải chỉ ra một cách rõ ràng đâu là một khối code server trong một văn bản HTML. Bộ phân tích Razor đủ thông minh để suy ra điều này.
- Easy to Learn.
- Is not a new language: Razor chỉ sử dụng cú pháp C#/VB (hoặc các ngôn ngữ khác) kết hợp với HTML.
- Works with any Text Editor.
- Has great Intellisense: Visual Studio 2010 và Visual Web Developer 2010 hỗ trợ đầy đủ tính năng intellisense cho Razor.
- Unit Testable: Razor hỗ trợ khả năng unit test cho các view mà không cần các controller hoặc web-server. Chúng có thể được host bấy kì dự án unit test nào.

1.5. DEPENDENCY INJECTION

1.5.1. SOLID – Nguyên lý SOLID

SOLID là những nguyên lý thiết kế trong OOP. Đây là những nguyên lý được đúc kết bởi máu xương vô số developer, rút ra từ hàng ngàn dự án thành công và thất bại. Một project áp dụng những nguyên lý này sẽ có code dễ đọc, dễ test, rõ ràng hơn. Và việc quan trọng nhất là việc *maintainace* code sẽ dễ hơn rất nhiều(*maintainance*: thêm bớt chức năng và sửa lỗi). Bao gồm 5 nguyên tắc:¹³

1.5.1.1. Single responsibility principle

Nguyên lý đầu tiên, tương ứng với chữ **S** trong SOLID. Nội dung nguyên lý: “*Một class chỉ nên giữ một trách nhiệm duy nhất (Chỉ có thể sửa đổi class với 1 lý do duy nhất)*”

¹² : [ASP.NET MVC – Bảng Razor Syntax Reference \(gồm Razor v2\)](#)

¹³ : [DEPENDENCY INJECTION VÀ INVERSION OF CONTROL – Phần 1,2,3](#)

Để hiểu nguyên lý này, ta hãy lấy ví dụ với 1 class vi phạm nguyên lý. Ta có 1 class như sau:

```
public class ReportManager()
{
    public void ReadDataFromDB();
    public void ProcessData();
    public void PrintReport();
}
```

Class này giữ tới 3 trách nhiệm: Đọc dữ liệu từ DB, xử lý dữ liệu, in kết quả. Do đó, chỉ cần ta thay đổi DB, thay đổi cách xuất kết quả,... ta sẽ phải sửa đổi class này. Càng về sau class sẽ càng phình to ra.

Theo đúng nguyên lý, ta phải tách class này ra làm 3 class riêng. Tuy số lượng class nhiều hơn những việc sửa chữa sẽ đơn giản hơn, class ngắn hơn nên cũng ít bug hơn.

1.5.1.2. *Open/closed principle*

Nguyên lý thứ hai, tương ứng với chữ **O** trong SOLID. Nội dung nguyên lý: “*Có thể thoải mái mở rộng một class, nhưng không được sửa đổi bên trong class đó.*”

Theo nguyên lý này, mỗi khi ta muốn thêm chức năng cho chương trình, chúng ta nên viết class mới mở rộng class cũ (bằng cách kế thừa hoặc sở hữu class cũ) không nên sửa đổi class cũ.

1.5.1.3. *Liskov substitution principle*

Nguyên lý thứ ba, tương ứng với chữ **L** trong SOLID. Nội dung nguyên lý: “*Trong một chương trình, các object của class con có thể thay thế class cha mà không làm thay đổi tính đúng đắn của chương trình.*”

Hãy tưởng tượng bạn có 1 class cha tên Vit. Các class VitBau, VitXiem, có thể kế thừa class này, chương trình chạy bình thường. Tuy nhiên nếu ta viết class VitChayPin, cần pin mới chạy được. Khi class này kế thừa class Vit, vì không có pin không chạy được, sẽ gây lỗi. Đó là 1 trường hợp vi phạm nguyên lý này.

1.5.1.4. *Interface segregation principle*

Nguyên lý thứ tư, tương ứng với chữ **I** trong SOLID. Nội dung nguyên lý: “*Thay vì dùng một interface lớn, ta nên tách thành nhiều interface nhỏ, với nhiều mục đích cụ thể.*”

Hãy tưởng tượng chúng ta có 1 interface lớn, khoảng 100 methods. Việc implements sẽ khá cực khổ, ngoài ra còn có thể dư thừa vì 1 class không cần dùng hết 100 method. Khi tách interface ra thành nhiều interface nhỏ, gồm các method liên quan tới nhau, việc implement và quản lý sẽ dễ hơn.

1.5.1.5. Dependency inversion principle

Nguyên lý cuối cùng, tương ứng với chữ **D** trong SOLID. Nội dung nguyên lý: “*Các module cấp cao không nên phụ thuộc vào các modules cấp thấp. Cả 2 nên phụ thuộc vào abstraction. Interface (abstraction) không nên phụ thuộc vào chi tiết, mà ngược lại.*”

Nguyên lý này khá lắt léo, mình sẽ lấy ví dụ thực tế. Chúng ta đều biết 2 loại đèn: đèn tròn và đèn huỳnh quang. Chúng cùng có đuôi tròn, do đó ta có thể thay thế đèn tròn bằng đèn huỳnh quang cho nhau 1 cách dễ dàng.

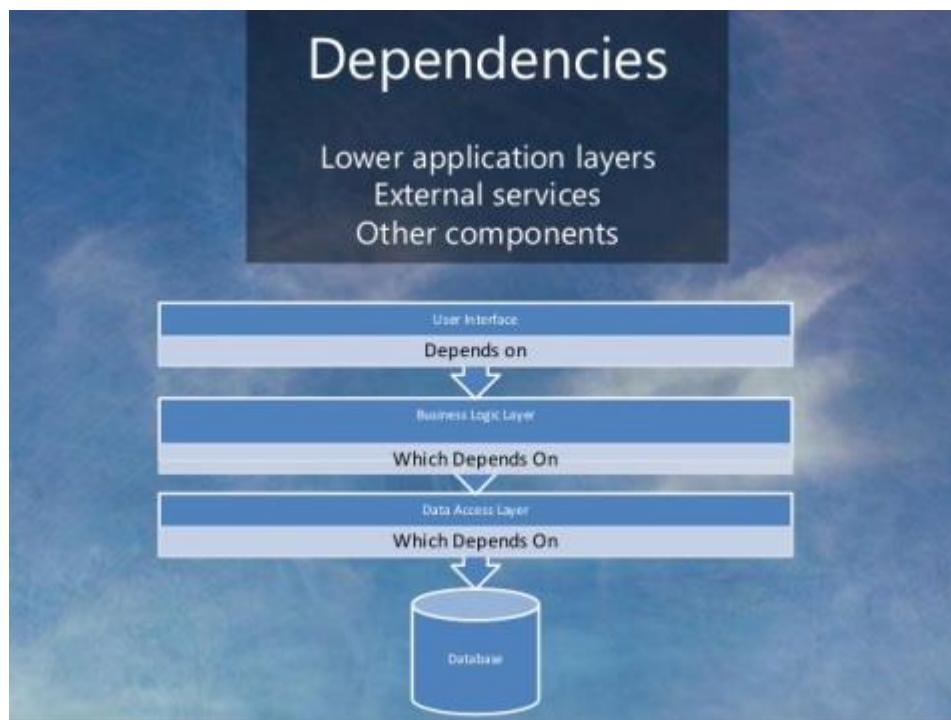
Ở đây, interface chính là đuôi tròn, implementation là bóng đèn tròn và bóng đèn huỳnh quang. Ta có thể hoán đổi dễ dàng giữa 2 loại bóng vì ô điện chỉ quan tâm tới interface (đuôi tròn), không quan tâm tới implementation.

Trong code cũng vậy, khi áp dụng Dependency Inverse, ta chỉ cần quan tâm tới interface. Để kết nối tới database, ta chỉ cần gọi hàm Get, Save,... của Interface IDataAccess. Khi thay database, ta chỉ cần thay implementation của interface này.

1.5.2. Dependency Injection

1.5.2.1. Tổng quan

Dependency là những module cấp thấp, hoặc cái service gọi từ bên ngoài. Với cách code thông thường, các module cấp cao sẽ gọi các module cấp thấp. Module cấp cao sẽ phụ thuộc và module cấp thấp, điều đó tạo ra các dependency.

**Hình 1. 8: Dependency¹⁴**

Dependency Injection có thể hiểu đơn giản là cách thức tổ chức, design source code sao cho có thể "tiêm" các đối tượng dependency vào trong đối tượng mà nó dependent. Hoặc Dependency Injection là một kiểu mẫu lập trình (design pattern) được sử dụng để cố gắng đạt được sự không phụ thuộc giữa các object với nhau khi có mối quan hệ phụ thuộc giữa một object này với một object khác.

Các phương pháp cơ bản để Dependency Injection:

- Constructor Injection: Các dependency sẽ được truyền vào (inject vào) 1 class thông qua constructor của class đó. Đây là cách thông dụng nhất.
- Setter Injection: Các dependency sẽ được truyền vào 1 class thông qua các hàm Setter.
- Public fields: Các dependency sẽ được truyền vào 1 class một cách trực tiếp vào các public field. Cách này ít được sử dụng nhất.

Ưu điểm	Nhược điểm
---------	------------

¹⁴ : [DEPENDENCY INJECTION VÀ INVERSION OF CONTROL – Phần 1,2,3](#)

<ul style="list-style-type: none"> - Giảm sự kết dính giữa các module. - Code dễ bảo trì, dễ thay thế module. - Rất dễ test và viết Unit Test. - Dễ dàng thấy quan hệ giữa các module (Vì các dependency đều được inject vào constructor) 	<ul style="list-style-type: none"> - Khái niệm DI khá khó hiểu, các developer mới sẽ gặp khó khăn khi học. - Sử dụng interface nên đôi khi sẽ khó debug, do không biết chính xác module nào được gọi. - Các object được khởi tạo toàn bộ ngay từ đầu, có thể làm giảm performance. - Làm tăng độ phức tạp của code.
---	---

1.5.2.2. Thời gian sống

Có 3 loại:

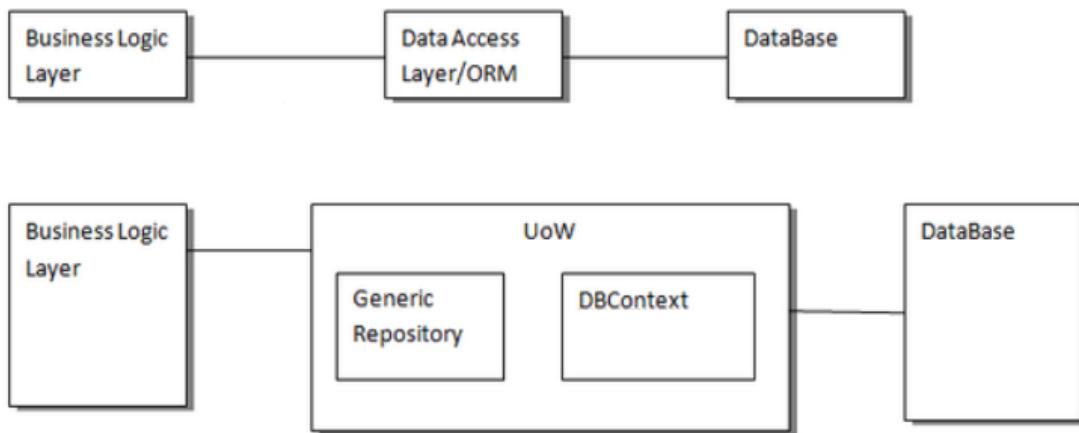
- Transient: mỗi lần gọi tới nó sẽ là 1 biến thể mới hoàn toàn. Cái này chỉ phù hợp cho các loại service siu nhẹ nhàng và ko có lưu trữ trạng thái.
- Scoped: Trong cùng 1 request, nó sẽ chỉ được tạo 1 lần duy nhất.
- Singleton: Nó sẽ được tạo lần đầu tiên khi được gọi, và sẽ sống mãi cho tới khi app bạn chết.

1.6. REPOSITORY & UNIT OF WORK PATTERNS

1.6.1. Tổng quan

Repository là một mẫu dùng để tạo ra một lớp abstraction trung gian giữa lớp data và lớp business. Lớp này chứa đựng phương thức thao tác mà để giao tiếp với lớp data để phục vụ cho business từ lớp logic. Mục đích tạo ra lớp này để cách ly với việc tiếp cận data sao cho những thay đổi không ảnh hưởng trực tiếp đến lớp logic business.

Unit of Work là một kỹ thuật để đảm bảo tất cả các request tới database mà có liên quan tới nhau đều được thực hiện trên cùng một DbContext.

**Hình 1. 9: Repository & Unit of Work¹⁵**

1.6.2. DbContext của Entity Framework Core

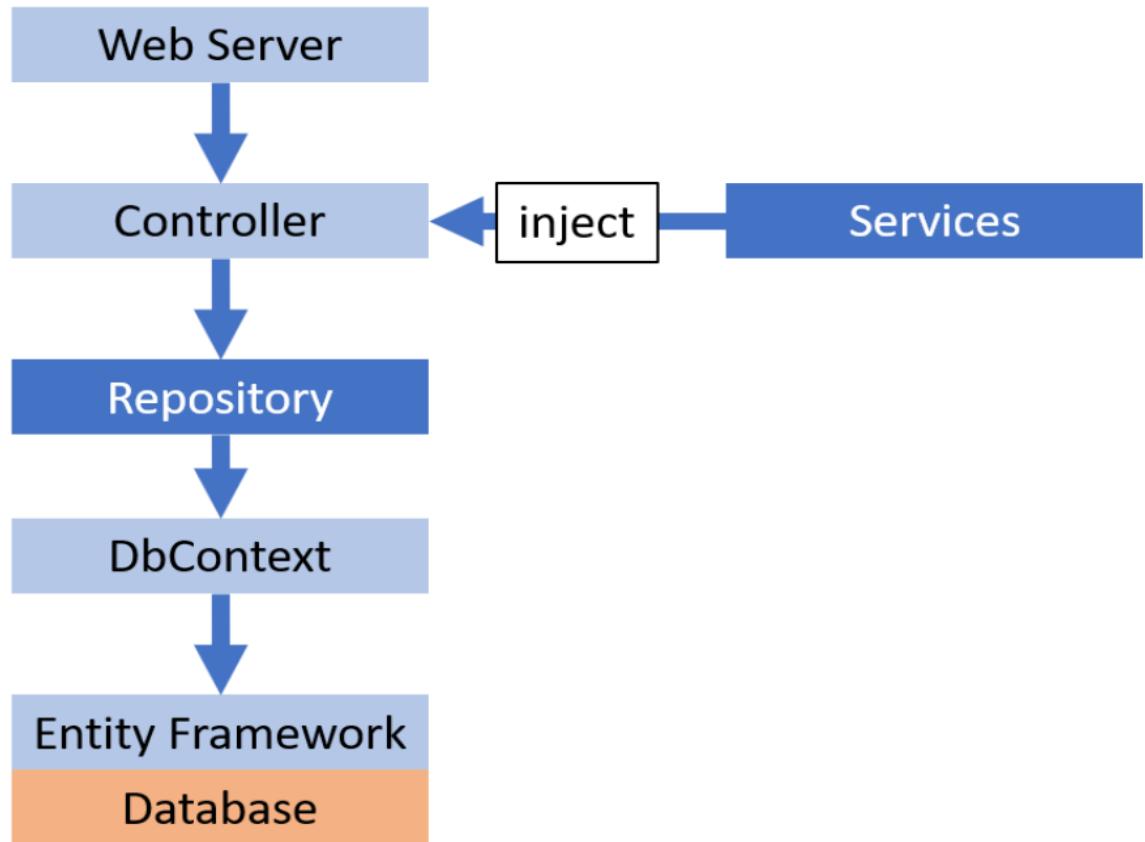
DbContext là một thực thể đại diện cho một phiên làm việc với database, dùng để query và lưu trữ dữ liệu của bạn. Vì nó chỉ đại diện cho 1 phiên làm việc, trong ASP.NET, mỗi khi có 1 request mới từ browser, 1 DbContext mới sẽ được tạo ra, và sẽ bị dispose khi return response cho browser. Thông thường, bạn sẽ kế thừa lại từ DbContext, nhét thêm các DbSet vào, từ đó mới query được.

Để đảm bảo tính toàn vẹn dữ liệu, DbContext dùng 1 cơ chế gọi là tracking. Khi bạn thay đổi 1 record (thêm, xóa, sửa), thì thay đổi đó ko được đưa xuống database ngay, mà sẽ còn vương vấn lại. DbContext sẽ theo dõi sự thay đổi này. Cho tới khi bạn đã thực hiện tất cả các thay đổi cần thiết, rồi gọi yourDbContext.SaveChanges(), thì lúc này, những thay đổi được track sẽ được ship xuống database.

Vậy tất cả những điều này liên quan gì tới Repository và Unit of Work? DbContext là một implement của Repository và Unit of Work, chỉ có điều nó nằm sâu trong framework, còn implement của bạn sẽ nằm ở application.

1.6.3. Repository

¹⁵ : [Repository Pattern và Unit of Work với Entity Framework trong ASP.NET MVC](#)

**Hình 1. 10: Repository¹⁶**

Repository đóng vai trò là 1 lớp trung gian giữa Business Logic Layer (controllers và services) và Data Access Layer (các DbContext).

Lợi ích của Repository:

- Tách việc xử lý logic và việc truy cập database:
 - Dễ trace bug.
 - Dễ unit test.
 - Dễ thay đổi logic hoặc database.
- Gom chung nhiều tác vụ cơ bản về 1 chỗ, không phải viết đi viết lại 1 tác vụ nhiều lần.

1.6.4. Unit of Work

Unit of Work chỉ có 1 nhiệm vụ duy nhất, đảm bảo tất cả các repository của bạn đều dùng chung một DbContext. Bằng cách này, khi thực hiện xong tất cả các tác vụ thay

¹⁶ : [Repository và Unit of Work Pattern](#)

đổi database, bạn chỉ cần gọi DbContext.SaveChanges() 1 lần duy nhất, và các thay đổi đó sẽ được lưu lại trong database.

1.7. ASP.NET IDENTITY

1.7.1. ASP.NET Membership

Vào những năm 2000, nhu cầu rõ rệt về websites phải có đăng nhập, đăng ký thành viên. Do vậy Microsoft đã cho ra đời ASP.NET Membership.

Phiên bản này cực kỳ hạn chế và Database được thiết kế cho SQL Server, và không thể thay đổi. Tuy các provider được thiết kế để có thể thay đổi, nhưng mà hàng loạt hardcoded và tư duy bắt buộc phải dùng SQL server của lập trình viên khiến việc thay đổi này vô cùng cực khổ không xài được OWIN.

1.7.2. ASP.NET Simple Membership

Sang tới 2010, lúc này WebMatrix đang thịnh, Microsoft cũng cho ra liền một bản rút gọn và nâng cấp của Membership, Mục tiêu của Simple Membership là giúp dễ dàng thêm chức năng thành viên vào ứng dụng web. Simple Membership đã giúp dễ dàng tùy chỉnh thông tin hồ sơ người dùng hơn, nhưng nó vẫn chia sẻ các vấn đề khác với ASP.NET Membership và nó có vẫn còn tồn tại một số hạn chế.

1.7.3. ASP.NET Identity

ASP.NET Identity là 1 công nghệ ra đời sau ASP.NET Membership vốn đã khá quen thuộc với các tín đồ của ASP.NET trong bài toán xác thực và phân quyền người dùng trên website. ASP.NET Identity khắc phục một số yếu điểm của ASP.NET Membership và bổ sung thêm nhiều tính năng hấp dẫn để bắt kịp xu thế như hỗ trợ đăng nhập qua mạng xã hội, đăng nhập thông qua 2 bước, gửi mail xác nhận khi đăng ký,...

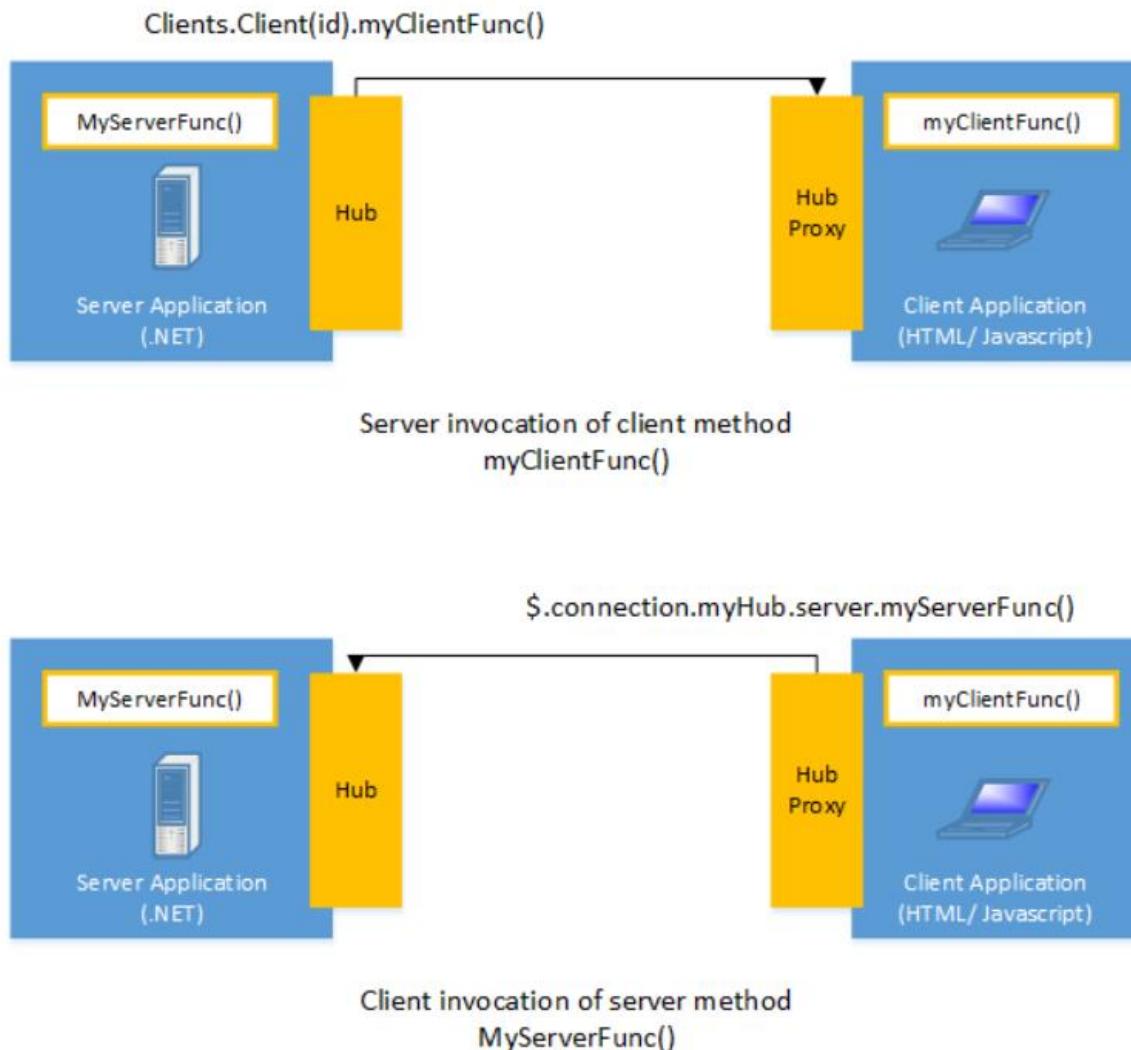
1.8. SIGNALR

1.8.1. SignalR là gì?

ASP.NET SignalR là một thư viện cho các lập trình viên Asp.Net đơn giản hóa quá trình thêm chức năng web real-time trong phát triển ứng dụng. Real-time web functionality là gì? Đó là khả năng server đẩy những nội dung tới client đã được kết nối một cách tức thì. Nó khác với giao thức HTTP thông thường: server đợi những yêu cầu

từ client và trả về nội dung tương ứng. SignalR có thể sử dụng trong bất kì chức năng web real-time nào. Trong đó ứng dụng chat trên web là một ví dụ điển hình.

SignalR cung cấp một API đơn giản cho việc tạo server-to-client remote procedure call (RPC) để gọi những hàm javascript trong trình duyệt (và những nền tảng khác) từ code .Net của server-side. SignalR cũng bao gồm API cho việc quản lý kết nối (connect và disconnect events) và những kết nối nhóm.



Hình 1. 11: Kết nối giữa client và server thông qua Hub

SignalR xử lý quản lý kết nối một cách tự động, và cho bạn truyền đi thông điệp tới tất cả các client đã được kết nối một cách đồng loạt, giống như một chat room. Bạn cũng có thể gửi những thông điệp tới những client được xác định. Kết nối giữa client và server

là liên tục, không giống như kết nối HTTP cổ điển, cái mà sẽ thành lập lại kết nối cho mỗi lần giao tiếp.

1.8.2. Connections và Hubs

SignalR API chứa hai chế độ cho việc giao tiếp giữa client và server: Persistent Connection và Hubs.

Một Hub là một high-level đã xây dựng dựa trên Connection API mà cho phép client và server gọi những methods của nhau một cách trực tiếp. SignalR xử lý việc điều phối qua biên giới máy như ảo thuật, cho phép clients gọi các methods trên server một cách dễ dàng như các methods cục bộ và ngược lại. Việc sử dụng chế độ giao tiếp Hubs sẽ là quen thuộc với lập trình viên mà đã sử dụng APIs triệu gọi từ xa giống như .Net Remoting. Sử dụng Hub cũng cho phép bạn truyền "strongly typed parameters" tới methods, enabling model binding.

Khi code bên server gọi một method trên client, một gói được gửi qua truyền tải chủ động cái mà chứa tên và những tham số của method được gọi (khi một đối tượng được gửi đi như một tham số của method, nó được serialized sử dụng JSON). Tiếp theo Client khớp tên method với những medthods được định nghĩa bên code client. Nếu có một trùng khớp, client method sẽ thực thi việc sử dụng deserialized parameter data.

1.8.3. Chọn một chế độ giao tiếp

Hầu hết các ứng dụng nên dùng Hubs API. Connection API có thể được sử dụng trong các trường hợp sau đây:

- Định dạng của thông điệp thực sự gửi cần được chỉ định.
- Lập trình viên thích làm việc với mô hình messaging và dispatching hơn là mô hình truy cập từ xa.
- Ứng dụng hiện có mà sử dụng mô hình messaging đang được chuyển qua sử dụng SignalR.

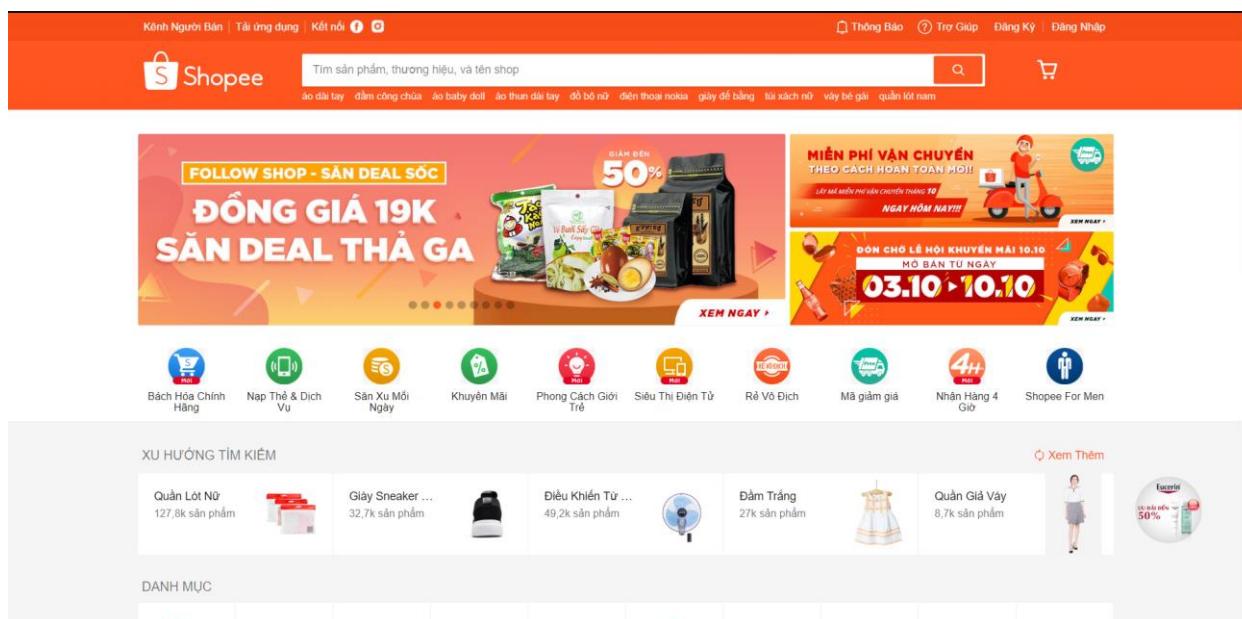
CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU

2.1. THAM KHẢO CÁC WEBSITE LIÊN QUAN

2.1.1. Shopee

Trang chủ: <https://shopee.vn/>:

Website thương mại điện tử đang rất nổi ở thời gian gần đây. Website có rất nhiều sản phẩm đa dạng, đầy đủ thể loại và web cũng cung cấp cho người dùng các chức năng mua bán, giao tiếp, đánh giá, xếp hạng giúp cho chúng ta có thể dễ dàng chọn và mua những thứ cần thiết cho chính bản thân. Đặc biệt là website luôn có sự kiện giảm giá trên các mặt hàng.



Hình 2. 1: Giao diện Shopee

- Giao diện đẹp, dễ nhìn.
- UI/UX dễ sử dụng, phù hợp với cả người không rành Internet.
- View thể hiện đầy đủ các chi tiết quan trọng. (số lượng, giá cả, giảm giá,...)
- Có chức năng theo dõi đơn hàng đã đặt rất hữu dụng.
- Có thể chat với người bán hàng, bình luận & đánh giá.

Kết: Giao diện rất đẹp, dễ thu hút người sử dụng. UI/UX dễ sử dụng.

2.1.2. Chợ tốt

Trang chủ: <https://www.chotot.com/>

Đây là 1 trong những website thương mại điện tử đầu tiên ở Việt Nam. Với mục đích tạo ra để đáp ứng nhu cầu mua-bán giữa người với người, Website này đã nhận được rất nhiều sự ủng hộ từ mọi người. Website cung cấp đầy đủ các chức năng giúp người dùng có thể đăng thông tin sản phẩm cần bán, hoặc tìm kiếm sản phẩm muốn mua. Đặc biệt là giá các sản phẩm ở đây cũng khá rẻ. Tuy nhiên nếu về độ tin cậy thì website này vẫn nằm danh sách đáng lưu ý.



Hình 2. 2: Giao diện Chợ Tốt

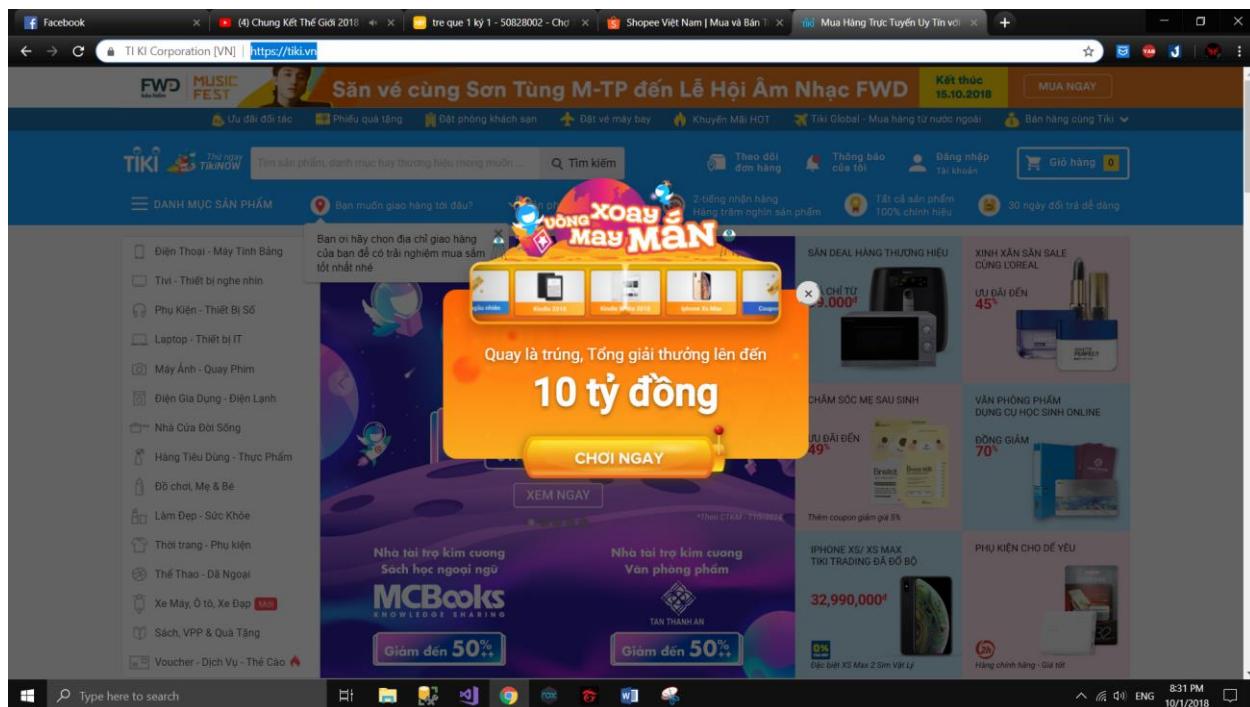
- Giao diện đơn giản.
- UX/UI rất dễ sử dụng.
- Không trợ thanh toán, chỉ là trang hỗ trợ, tìm kiếm giữa người mua với người bán.
- Độ tin cậy không cao.

Kết: Giao diện đẹp, dễ sử dụng. Nhưng không tiện dụng, tin cậy (không thanh toán).

2.1.3. Tiki

Trang chủ: <https://tiki.vn/>

Tiki hiện tại đang là một trong những “ông trùm” của thương mại điện tử ở Việt Nam ngày nay. Ngoài các chức năng cơ bản như mua-bán sản phẩm, website còn có nhiều trò chơi, sự kiện, và ưu đãi rất thú vị. Với một giao diện bắt mắt, UI/UX dễ sử dụng, Tiki rất xứng đáng để có thể lọt vào top những website thương mại điện tử hàng đầu Việt Nam trong ngày này.



Hình 2. 3: Giao diện Tiki

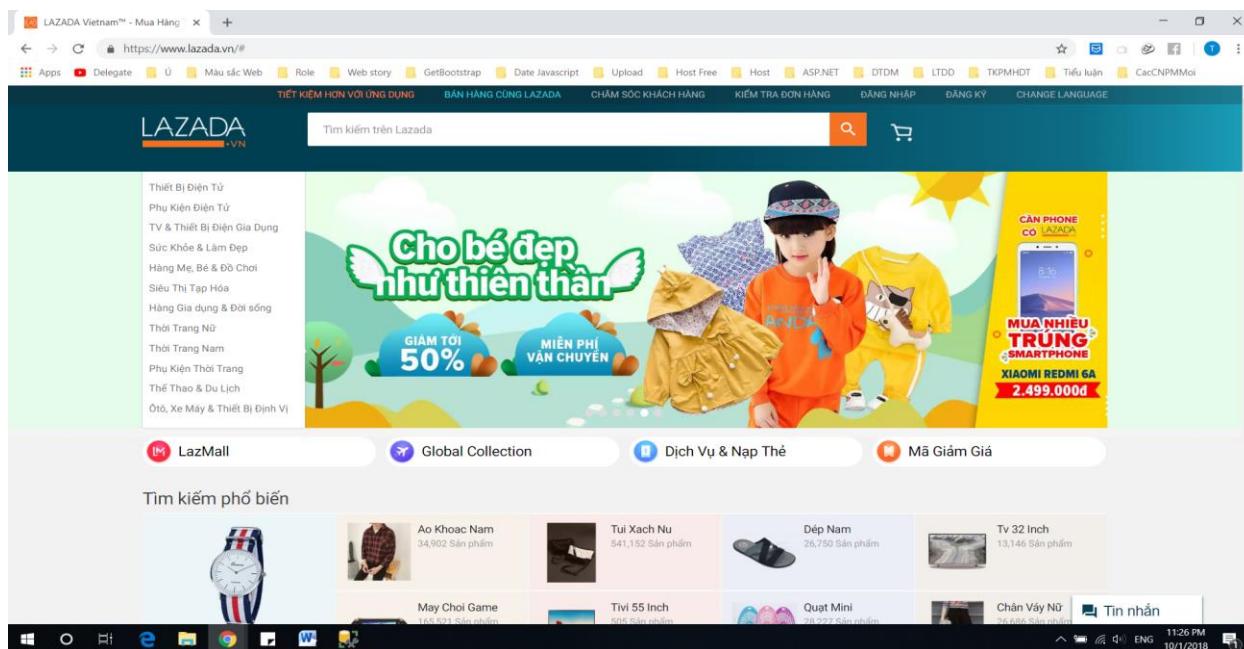
- Giao diện dễ nhìn, đẹp.
- UI/UX thân thiện.
- Có nhiều trò chơi, sự kiện để săn mã giảm giá.
- Trang thể hiện đầy đủ các thông tin cần thiết, hữu ích.
- Có bình luận, đánh giá, chat.
- Có sự kiểm duyệt từ Nhà phát triển, đảm bảo độ an toàn của sản phẩm.
- Độ tin cậy của website cao.

Kết: Giao diện rất đẹp, dễ sử dụng. UI/UX dễ sử dụng. Chức năng rất tốt.

2.1.4. Lazada

Trang chủ: <https://www.lazada.vn/>

Lazada là đối thủ cạnh tranh số một với Tiki. Nó cũng có các chức năng tương tự như tiki, cũng có rất nhiều sản phẩm đa dạng, đầy đủ thể loại.



Hình 2. 4: Giao diện Lazada

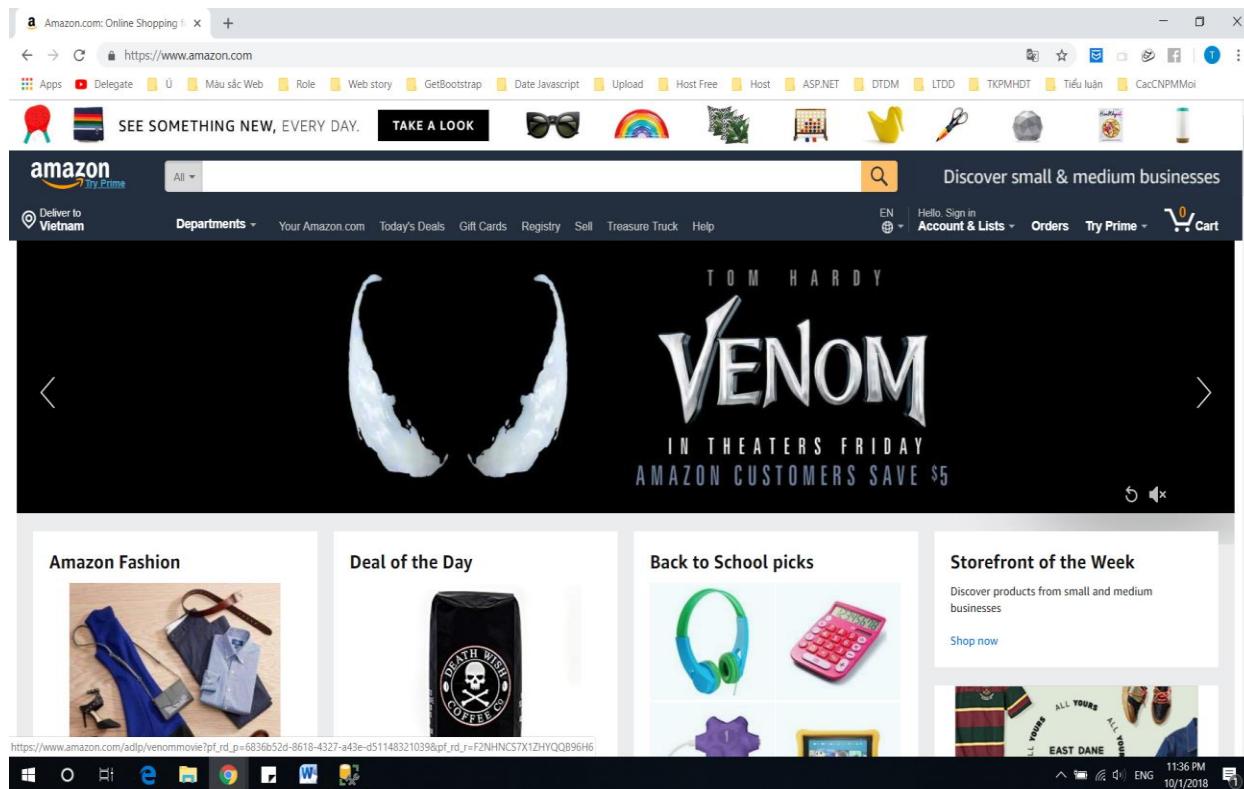
- Giao diện đẹp, bắt mắt, bố cục rõ ràng, thể hiện đầy đủ các thông tin của sản phẩm.
- Thao tác mua hàng thân thiện, tính tiện dụng cao.
- Dễ dàng tìm kiếm các sản phẩm theo danh mục
- Hình ảnh các sản phẩm chân thực.
- Có các tính năng đánh giá, bình luận, chat trực tuyến.
- Có sự kiểm duyệt đảm bảo độ tin cậy của sản phẩm.

Kết: Giao diện rất đẹp, dễ sử dụng. UI/UX dễ sử dụng.

2.1.5. Amazon

Trang chủ: <https://www.amazon.com/>

Đây là website thương mại điện tử đa quốc gia. Các sản phẩm trong đây rất đa dạng, đến từ khắp nơi trên thế giới. Có thể nói đây là trang web nên tham khảo để xây dựng một website bán hàng hoàn chỉnh.



Hình 2. 5: Giao diện Amazon

- Bố cục trang web hợp lý, theo từng loại sản phẩm giúp người tiêu dùng dễ dàng xem.
- Đủ các thể loại sản phẩm, trang mua bán trực tuyến lớn nhất.
- Thao tác tìm kiếm và mua hàng thể hiện rõ tính tiện dụng.
- Có tính năng đánh giá, bình luân.
- Sản phẩm được kiểm duyệt để đảm bảo độ tin cậy.

Kết: Giao diện đẹp, dễ sử dụng, tiện lợi và đơn giản.

2.2. PHÂN TÍCH HIỆN TRẠNG

Phạm vi: Người dùng cá nhân.

Website thương mại điện tử B-2-C nhằm mục đích cung cấp thể giới quan mua-bán giữa doanh nghiệp với người, là cầu nối mọi người, giúp rút ngắn khoảng cách giữa người bán và người mua, giúp cho người quản trị và người bán có thể quản lý hiệu quả hơn cũng như người mua có thể mua sắm tiện lợi, dễ dàng hơn. Sau đây là phần mô tả các nghiệp vụ hằng ngày của trang web:

- Website này là một hệ thống quản lý, trình bày các sản phẩm của 1 doanh nghiệp. Mỗi sản phẩm thuộc vào một danh mục sản phẩm. Tùy vào từng sản phẩm sẽ có giá cả, màu sắc, kích thước và số lượng khác nhau.
- Người dùng vào trang web có thể Tìm kiếm, Xem thông tin các sản phẩm họ mong muốn với tư cách là Khách. Người dùng có thể tạo tài khoản đăng nhập, tài khoản này sẽ gồm các thông tin cá nhân cần thiết và có thêm một tính năng bảo mật người dùng. Khi đăng nhập vào tài khoản thành công, người dùng dễ dàng hơn trong việc lựa chọn và lưu sản phẩm vào giỏ hàng, hoặc liên hệ trực tiếp với doanh nghiệp, mua hàng, xem đơn đặt hàng, đánh giá bình luận các sản phẩm.
- Mỗi người dùng sẽ có một giỏ hàng để dễ dàng mua và chọn lọc sản phẩm, ngoài ra còn có thêm chi tiết giỏ hàng để lưu các sản phẩm mà người dùng ưng ý. Người dùng sau khi mua hàng sẽ tạo thành một đơn hàng (hóa đơn).
- Trang web sẽ có thêm những người quản trị, là nhân viên của doanh nghiệp. Những người này sẽ tham gia vào việc thêm mới, kiểm soát, kiểm duyệt sản phẩm để trang web có thể phát huy hết toàn bộ khả năng.
- Ngoài ra. Website còn có chức năng thống kê để giúp các quản trị viên dễ dàng theo dõi tình hình kinh doanh theo ngày, tháng.

2.3. MỤC TIÊU

- Tìm hiểu .NET Core & Phát triển ứng dụng website bán hàng.
- Website cho phép các người dùng có thể đăng bán các sản phẩm của họ. Những khách hàng sẽ vào website và tìm lấy những sản phẩm mà mình mong muốn. Họ có thể chọn

sản phẩm trên màn hình hoặc là tìm kiếm sản phẩm từ khóa theo ý muốn của họ. Khi chọn lựa nếu có thắc mắc gì về sản phẩm họ có thể để lại bình luận hoặc là nhắn tin tới người bán. Khi đơn hàng được hoàn tất và giao tới khách hàng thì cũng có phần đánh giá để giúp những người khác dễ dàng chọn lựa đồ hơn.

- Áp dụng các kiến thức đã học ở những môn khác để thiết kế, xây dựng và hoàn thiện website.

2.4. PHÂN TÍCH YÊU CẦU

2.4.1. Yêu cầu chức năng

2.4.1.1. Lưu trữ

- Thông tin người dùng khi họ đăng ký tài khoản AspNetUsers: UserName, Mail, PasswordHash, PhoneNumber, HoTen, BirthDate, Sex. Thông tin sản phẩm khi đăng bán/mua Product: Name, UserId, Description.
- Thông tin chi tiết sản phẩm khi đã phân loại ProductDetail: ProductId, Price, Size, Color, Quantity.
- Giỏ hàng để thêm các sản phẩm ưa thích của khách hàng Cart: UserId.
- Đơn hàng để lưu thông tin mua hàng của khách hàng Order: UserId, PriceSum, Date.
- Các danh mục hàng trong website Catalog: Name.
- Danh sách hình ảnh để hiển thị sản phẩm Image: Url.

2.4.1.2. Tra cứu

- Tìm kiếm sản phẩm theo từ khóa (gần đúng), hoặc tìm kiếm chính xác sản phẩm theo tên.
- Tìm kiếm tên người dùng theo từ khóa.

2.4.1.3. Tính toán

- Tính tổng giá của đơn hàng công thức: Giá x Số lượng.
- Tính tổng số lượng sản phẩm theo danh mục.

2.4.1.4. Kết xuất

- Thống kê sản phẩm đang được bán trên website.
- Thống kê sản phẩm được bán trong ngày, tháng,...

- Thống kê chi tiết sản phẩm theo sản phẩm.
- Thống kê doanh thu theo ngày, tháng,...

2.4.1.5. *Ghi chú*

- Tất cả các bảng ở Lưu trữ sẽ có thêm thuộc tính là State (mặc định = 1) để tương tác với Chức năng Xóa (State = 0: Đã xóa).
- Tất cả các bảng sẽ có Id kiểu Int tăng dần, bắt đầu từ 0, bước nhảy là 1. (Trừ bảng AspNetUser.Id là kiểu GUID).
- Do quan hệ giữa Cart-Product là n-n nên sẽ có Bảng CartDetail phát sinh để lưu trữ thông tin giữa 2 bảng. Order-Product tương tự và phát sinh bảng OrderDetail.
- Riêng Quản lý sản phẩm sẽ có thêm chức năng Xem lịch sử quản lý (xem các mốc thời gian, sự thay đổi của sản phẩm).

2.4.2. Yêu cầu phi chức năng

Bảng 2. 1: Bảng yêu cầu phi chức năng

STT	Nội dung	Tiêu chuẩn	Mô tả chi tiết
1	Tốc độ xử lý quá trình mua hàng nhanh.	Hiệu quả	Một lần có thể cho nhiều tài khoản mua hàng.
2	Tốc độ tìm kiếm nhanh và chính xác.	Hiệu quả	Tối đa 3s phải có kết quả tìm kiếm.
3	Tiết kiệm được thời gian, thu hẹp không gian lưu trữ, tránh thất lạc dữ liệu.	Hiệu quả	Tiết kiệm thời gian so với quản lý thủ công. Dữ liệu được sao lưu trên máy, có thể dễ dàng phục hồi.

4	Dễ dàng quản lý người dùng, bán hàng và mua hàng.	Tiện dụng	Chỉ cần thông qua chức năng quản sản phẩm, người dùng có thể dễ dàng kiểm soát được tình hình từng sản phẩm đã đăng bán của mình, không cần tốn nhiều thời gian, công sức.
5	Có thể thiết kế thêm theo yêu cầu của quản lý bán hàng trực tuyến.	Tiến hóa	Quản lý có thể yêu cầu thêm tính năng hữu ích cho chương trình quản bán hàng trực tuyến.
6	Cho phép thay đổi quy định, công thức tính toán	Tiến hóa	Người dùng có thể dễ dàng thay đổi quy định tính tổng tiền thanh toán khi áp dụng mã giảm giá, quy định kiểm duyệt sản phẩm.
7	Giao diện thân thiện, đơn giản	Tiện dụng	Khách hàng ở mọi lứa tuổi đều có thể dễ dàng sử dụng.
8	Dễ thao tác	Tiện dụng	Các thao tác nhập xuất chỉnh sửa đơn giản, gần gũi với người dùng.

2.5. MÔ HÌNH HÓA YÊU CẦU

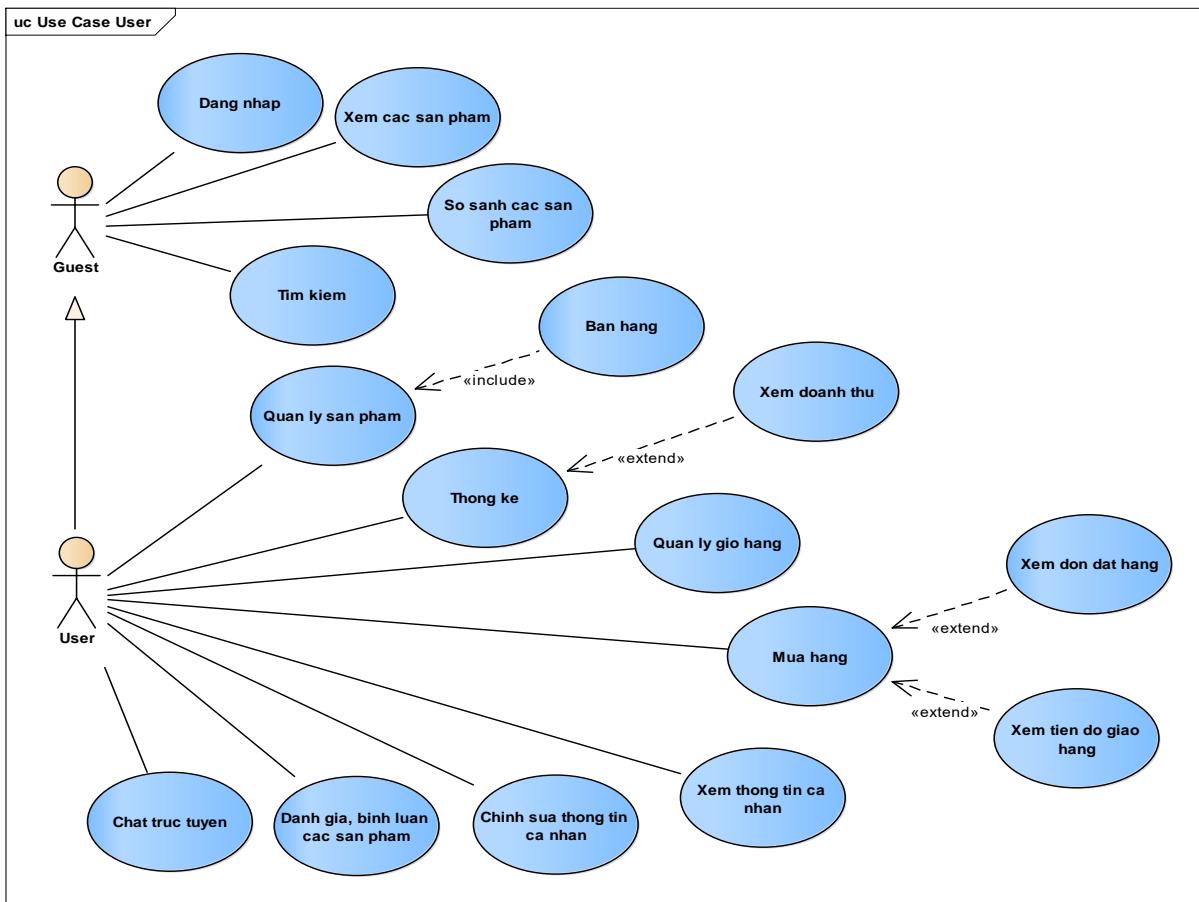
2.5.1. Nhận diện tác nhân và chức năng trong sơ đồ use case

Bảng 2. 2: Nhận diện tác nhân và chức năng trong sơ đồ use case

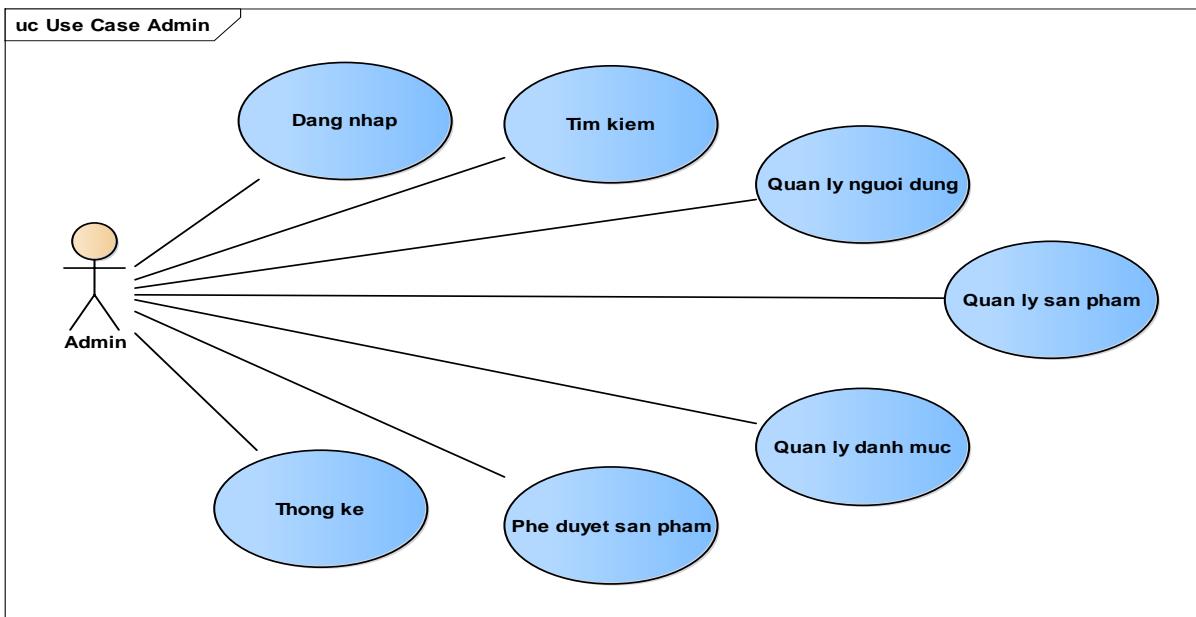
Tác nhân	Chức năng

Khách	<ul style="list-style-type: none"> - Đăng nhập, đăng xuất - Tìm kiếm - Xem thông tin các sản phẩm
Người dùng	<ul style="list-style-type: none"> - Đăng nhập, đăng xuất - Tìm kiếm - Xem thông tin các sản phẩm - Thống kê - Mua hàng - Quản lý thông tin cá nhân
Quản trị viên	<ul style="list-style-type: none"> - Đăng nhập, đăng xuất - Tìm kiếm - Quản lý người dùng - Quản lý sản phẩm - Quản lý danh mục - Thống kê doanh số

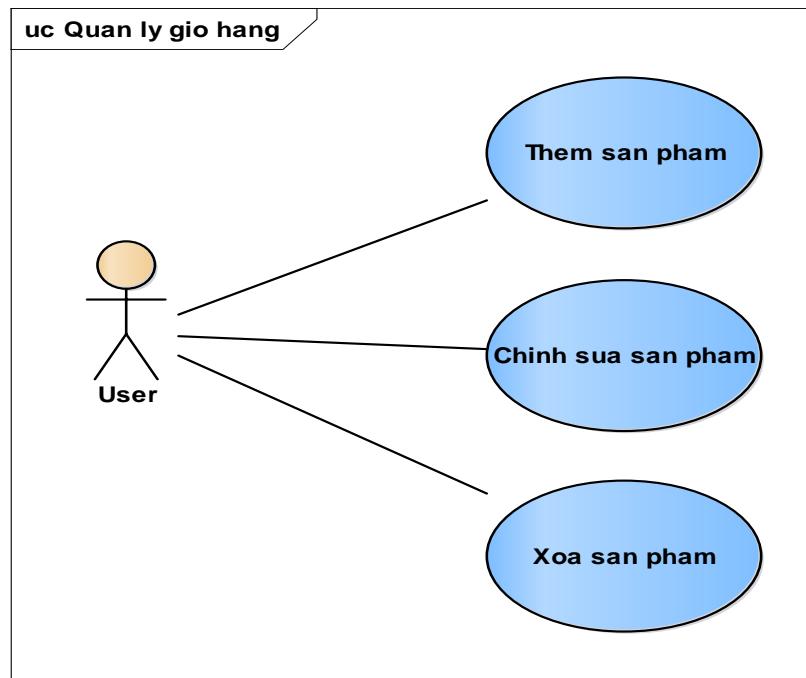
2.5.2. Sơ đồ Usecase



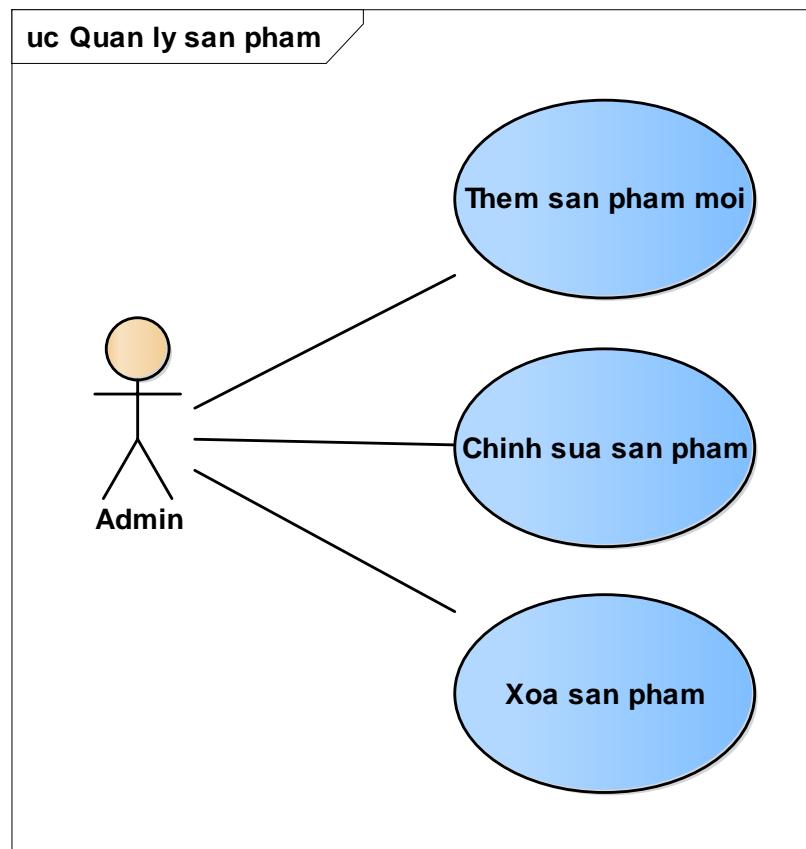
Hình 2. 6: Usecase Người dùng



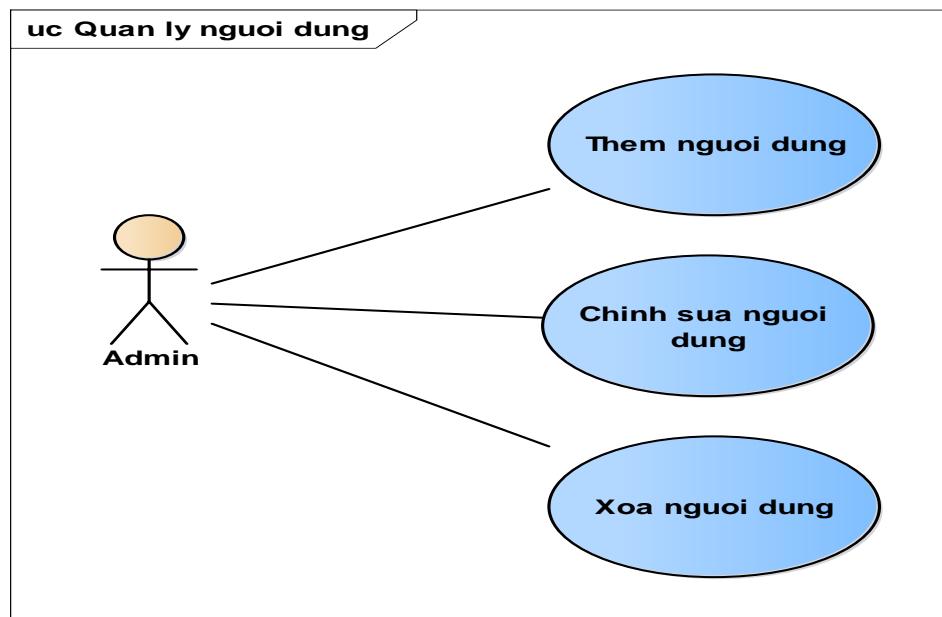
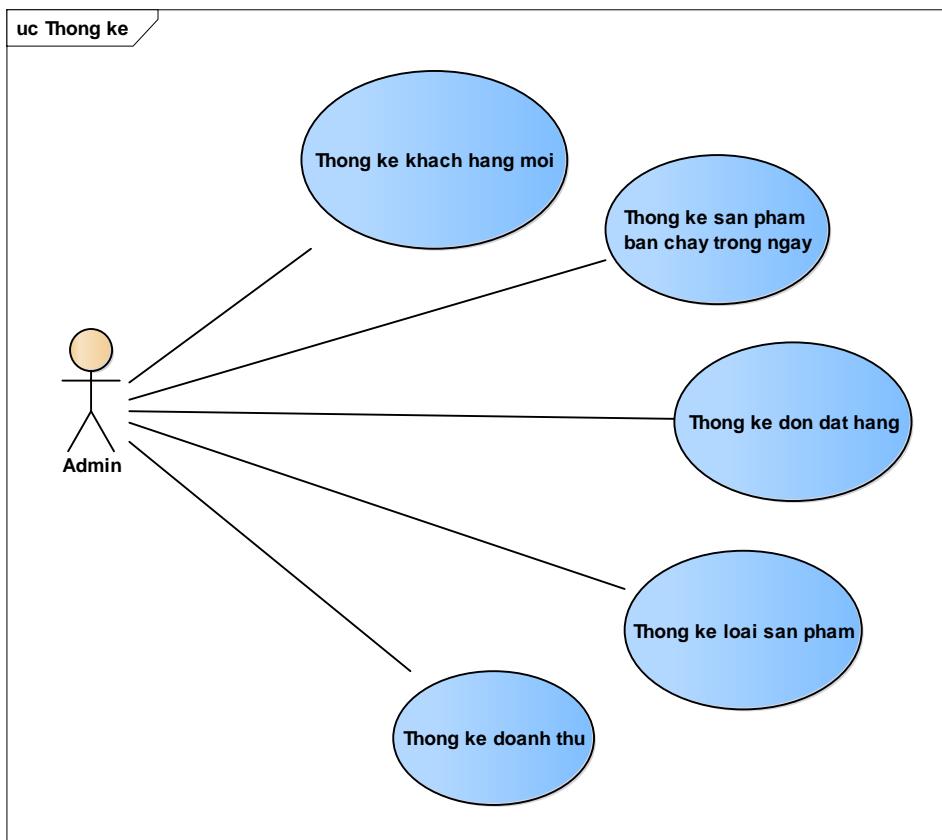
Hình 2. 7: Usecase Quản trị viên

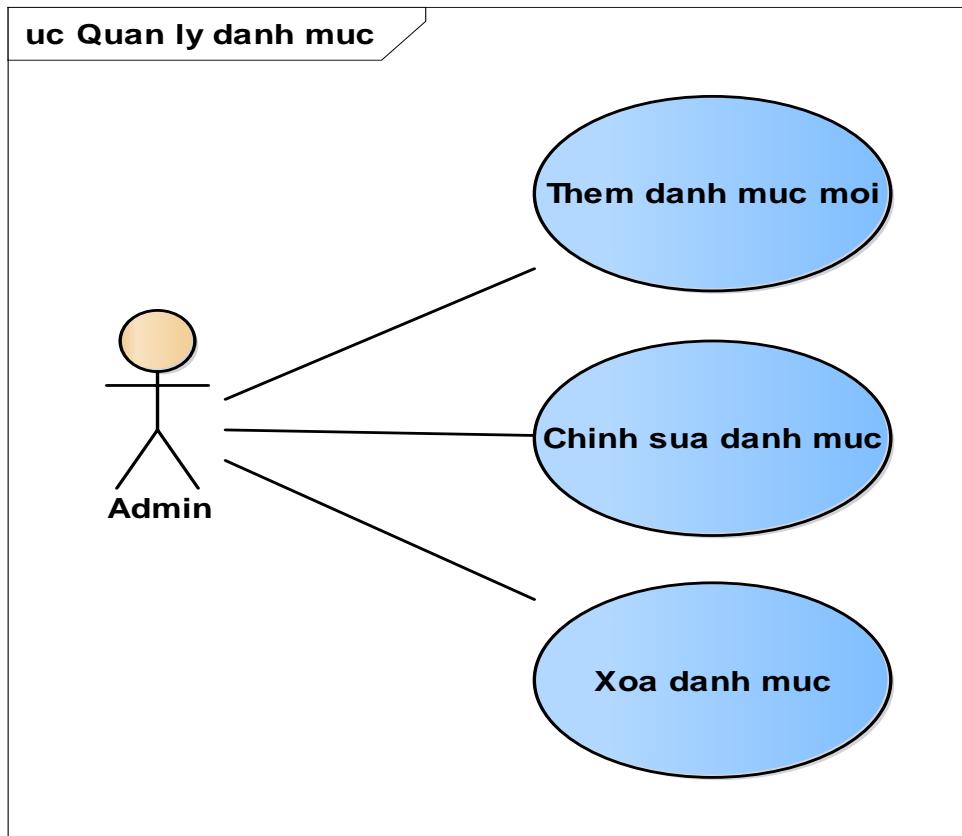


Hình 2. 8: Usecase Quản lý giỏ hàng



Hình 2. 9: Usecase Quản lý sản phẩm

**Hình 2. 10: Usecase Quản lý người dùng****Hình 2. 11: Usecase Thông kê**

**Hình 2. 12: Usecase Quản lý danh mục**

2.5.3. Đặc tả Usecase

2.5.3.1. Đăng nhập

Bảng 2. 3: Mô tả Usecase Đăng nhập

Đăng nhập	
Mô tả	Sử dụng tài khoản đã đăng ký để đăng nhập vào website.
Tác nhân kích hoạt	Khách
Tiền điều kiện	Người dùng truy cập vào website
Các bước thực hiện	<ol style="list-style-type: none"> (1) Người dùng truy cập vào website (2) Click button Đăng nhập ở góc phải trên của website. (3) Nhập form đăng nhập: Email/ Số điện thoại + Mật khẩu. (4) So sánh thông tin form đăng nhập với Database:

	<ul style="list-style-type: none"> Đúng: Chuyển đến Trang chủ. Sai/Lỗi: Thông báo “Đăng nhập thất bại”.
--	---

2.5.3.2. *Thêm sản phẩm*

Bảng 2. 4: Mô tả Usecase Thêm sản phẩm

Thêm sản phẩm	
Mô tả	Quản trị viên thêm một sản phẩm mới của mình
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Người dùng đăng nhập thành công
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Quản lý sản phẩm, sau đó chọn Thêm sản phẩm.</p> <p>(4) Chuyển đến trang Thêm sản phẩm mới.</p> <p>(5) Quản trị viên nhập các thông tin như: Danh mục sản phẩm, tên sản phẩm, mô tả sản phẩm.</p> <p>(6) Sau đó nhập thông tin chi tiết cho sản phẩm như: Kích thước, số lượng, màu sắc, đơn giá và chọn hình cho sản phẩm.</p> <p>(7) Bấm vào Thêm sản phẩm để lưu thông tin sản phẩm.</p>

2.5.3.3. *Chỉnh sửa sản phẩm*

Bảng 2. 5: Mô tả Usecase Chính sửa sản phẩm

Chỉnh sửa sản phẩm	
Mô tả	Quản trị viên chỉnh sửa sản phẩm đã có trong cơ sở dữ liệu

Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Thông tin sản phẩm đã có trên cơ sở dữ liệu
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Quản lý sản phẩm.</p> <p>(4) Danh sách các sản phẩm của người dùng đã thêm trước đó sẽ được hiển thị lên màn hình.</p> <p>(5) Quản trị viên có thể chọn trực tiếp từ danh sách trên màn hình hoặc nhập thông tin vào khung tìm kiếm theo các hướng khác nhau: mã sản phẩm, tên sản phẩm.</p> <p>(6) Sau khi tìm đúng sản phẩm cần chỉnh sửa, người dùng nhập lại thông tin mới.</p> <p>(7) Lưu thông tin mới cho sản phẩm.</p>

2.5.3.4. Xóa sản phẩm

Bảng 2.6: Mô tả Usecase Xóa sản phẩm

Xóa sản phẩm	
Mô tả	Quản trị viên xóa sản phẩm đã có trong cơ sở dữ liệu
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Thông tin sản phẩm đã có trên cơ sở dữ liệu
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Quản lý sản phẩm.</p> <p>(4) Danh sách các sản phẩm của người dùng đã thêm trước đó sẽ được hiển thị lên màn hình.</p>

	<p>(5) Người dùng có thể chọn trực tiếp từ danh sách trên màn hình hoặc nhập thông tin vào khung tìm kiếm theo các hướng khác nhau: mã sản phẩm, tên sản phẩm.</p> <p>(6) Sau khi tìm đúng sản phẩm cần xóa, người dùng nhấn nút xóa.</p> <p>(7) Chọn “Xác nhận” để đồng ý xóa sản phẩm.</p>
--	--

2.5.3.5. *Thêm sản phẩm vào giỏ hàng*

Bảng 2. 7: Mô tả Usecase Thêm sản phẩm vào giỏ hàng

Thêm sản phẩm vào giỏ hàng	
Mô tả	Người dùng thêm một sản phẩm vào giỏ hàng
Tác nhân kích hoạt	Người dùng
Tiền điều kiện	Người dùng truy cập vào website
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập.</p> <p>(3) Chọn sản phẩm mình thích và nhấn vào nút Thêm vào giỏ hàng.</p> <p>(4) Người dùng có thể tìm kiếm sản phẩm thêm tên sản phẩm, loại sản phẩm sau đó nhấn vào Thêm vào giỏ hàng.</p> <p>(5) Người dùng có thể bấm vào nút Đến giỏ hàng để kiểm tra hoặc tiếp tục xem sản phẩm.</p> <p>(6) Các sản phẩm đã thêm vẫn sẽ tồn tại trong giỏ hàng của người dùng ở các lần mua hàng tiếp cho đến khi sản phẩm đó đã được thanh toán hoặc bị xóa khỏi giỏ hàng.</p>

2.5.3.6. *Chỉnh sửa giỏ hàng*

Bảng 2. 8: Mô tả Usecase *Chỉnh sửa giỏ hàng*

Chỉnh sửa sản phẩm trong giỏ hàng	
Mô tả	Người dùng chỉnh sửa sản phẩm đã được thêm vào giỏ hàng
Tác nhân kích hoạt	Người dùng
Tiền điều kiện	Người dùng truy cập vào website
Các bước thực hiện	<ul style="list-style-type: none"> (1) Người dùng truy cập vào website. (2) Đăng nhập. (3) Nhấp chọn vào Giỏ hàng. (4) Danh sách các sản phẩm chưa được mua trong giỏ hàng của người dùng sẽ hiện ra. (5) Người dùng có thể chỉnh sửa số lượng của các sản phẩm và chọn các sản phẩm muốn thanh toán.

2.5.3.7. *Xóa sản phẩm trong giỏ hàng*

Bảng 2. 9: Mô tả Usecase *Xóa sản phẩm trong giỏ hàng*

Xóa sản phẩm trong giỏ hàng	
Mô tả	Người dùng xóa sản phẩm đã được thêm vào giỏ hàng
Tác nhân kích hoạt	Người dùng
Tiền điều kiện	Người dùng truy cập vào website
Các bước thực hiện	<ul style="list-style-type: none"> (1) Người dùng truy cập vào website. (2) Đăng nhập. (3) Nhấp chọn vào Giỏ hàng.

	<p>(4) Danh sách các sản phẩm chưa được mua trong giỏ hàng của người dùng sẽ hiện ra.</p> <p>(5) Chọn sản phẩm muốn xóa và nhấn vào Xóa.</p> <p>(6) Nhấn chọn Xác nhận để hoàn tất xóa sản phẩm khỏi giỏ hàng.</p>
--	--

2.5.3.8. *Mua hàng*

Bảng 2. 10: Mô tả Usecase Mua hàng

Mua hàng	
Mô tả	Người dùng xác nhận mua các sản phẩm trong giỏ hàng
Tác nhân kích hoạt	Người dùng
Tiền điều kiện	Người dùng truy cập vào website
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập.</p> <p>(3) Nhấp chọn vào Giỏ hàng.</p> <p>(4) Danh sách các sản phẩm chưa được mua trong giỏ hàng của người dùng sẽ hiện ra.</p> <p>(5) Nhấn chọn vào Mua hàng.</p> <p>(6) Hệ thống sẽ chuyển người dùng đến trang Xác nhận mua hàng.</p> <p>(7) Danh sách các sản phẩm chưa được mua hiện ra. Người dùng có thể chọn tất cả sản phẩm hoặc chọn sản phẩm mình muốn mua.</p> <p>(8) Số tiền cần thanh toán tương ứng sẽ hiện ra, người dùng nhấn vào Hoàn tất mua hàng.</p>

2.5.3.9. Xem thông tin cá nhân

Bảng 2. 11: Mô tả Usecase Xem thông tin cá nhân

Xem thông tin cá nhân	
Mô tả	Người dùng xem thông tin cá nhân của mình
Tác nhân kích hoạt	Người dùng
Tiền điều kiện	Khi người dùng muốn xem thông tin cá nhân của mình
Các bước thực hiện	<ul style="list-style-type: none"> (1) Người dùng truy cập vào website. (2) Đăng nhập. (3) Chọn mục Thông tin cá nhân. (4) Thông tin cá nhân của người dùng hiện ra. (5) Người dùng có thể nhấn vào Chỉnh sửa để tiến hành chỉnh sửa thông tin cá nhân.

2.5.3.10. Chỉnh sửa thông tin cá nhân

Bảng 2. 12: Mô tả Usecase Chỉnh sửa thông tin cá nhân

Use Case Chỉnh sửa thông tin cá nhân	
Mô tả	Người dùng chỉnh sửa thông tin cá nhân của mình
Tác nhân kích hoạt	Người dùng
Tiền điều kiện	Khi người dùng muốn chỉnh sửa thông tin cá nhân của mình
Các bước thực hiện	<ul style="list-style-type: none"> (1) Người dùng truy cập vào website. (2) Đăng nhập. (3) Chọn mục Thông tin cá nhân. (4) Nhấn chọn Chỉnh sửa.

	(5) Tiến hành chỉnh sửa các thông tin muốn sửa sau đó nhấn Lưu để lưu vào cơ sở dữ liệu.
--	---

2.5.3.11. *Dánh giá, bình luận các sản phẩm*

Bảng 2. 13: Mô tả Usecase Đánh giá, bình luận các sản phẩm

Use Case Đánh giá, bình luận các sản phẩm	
Mô tả	Người dùng đánh giá, bình luận các sản phẩm được đăng bán trên website
Tác nhân kích hoạt	Người dùng
Tiền điều kiện	Khi người dùng quan tâm đến một sản phẩm nào đó trên web
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập.</p> <p>(3) Chọn một sản phẩm muốn bình luận, đánh giá.</p> <p>(4) Tiến hành đánh giá và bình luận sản phẩm.</p> <p>(5) Người dùng cũng có thể trả lời bình luận của người khác.</p>

2.5.3.12. *Tìm kiếm sản phẩm*

Bảng 2. 14: Mô tả Usecase Tìm kiếm sản phẩm

Tìm kiếm sản phẩm	
Mô tả	Tìm kiếm theo Tên sản phẩm (từ khóa).
Tác nhân kích hoạt	Người dùng, Admin
Tiền điều kiện	Khi người dùng có muốn tìm sản phẩm nào đó.
Các bước thực hiện	<p>(1) Người dùng truy cập vào website</p>

	<p>(2) Nhập Từ khóa tìm kiếm vào Thanh Tìm Kiếm ở đầu trang web.</p> <p>(3) Web sẽ tự hiển thị gợi ý theo từ khóa đã nhập. Người dùng có thể chọn gợi ý, hoặc là theo từ khóa.</p> <p>(4) Click button Tìm Kiếm.</p> <p>(5) Web sẽ load sản phẩm liên quan theo Gợi ý, Từ Khóa trên.</p>
--	--

2.5.3.13. Thêm người dùng

Bảng 2. 15: Mô tả Usecase Thêm người dùng

Thêm Người dùng	
Mô tả	Quản trị viên thêm người dùng vào website
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Đăng nhập vào quyền Admin thành công.
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Quản lý người dùng, sau đó chọn Thêm người dùng.</p> <p>(4) Chuyển đến trang Thêm người dùng.</p> <p>(5) Quản trị viên nhập các thông tin như: Tài khoản, mật khẩu, Quyền, Họ Tên.</p> <p>(6) Nhấn vào Thêm người dùng để hệ thống lưu thông tin vào cơ sở dữ liệu.</p>

2.5.3.14. Chính sửa người dùng

Chỉnh sửa Người dùng

Mô tả	Quản trị viên chỉnh sửa người dùng đã có trong cơ sở dữ liệu
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Đăng nhập vào quyền Admin thành công.
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Quản lý người dùng.</p> <p>(4) Danh sách các người dùng đã thêm trước đó sẽ được hiển thị lên màn hình.</p> <p>(5) Quản trị viên có thể chọn trực tiếp từ danh sách trên màn hình hoặc nhập thông tin vào khung tìm kiếm theo: Tên người dùng.</p> <p>(6) Sau khi tìm đúng người dùng quản trị viên chọn nút Chính sửa.</p> <p>(7) Chuyển đến trang Chính sửa người dùng.</p> <p>(8) Quản trị viên nhập thông tin mới của người dùng sau đó nhấn nút Cập nhật.</p>

2.5.3.15. Xóa người dùng

Bảng 2. 16: Mô tả Usecase Xóa người dùng

Xóa Người dùng	
Mô tả	Quản trị viên xóa người dùng trong cơ sở dữ liệu
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	<p>Đăng nhập vào quyền Admin thành công.</p> <p>Người dùng đã có trên cơ sở dữ liệu.</p>

Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Quản lý người dùng.</p> <p>(4) Hiển thị Danh sách các người dùng trong cơ sở dữ liệu.</p> <p>(5) Quản trị viên có thể chọn trực tiếp từ danh sách trên màn hình hoặc nhập thông tin vào khung tìm kiếm theo các hướng khác nhau: Mã người dùng, tên người dùng.</p> <p>(6) Click button Xóa ở dòng của người dùng muốn xóa.</p> <p>(7) Người dùng bị disable => tương đồng với Xóa</p> <p>(8) Lưu.</p>
---------------------------	---

2.5.3.16. Tìm kiếm người dùng

Bảng 2. 17: Mô tả Usecase Tìm kiếm người dùng

Tìm kiếm người dùng	
Mô tả	Tìm kiếm theo Tên (từ khóa).
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Khi người dùng có muốn tìm người dùng nào đó.
Các bước thực hiện	<p>(1) Người dùng truy cập vào website</p> <p>(2) Nhập Từ khóa tìm kiếm vào Thanh Tìm Kiếm ở trang quản lý người dùng.</p> <p>(3) Click button Tìm Kiếm.</p> <p>(4) Trang sẽ load sản phẩm liên quan theo Từ Khóa lên.</p>

2.5.3.17. Thêm danh mục

Bảng 2. 18: Mô tả Usecase Thêm danh mục

Thêm danh mục

Mô tả	Quản trị viên thêm danh mục sản phẩm vào website
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Đăng nhập vào quyền Admin thành công.
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Quản lý danh mục, sau đó chọn Thêm danh mục mới.</p> <p>(4) Chuyển đến trang Thêm danh mục.</p> <p>(5) Quản trị viên nhập các thông tin như: Tên danh mục, mô tả danh mục.</p> <p>(6) Nhấn vào Thêm danh mục để hệ thống lưu thông tin vào cơ sở dữ liệu.</p>

2.5.3.18. *Chỉnh sửa danh mục*

Bảng 2. 19: Mô tả Usecase *Chỉnh sửa danh mục*

Chỉnh sửa danh mục	
Mô tả	Quản trị viên chỉnh sửa danh mục đã có trong cơ sở dữ liệu
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Đăng nhập vào quyền Admin thành công
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Quản lý danh mục.</p> <p>(4) Danh sách các danh mục đã thêm trước đó sẽ được hiển thị lên màn hình.</p>

	<p>(5) Quản trị viên có thể chọn trực tiếp từ danh sách trên màn hình hoặc nhập thông tin vào khung tìm kiếm theo: Tên danh mục.</p> <p>(6) Sau khi tìm đúng danh mục quản trị viên chọn nút Chính sửa.</p> <p>(7) Chuyển đến trang Chính sửa danh mục.</p> <p>(8) Quản trị viên nhập thông tin mới của danh mục sau đó nhấn nút Cập nhật.</p>
--	--

2.5.3.19. Xóa danh mục

Bảng 2. 20: Mô tả Usecase Xóa danh mục

Xóa danh mục	
Mô tả	Quản trị viên xóa danh mục trong cơ sở dữ liệu
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Đăng nhập vào quyền Admin thành công
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Quản lý danh mục.</p> <p>(4) Hiển thị Danh sách các danh mục trong cơ sở dữ liệu.</p> <p>(5) Quản trị viên có thể chọn trực tiếp từ danh sách trên màn hình hoặc nhập thông tin vào khung tìm kiếm theo các hướng khác nhau: Tên danh mục.</p> <p>(6) Click button Xóa ở dòng của danh mục muốn xóa.</p> <p>(7) Danh mục bị disable => tương đồng với Xóa</p> <p>(8) Lưu.</p>

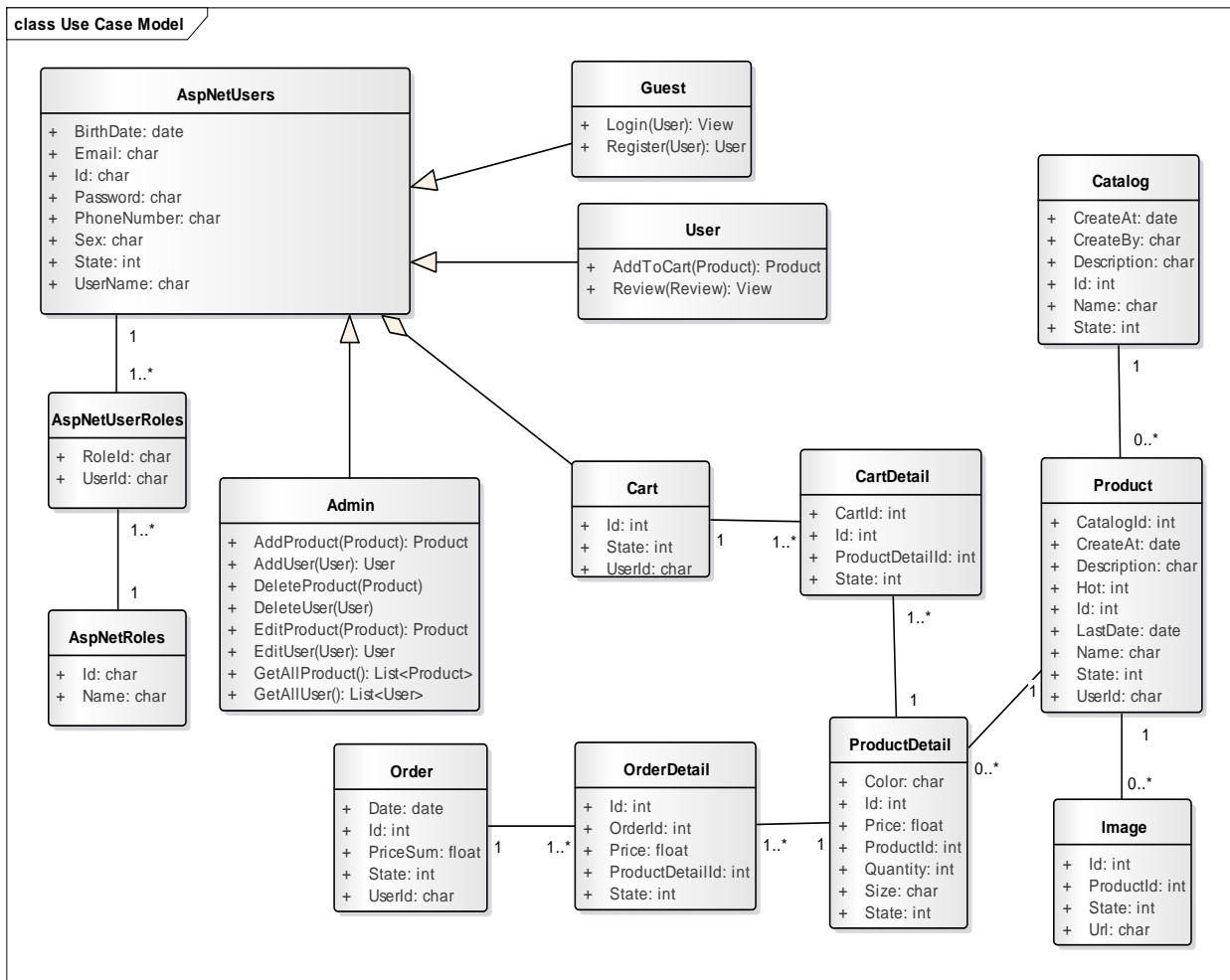
2.5.3.20. *Thống kê***Bảng 2. 21: Mô tả Usecase Thống kê**

Xóa danh mục	
Mô tả	Quản trị viên xem chi tiết thống kê
Tác nhân kích hoạt	Quản trị viên
Tiền điều kiện	Đăng nhập vào quyền Admin thành công
Các bước thực hiện	<p>(1) Người dùng truy cập vào website.</p> <p>(2) Đăng nhập với quyền Admin.</p> <p>(3) Chọn mục Thống kê.</p> <p>(4) Hệ thống sẽ hiển thị biểu đồ thống kê doanh thu và đơn đặt hàng theo từng tháng.</p> <p>(5) Đồng thời hệ thống còn hiển thị số lượng đơn hàng, khách hàng mới, doanh thu trong ngày.</p> <p>(6) Người dùng có thể bấm vào nút Export to Excel để xuất file excel thống kê.</p>

CHƯƠNG 3: THIẾT KẾ PHẦN MỀM

3.1. THIẾT KẾ HỆ THỐNG

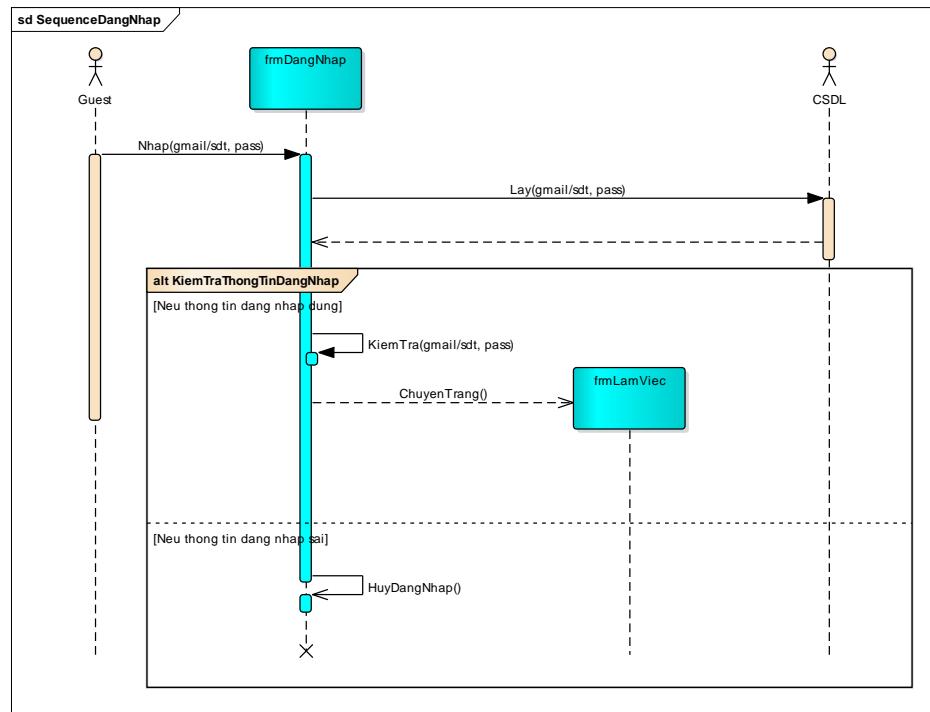
3.1.1. Lược đồ lớp



Hình 3. 1: Lược đồ lớp

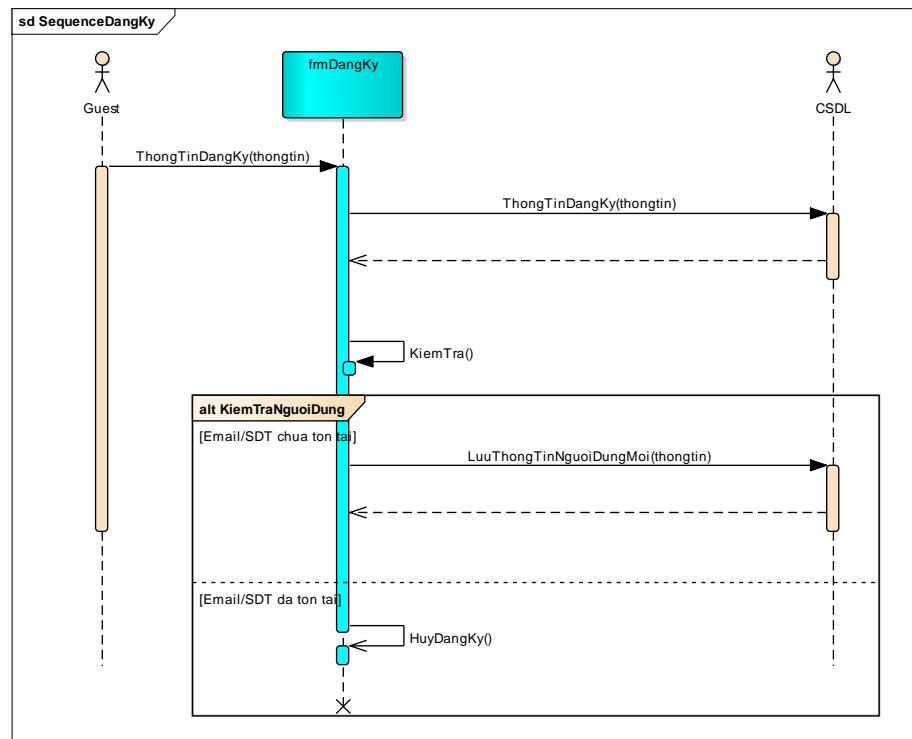
3.1.2. Lược đồ Sequence

3.1.2.1. Sequence Đăng nhập



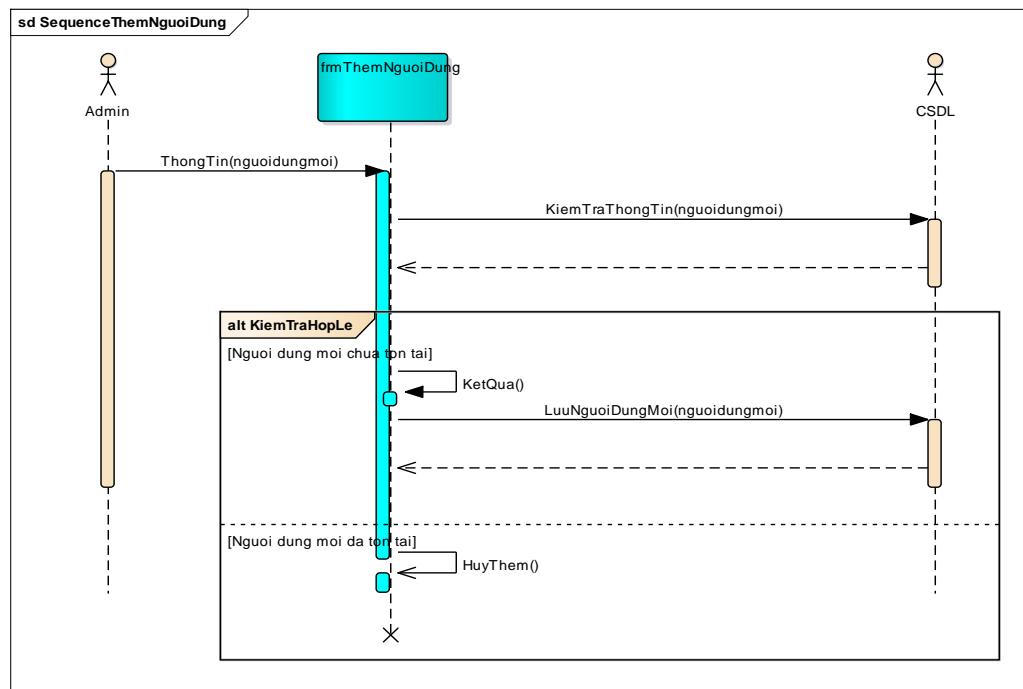
Hình 3. 2: Sequence Đăng nhập

3.1.2.2. Sequence Đăng ký



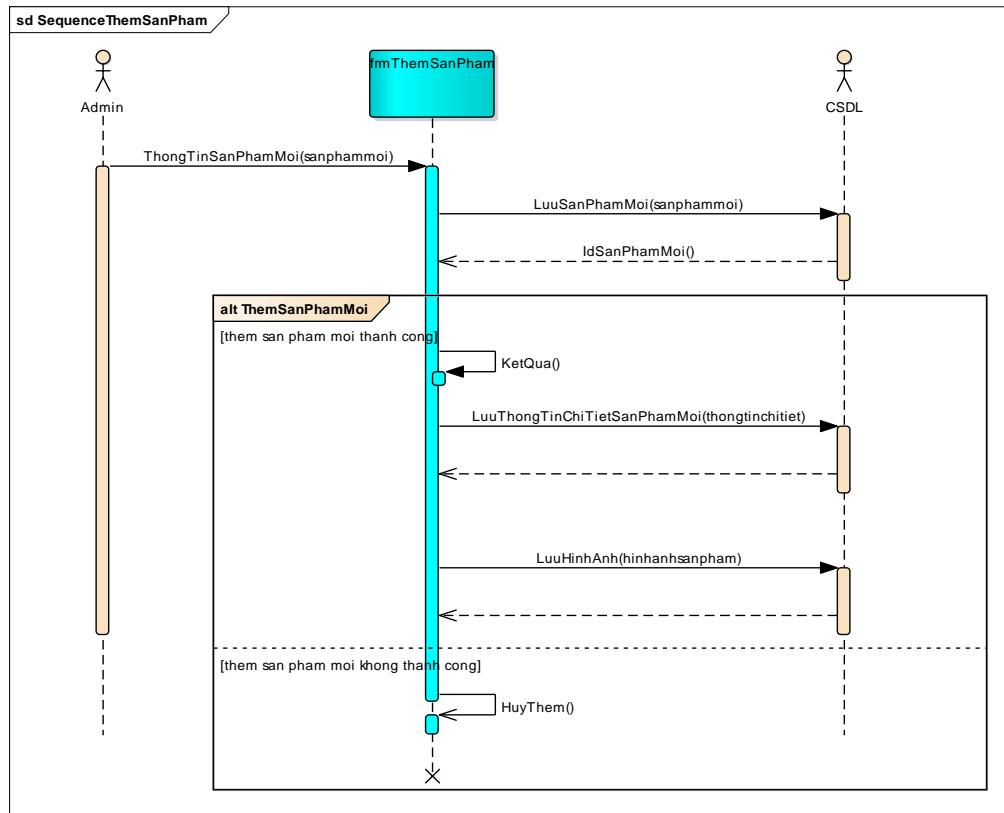
Hình 3. 3: Sequence Đăng ký

3.1.2.3. Sequence Thêm người dùng



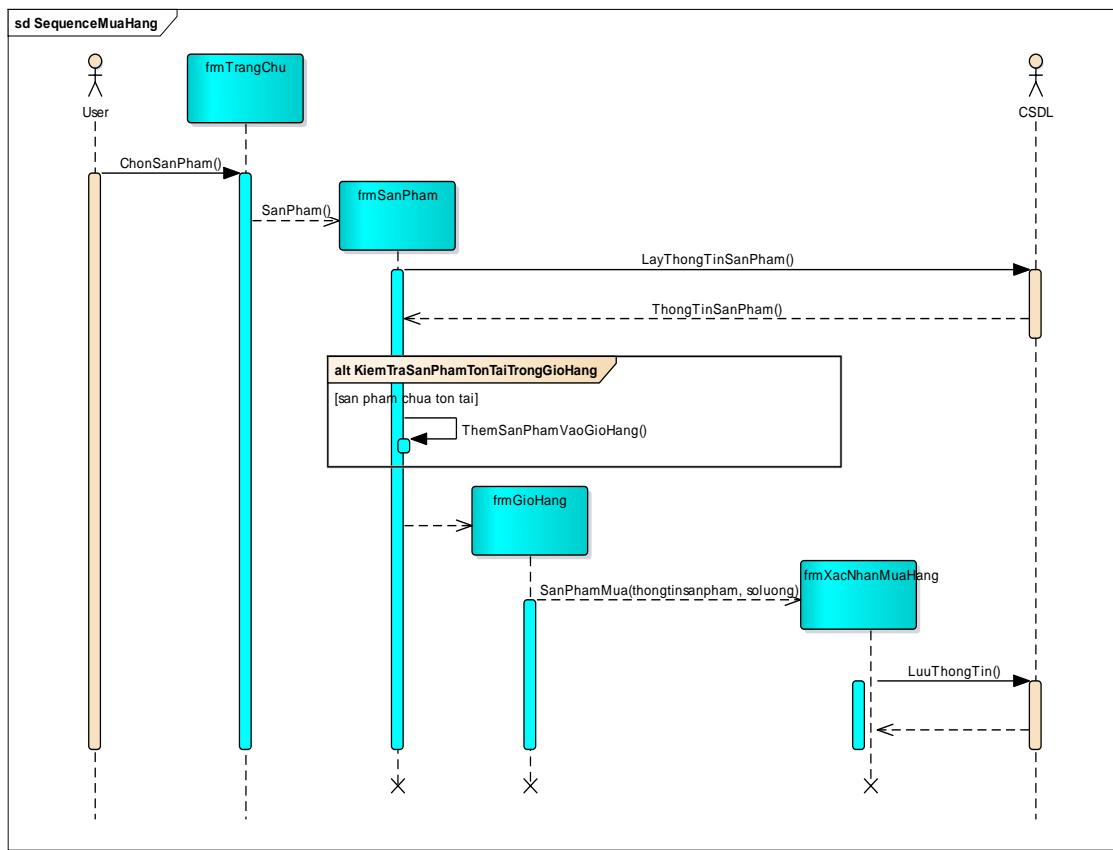
Hình 3. 4: Sequence Thêm người dùng

3.1.2.4. Sequence Thêm sản phẩm



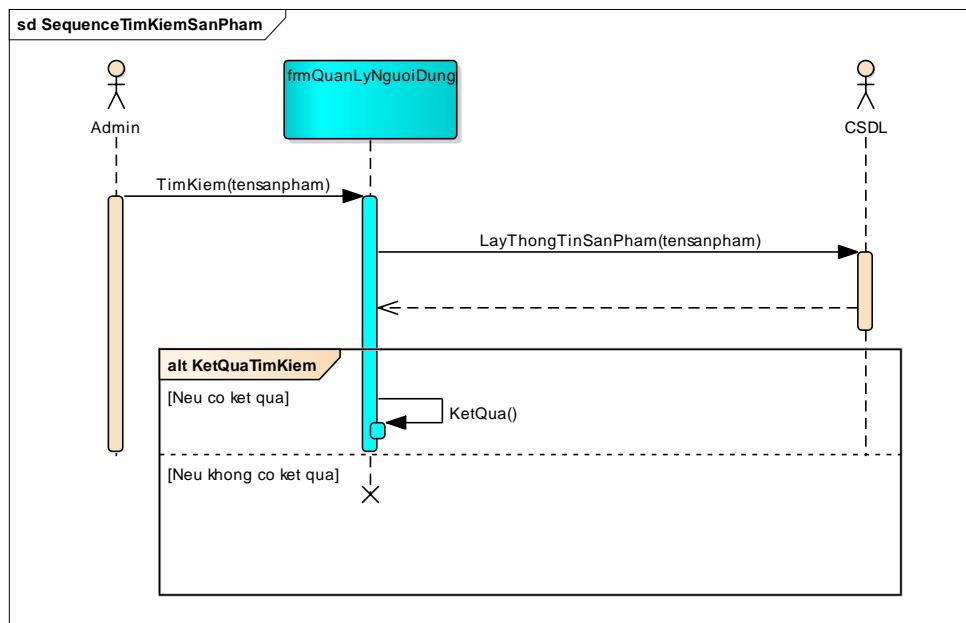
Hình 3. 5: Sequence Thêm sản phẩm

3.1.2.5. Sequence Mua hàng



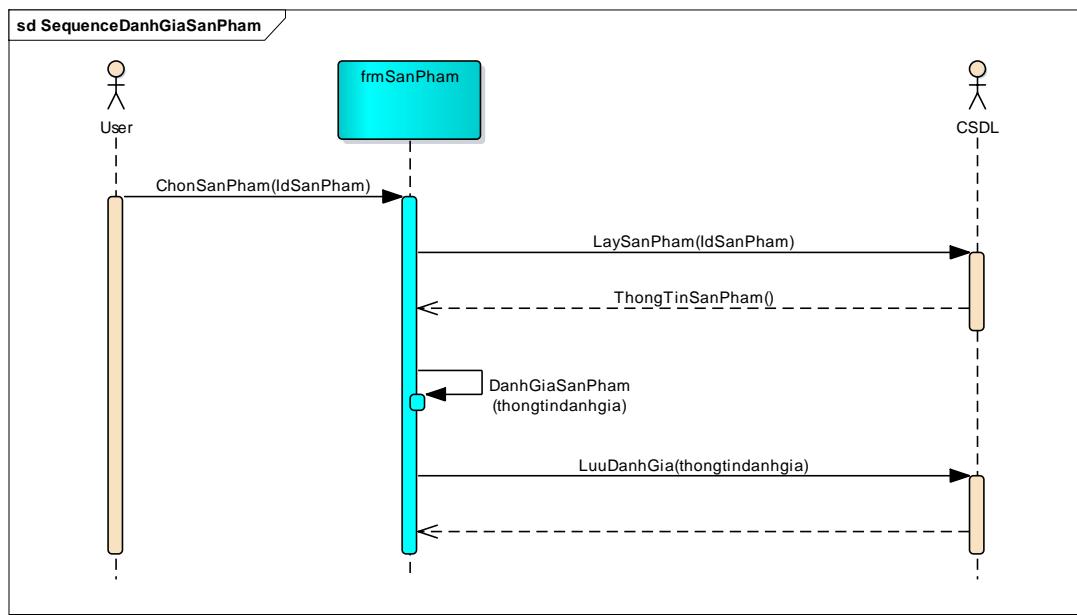
Hình 3. 6: Sequence Mua hàng

3.1.2.6. Sequence Tìm kiếm sản phẩm



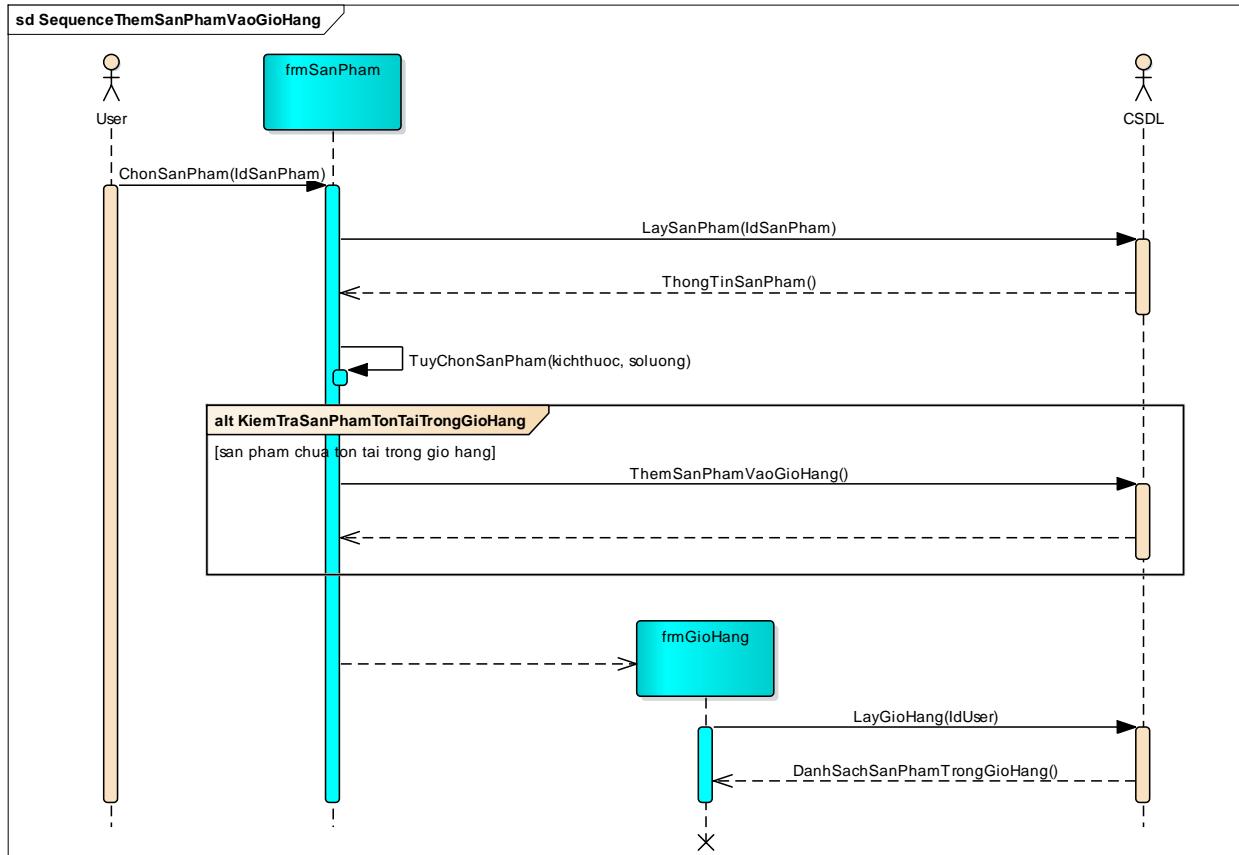
Hình 3. 7: Sequence Tìm kiém sản phẩm

3.1.2.7. Sequence Đánh giá sản phẩm



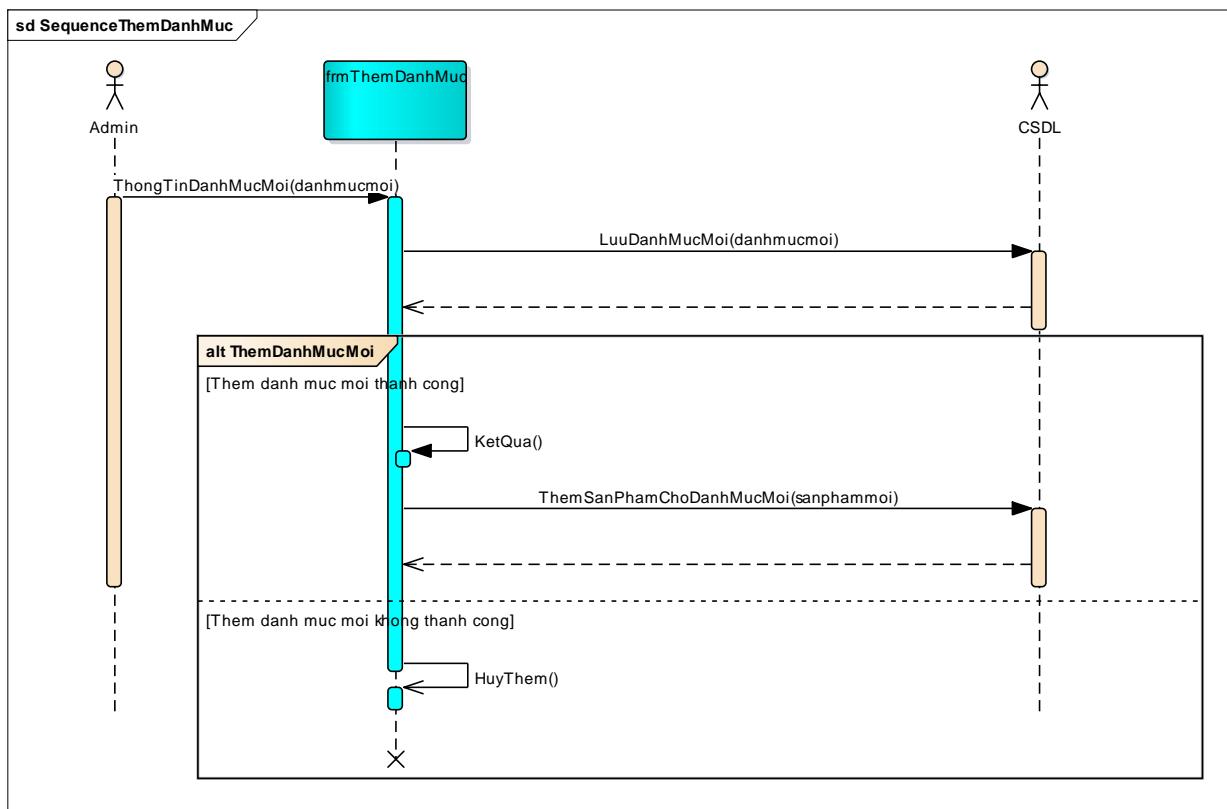
Hình 3. 8: Sequence Đánh giá sản phẩm

3.1.2.8. Sequence Thêm sản phẩm vào giỏ hàng



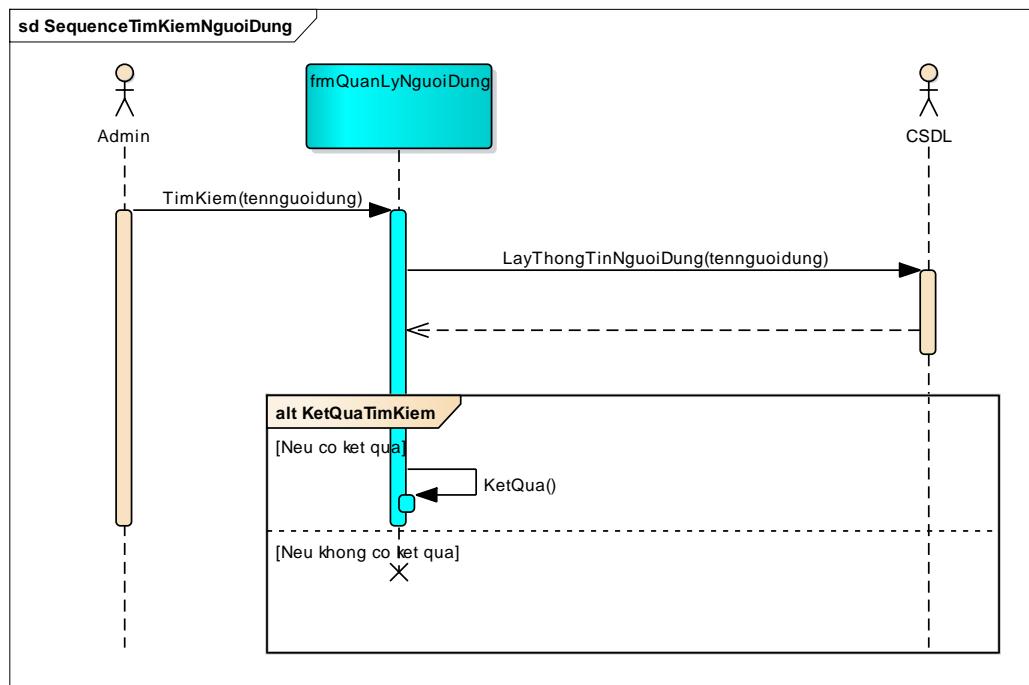
Hình 3. 9: Sequence Thêm sản phẩm vào giỏ hàng

3.1.2.9. Sequence Thêm danh mục mới



Hình 3. 10: Sequence Thêm danh mục mới

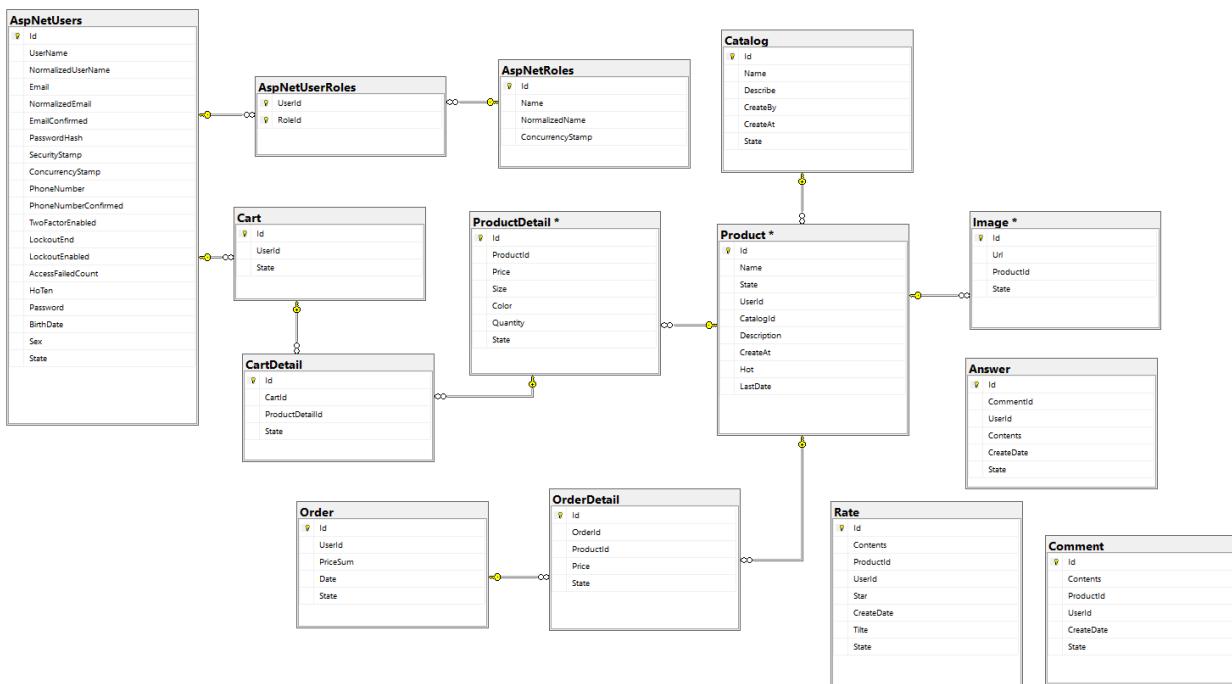
3.1.2.10. Sequence Tìm kiếm người dùng



Hình 3. 11: Sequence Tìm kiếm người dùng

3.2. THIẾT KẾ CƠ SỞ DỮ LIỆU

3.2.1. Mô hình cơ sở dữ liệu



Hình 3. 12: Mô hình cơ sở dữ liệu

3.2.2. Mô tả cơ sở dữ liệu

3.2.2.1. Bảng AspNetUsers

AspNetUsers (Id, UserName, Email, PasswordHash, PhoneNumber, HoTen, Password, BirthDate, Sex, State)

Bảng 3. 1: Chi tiết bảng AspNetUsers

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	nvarchar(450)	Khóa chính	Id	
2	UserName	nvarchar(256)		Số điện thoại	
3	Email	nvarchar(256)		Email người dùng	
4	PasswordHash	nvarchar(MAX)		Password mã hóa	

5	PhoneNumber	nvarchar(MAX)		SĐT	
6	HoTen	nvarchar(MAX)		Họ tên	
7	Password	varchar(100)		Mật khẩu	
8	BirthDate	date		Ngày sinh	
9	Sex	nvarchar(50)		Giới tính	
10	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động

3.2.2.2. *Bảng AspNetRoles*

AspNetRoles (Id, Name)

Bảng 3. 2: Chi tiết bảng AspNetRoles

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	nvarchar(450)	Khóa chính	Id	
2	Name	nvarchar(256)		Tên quyền	

3.2.2.3. *Bảng AspNetUserRoles*

AspNetUserRoles (UserId, RoleId)

Bảng 3. 3: Chi tiết bảng AspNetUserRoles

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	UserId	nvarchar(450)	Khóa chính	Mã người dùng	
2	RoleId	nvarchar(256)	Khóa chính	Mã quyền người dùng	

3.2.2.4. *Bảng Cart*

Cart (Id, UserId, State)

Bảng 3. 4: Chi tiết bảng Cart

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	UserId	nvarchar(450)	Khóa ngoại	Mã người dùng	
3	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động

3.2.2.5. Bảng CartDetail

CartDetail (Id, CartId, ProductId, State)

Bảng 3. 5: Chi tiết bảng CartDetail

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	CartId	int	Khóa ngoại	Mã giỏ hàng	
3	ProductDetailId	int	Khóa ngoại	Mã chi tiết sản phẩm	
4	State	int		Trạng thái	0: Hết hàng 1: Còn hàng 2: Đã thanh toán

3.2.2.6. Bảng Catalog

Catalog (Id, Name, Describe, CreateBy, CreateAt, State)

Bảng 3. 6: Chi tiết bảng Catalog

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	Name	nvarchar(2000)		Tên danh mục	
3	Describe	nvarchar(2000)		Mô tả danh mục	
4	CreateBy	nvarchar(450)	Khóa ngoại	Người tạo	

5	CreateAt	date		Ngày tạo	
6	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động

3.2.2.7. *Bảng Comment*

Comment (Id, Contents, ProductId, UserId, CreateDate, State)

Bảng 3. 7: Chi tiết bảng Comment

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	Contents	nvarchar(500)		Nội dung	
3	ProductId	int	Khóa ngoại	Mã sản phẩm	
4	UserId	nvarchar(450)	Khóa ngoại	Mã người dùng	
5	CreateDate	date		Ngày bình luận	
6	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động

3.2.2.8. *Bảng Image*

Image (Id, Url, ProductId, State)

Bảng 3. 8: Chi tiết bảng Image

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	Url	nvarchar(1000)		Đường dẫn hình ảnh	
3	ProductId	int	Khóa ngoại	Mã sản phẩm	
4	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động

3.2.2.9. *Bảng Order*

Order (Id, UserId, PriceSum, Date, State)

Bảng 3. 9: Chi tiết bảng Order

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	UserId	nvarchar(450)	Khóa ngoại	Mã người dùng	
3	PriceSum	float		Tổng giá tiền của đơn hàng	
4	Date	date		Ngày tạo đơn hàng	
5	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động

3.2.2.10. *Bảng OrderDetail*

OrderDetail (Id, OrderId, ProductId, Price, State)

Bảng 3. 10: Chi tiết bảng OrderDetail

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	OrderId	int	Khóa ngoại	Mã đơn hàng	
3	ProductId	int	Khóa ngoại	Mã sản phẩm	
4	Price	float		Giá sản phẩm lúc đơn hàng được tạo	Giá của sản phẩm sau khi giảm giá
5	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động

3.2.2.11. *Bảng Product*

Product (Id, Name, State, UserId, CatalogId, Description)

Bảng 3. 11: Chi tiết bảng Product

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	Name	nvarchar(2000)		Tên sản phẩm	
3	Description	nvarchar(MAX)		Mô tả sản phẩm	
4	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động
5	UserId	nvarchar(450)	Khóa ngoại	Mã người bán	
6	CatalogId	int	Khóa ngoại	Mã danh mục	
7	CreateDate	datetime		Người tạo	
8	LastModify Date	datetime		Ngày chỉnh sửa	
9	CreateByUser	nvarchar(450)	Khóa ngoại	Người tạo	
10	LastModify ByUser	nvarchar(450)	Khóa ngoại	Người chỉnh sửa	

3.2.2.12. *Bảng ProductDetail*

ProductDetail (Id, ProductId, Price, Size, Color, Quantity, State)

Bảng 3. 12: Chi tiết bảng ProductDetail

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	ProductId	int	Khóa ngoại	Mã sản phẩm	
3	Price	float		Giá sản phẩm	

4	Size	nvarchar(100)		Kích cỡ	
5	Color	nvarchar(50)		Màu sắc	
6	Quantity	int		Số lượng tồn	
7	State	int		Trạng thái	0: Tạm ngưng 1: Còn hàng 2: Hết hàng
8	CreateDate	datetime		Người tạo	
9	LastModifyDate	datetime		Ngày chỉnh sửa	
10	CreateByUser	nvarchar(450)	Khóa ngoại	Người tạo	
11	LastModifyByUser	nvarchar(450)	Khóa ngoại	Người chỉnh sửa	

3.2.2.13. *Bảng Rate*

Rate (Id, Contents, ProductId, UserId, Star, Tilte, State)

Bảng 3. 13: Chi tiết bảng Rate

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	Contents	nvarchar(MAX)		Nội dung	
3	ProductId	int	Khóa ngoại	Mã sản phẩm	
4	UserId	nvarchar(450)	Khóa ngoại	Mã người dùng	
5	Star	int		Số sao đánh giá	
6	CreateDate	date		Ngày đánh giá	
7	Tilte	nvarchar(2000)		Tiêu đề	

9	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động
---	-------	-----	--	------------	------------------------------

3.2.2.14. *Bảng Answer*

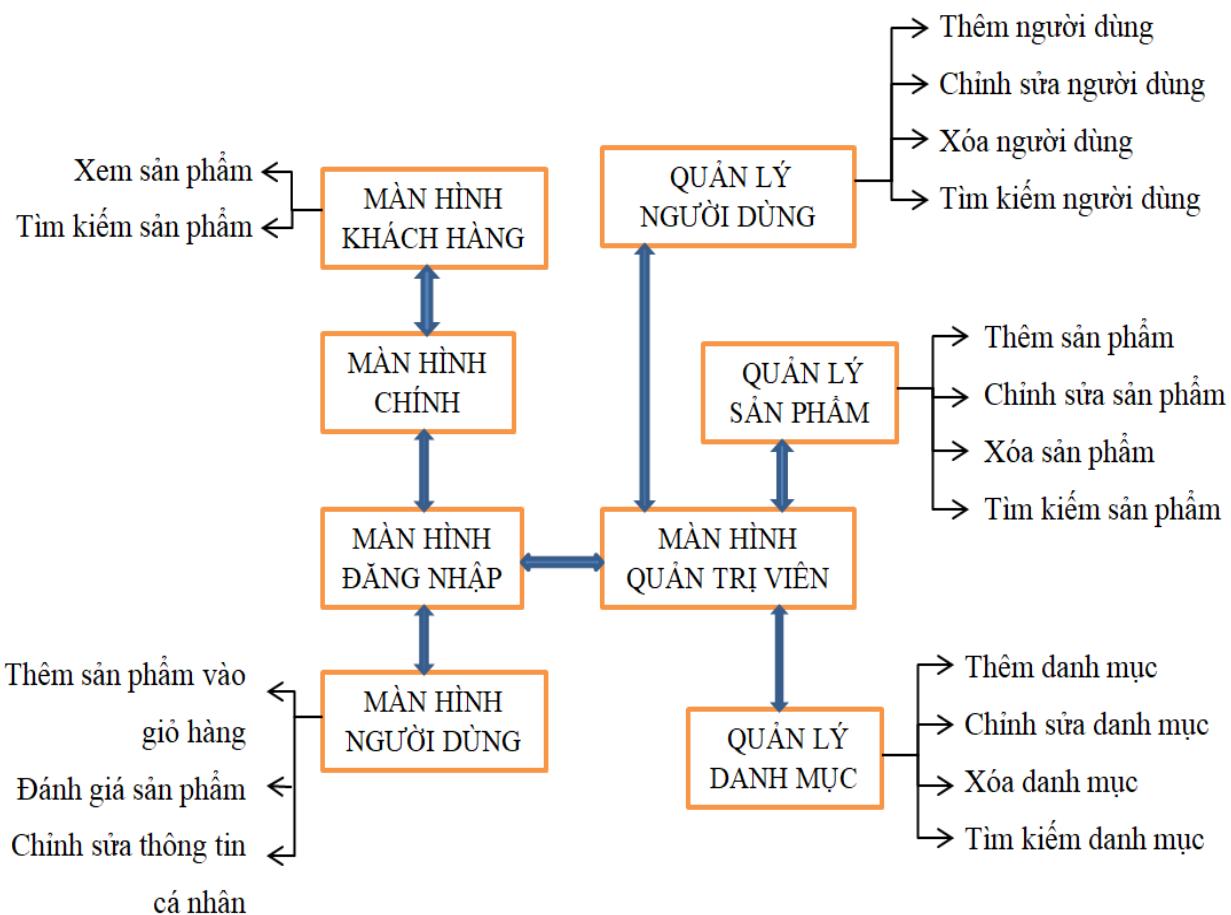
Answer (Id, CommentId, UserId, Contents, CreateDate, State)

Bảng 3. 14: Chi tiết bảng Answer

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	Id	int	Khóa chính	Id	
2	CommentId	int	Khóa ngoại	Mã bình luận	
3	UserId	nvarchar(450)	Khóa ngoại	Mã người dùng	
4	Contents	nvarchar(MAX)		Nội dung trả lời	
5	CreateDate	date		Ngày trả lời	
6	State	int		Trạng thái	0: Tạm ngưng 1: Hoạt động

3.3. THIẾT KẾ GIAO DIỆN

3.3.1. Danh sách màn hình và sơ đồ chuyển đổi

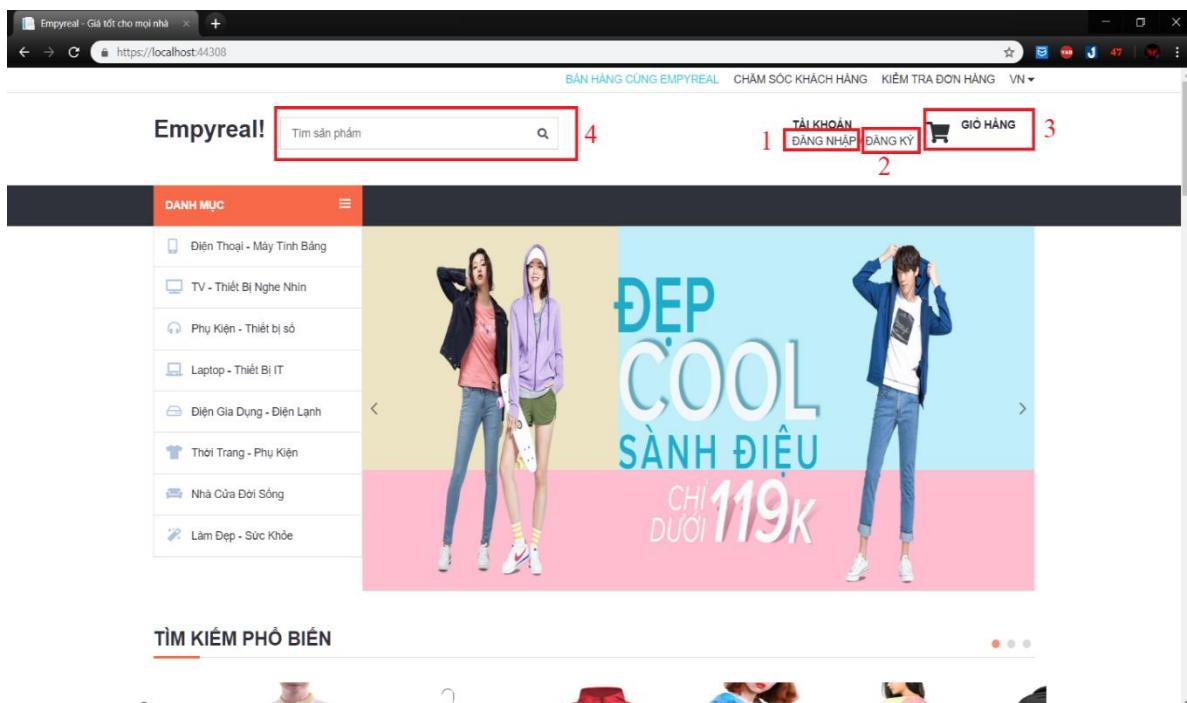


Hình 3. 13: Danh sách màn hình và sơ đồ chuyển đổi

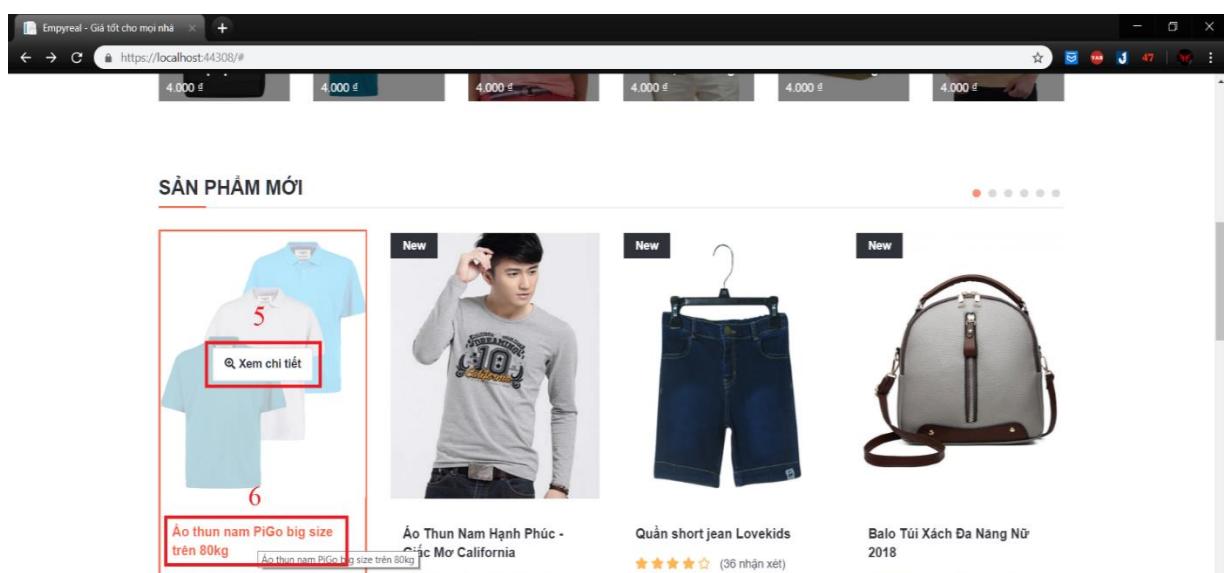
3.3.2. Mô tả chi tiết các màn hình

3.3.2.1. Màn hình Trang chủ

- **Ý nghĩa:** Trang chủ của trang web.
- **Chi tiết màn hình**



Hình 3. 14: Màn hình trang chủ



Hình 3. 15: Màn hình trang chủ

- **Các đối tượng trong màn hình**

Bảng 3. 15: Các đối tượng màn hình Trang chủ

STT	Loại	Ý nghĩa
1	Button	Chuyển đến trang đăng nhập
2	Button	Chuyển đến trang đăng ký
3	Button	Chuyển đến giỏ hàng

4	TextBox	Nhập từ khóa tìm kiếm tên sản phẩm
5	Button	Chuyển đến trang chi tiết sản phẩm của sản phẩm đó
6	Label	Chuyển đến trang chi tiết sản phẩm của sản phẩm đó

3.3.2.2. Màn hình Đăng ký

- **Ý nghĩa:** Cho phép người dùng đăng ký tài khoản vào hệ thống.

- Chi tiết màn hình

Hình 3. 16: Màn hình Đăng ký

- Các đối tượng trong màn hình

Bảng 3. 16: Các đối tượng màn hình Đăng ký

STT	Loại	Ý nghĩa
1	Button	Chuyển đến trang đăng nhập
2	TextBox	Nhập họ tên
3	TextBox	Nhập Sđt
4	TextBox	Nhập Email
5	TextBox	Nhập mật khẩu
6	TextBox	Nhập xác nhận lại mật khẩu
7	RadioButton	Chọn Nam hoặc Nữ

8	Selector	Chọn ngày/tháng/năm sinh
9	Button	Submit Form

3.3.2.3. Màn hình Đăng nhập

- **Ý nghĩa:** Cho phép người dùng đăng nhập vào hệ thống.

- **Chi tiết màn hình**

Hình 3. 17: Màn hình Đăng nhập

- **Các đối tượng trong màn hình**

Bảng 3. 17: Các đối tượng màn hình Đăng nhập

STT	Loại	Ý nghĩa
1	Button	Chuyển đến trang đăng ký
2	TextBox	Nhập Email/ SĐT
3	TextBox	Nhập Mật khẩu
4	Button	Submit Form

3.3.2.4. Màn hình Thông tin cá nhân

- **Ý nghĩa:** Hiển thị thông tin cá nhân của người dùng và cho phép người dùng chỉnh sửa thông tin.

- **Chi tiết màn hình**

Hình 3. 18: Màn hình Thông tin tài khoản

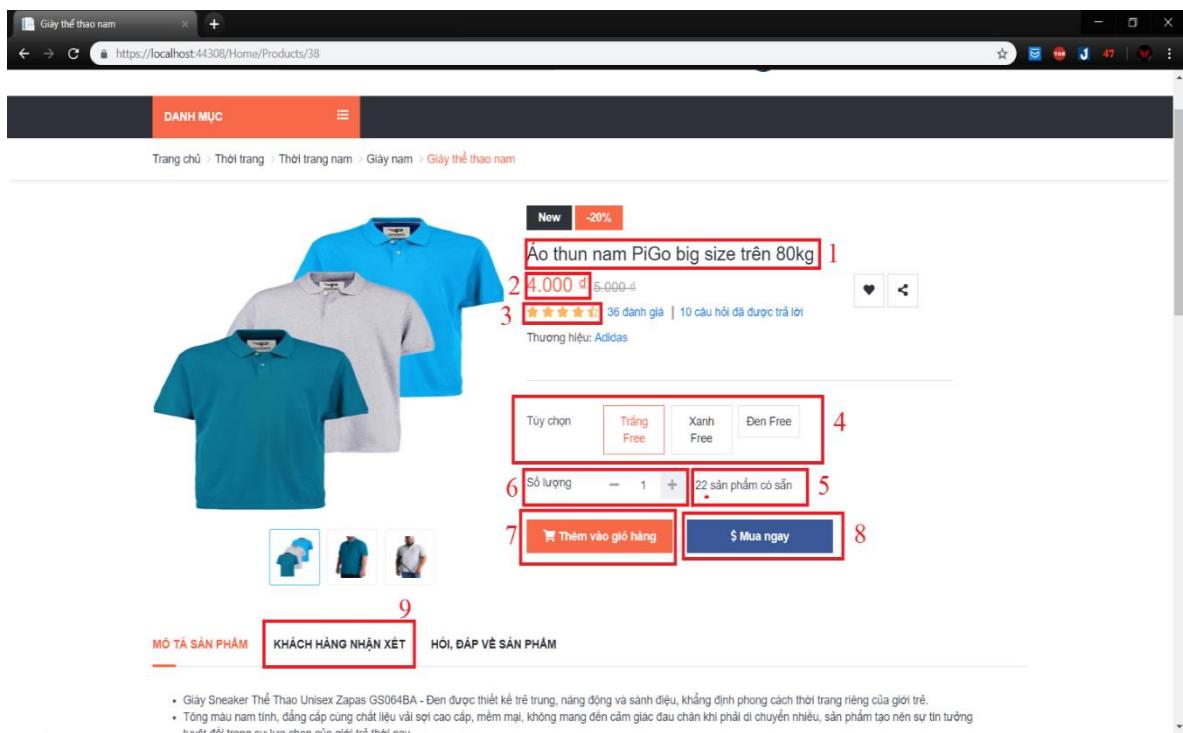
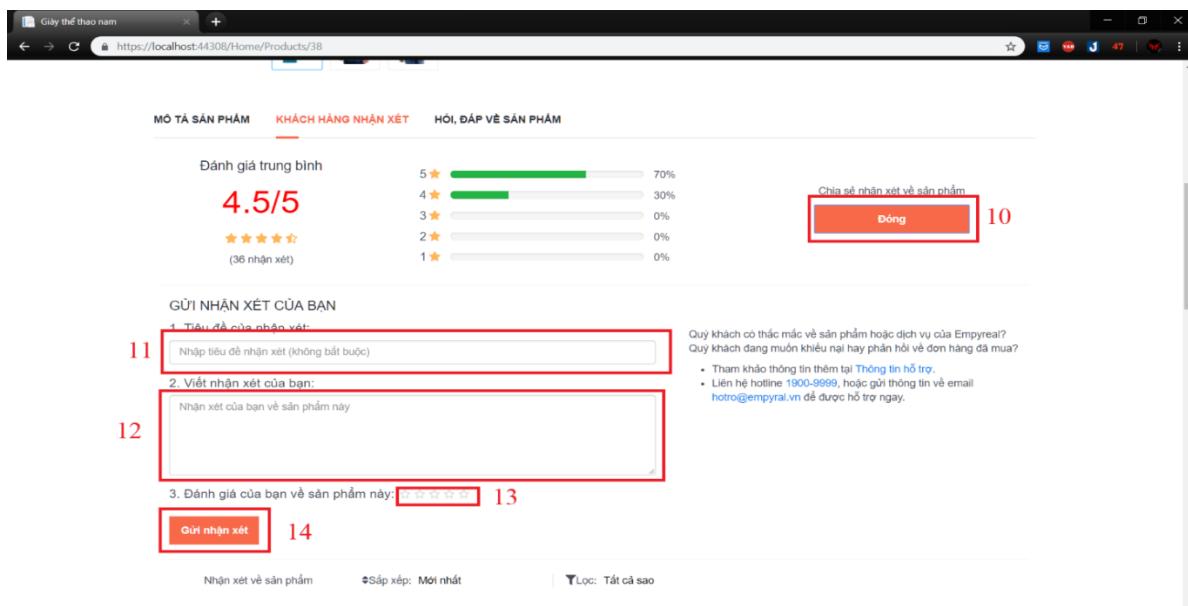
- **Các đối tượng trong màn hình**

Bảng 3. 18: Các đối tượng màn hình Thông tin tài khoản

STT	Loại	Ý nghĩa
1	TextBox	Nhập Họ tên
2	TextBox	Nhập Sđt
3	TextBox	Nhập Email
4	TextBox	Chọn giới tính
5	Selector	Chọn ngày sinh
6	CheckBox	Ẩn/Hiện form đổi mật khẩu
7	TextBox	Nhập lại mật khẩu hiện tại
8	TextBox	Nhập mật khẩu mới
9	TextBox	Xác nhận lại mật khẩu mới
10	Button	Submit form

3.3.2.5. Màn hình Chi tiết sản phẩm

- **Ý nghĩa:** Hiển thị thông tin chi tiết của sản phẩm.
- **Chi tiết màn hình**

**Hình 3. 19: Màn hình Chi tiết sản phẩm****Hình 3. 20: Màn hình Chi tiết sản phẩm**

- **Các đối tượng trong màn hình**

Bảng 3. 19: Các đối tượng màn hình Chi tiết sản phẩm

STT	Loại	Ý nghĩa
1	Label	Tên sản phẩm

2	Label	Giá
3	Image	Đánh giá
4	Button	Chọn loại sản phẩm
5	Label	Sản phẩm còn tồn tại trong kho
6	TextBox	Nhập số lượng sản phẩm
7	Button	Thêm sản phẩm vào giỏ hàng
8	Button	Chuyển đến trang Mua hàng
9	Button	Chuyển sang tab nhận xét
10	Button	Ẩn hiện form nhận xét
11	TextBox	Tiêu đề của nhận xét
12	TextBox	Nội dung nhận xét
13	Image	Đánh giá theo số sao của bạn
14	Button	Submit Form

3.3.2.6. Màn hình Quản lý sản phẩm

- **Ý nghĩa:** Hiển thị danh sách các sản phẩm và cho phép người dùng tìm kiếm sản phẩm.
- **Chi tiết màn hình**

The screenshot shows a web-based application for managing products. On the left, there's a sidebar with icons for Home, Admin, Product Management, and others. The main area has a header with a search bar and navigation links. Below is a table listing products, each with a 'Actions' column containing edit and delete buttons. Red numbers 1 through 5 are overlaid on specific parts of the interface:

- 1**: Points to the sidebar icon for product management.
- 2**: Points to the search bar.
- 3**: Points to the dropdown menu next to the search bar.
- 4**: Points to the 'Thêm mới sản phẩm' (Add new product) button.
- 5**: Points to the bottom navigation bar with page numbers and a 'Lịch sử' (History) link.

Hình 3. 21: Màn hình Quản lý sản phẩm

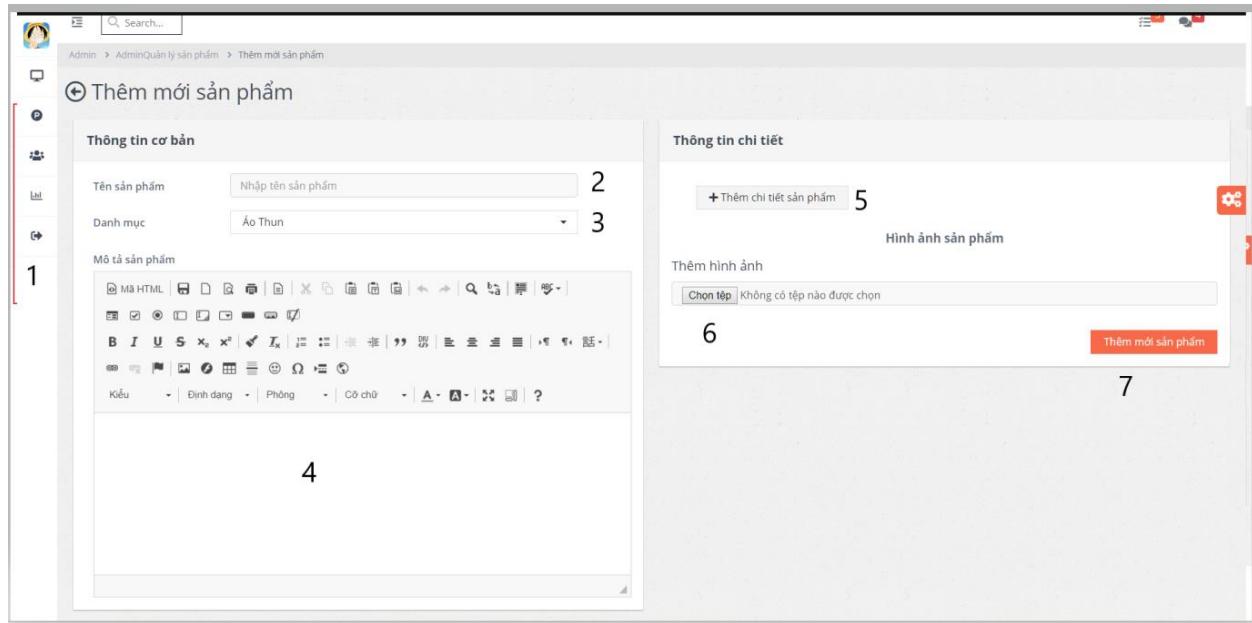
- **Các đối tượng trong màn hình**

Bảng 3. 20: Các đối tượng màn hình Quản lý sản phẩm

STT	Loại	Ý nghĩa
1	Menu	Menu đi tới các trang quản lý khác.
2	Text	Tìm kiếm sản phẩm theo tên hoặc mã.
3	ComboBox	Tìm kiếm sản phẩm theo loại danh mục
4	Button	Đi đến trang Thêm mới sản phẩm
5	Table	Dữ liệu của sản phẩm

3.3.2.7. Màn hình Thêm sản phẩm

- **Ý nghĩa:** Thêm mới một sản phẩm.
- **Chi tiết màn hình**



Hình 3. 22: Màn hình Thêm sản phẩm

Thông tin chi tiết

Kích cỡ	FreeSize	8	Màu sắc	Đen	9	-	10	
Số lượng	1	11	Đơn giá	0	12	(đồng)		

Kích cỡ	FreeSize	Màu sắc	Đen	-
Số lượng	1	Đơn giá	0	

+ Thêm chi tiết sản phẩm

Hình ảnh sản phẩm

Thêm hình ảnh

Chọn tệp Không có tệp nào được chọn

Thêm mới sản phẩm

Hình 3. 23: Màn hình Thêm sản phẩm

- **Các đối tượng trong màn hình**

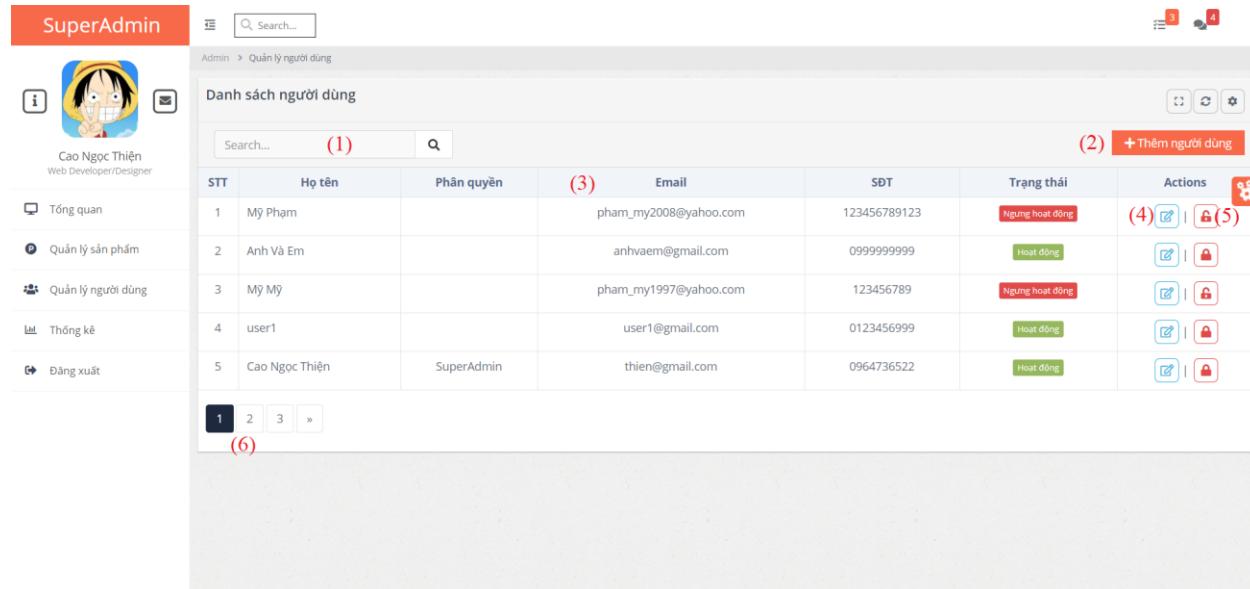
Bảng 3. 21: Các đối tượng màn hình Thêm sản phẩm

STT	Loại	Ý nghĩa
1	Menu	Đi tới các trang quản lý khác
2	Textbox	Tên sản phẩm
3	ComboBox	Chọn danh mục cho sản phẩm
4	TextArea	Mô tả cho sản phẩm
5	Button	Thêm chi tiết cho sản phẩm (hình 3.23)
6	Button	Chọn hình ảnh cho sản phẩm
7	Button	Lưu sản phẩm
8	ComboBox	Chọn kích cỡ cho chi tiết
9	ComboBox	Chọn màu cho chi tiết
10	Button	Xóa chi tiết sản phẩm
11	TextBox	Số lượng tồn của sản phẩm

12	TextBox	Giá của từng chi tiết sản phẩm
----	---------	--------------------------------

3.3.2.8. Màn hình Quản lý người dùng

- **Ý nghĩa:** Hiển thị danh sách tất cả người dùng của hệ thống và cho phép chọn người dùng để chỉnh sửa hoặc xóa.
- **Chi tiết màn hình**



Hình 3. 24: Màn hình Quản lý người dùng

- **Các đối tượng trong màn hình**

Bảng 3. 22: Các đối tượng màn hình Quản lý người dùng

STT	Loại	Ý nghĩa
1	Textbox	Tìm kiếm theo tên người dùng
2	Button	Chuyển đến trang thêm người dùng
3	Table	Hiển thị danh sách tất cả người dùng
4	Button	Chỉnh sửa người dùng tương ứng
5	Button	Xóa người dùng tương ứng
6	Button	Di chuyển tới trang số

3.3.2.9. Màn hình Thêm người dùng

- **Ý nghĩa:** Thêm mới người dùng.

- Chi tiết màn hình

Hình 3. 25: Màn hình Thêm người dùng

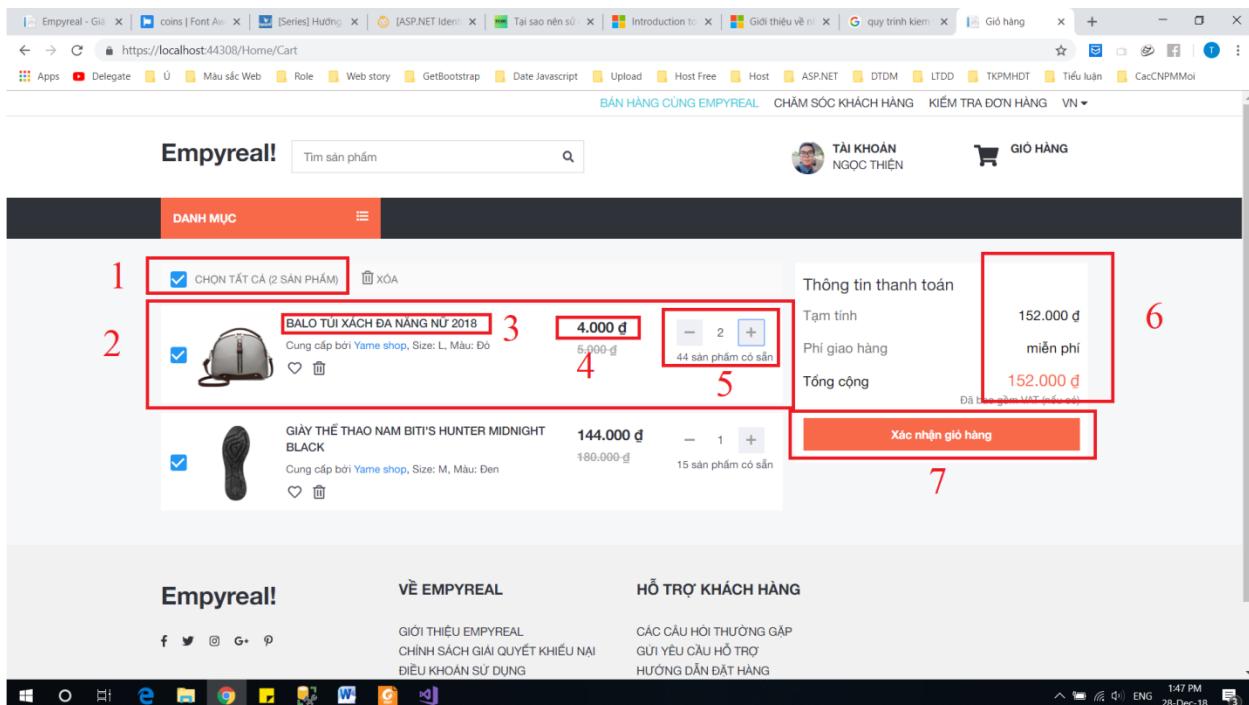
- Các đối tượng trong màn hình

Bảng 3. 23: Các đối tượng màn hình Thêm người dùng

STT	Loại	Ý nghĩa
1	Textbox	Nhập họ tên người dùng
2	Textbox	Nhập số điện thoại
3	Textbox	Nhập email
4	Selector	Chọn quyền cho người dùng mới
5	Radio	Giới tính
6	Selector	Chọn ngày sinh
7	Selector	Chọn tháng sinh
8	Selector	Chọn năm sinh
9	Textbox	Nhập mật khẩu cho người dùng mới
10	Textbox	Xác nhận mật khẩu
11	Button	Xác nhận thêm người dùng

3.3.2.10. Màn hình Giỏ hàng

- **Ý nghĩa:** Xem các sản phẩm trong giỏ hàng và chọn sản phẩm muốn mua để thanh toán.
- **Chi tiết màn hình**



Hình 3. 26: Màn hình Giỏ hàng

- **Các đối tượng trong màn hình**

Bảng 3. 24: Các đối tượng màn hình Giỏ hàng

STT	Loại	Ý nghĩa
1	Checkbox	Chọn tất cả sản phẩm trong giỏ hàng
2	Label	Thông tin một sản phẩm
3	Label	Tên sản phẩm
4	Label	Đơn giá của sản phẩm
5	Button	Chọn số lượng sản phẩm
6	Label	Tổng số tiền phải trả
7	Button	Xác nhận giỏ hàng

3.3.2.11. Màn hình Địa chỉ giao hàng

- **Ý nghĩa:** Cho phép người dùng nhập địa chỉ giao hàng.
- **Chi tiết màn hình**

Hình 3. 27: Màn hình Địa chỉ giao hàng
Hình 3. 28: Màn hình Địa chỉ giao hàng

- Các đối tượng trong màn hình**

Bảng 3. 25: Các đối tượng màn hình Địa chỉ giao hàng

STT	Loại	Ý nghĩa
1	Label	Thông tin giao hàng lần trước
2	Button	Chọn địa chỉ giao hàng lần trước
3	Button	Chỉnh sửa địa chỉ giao hàng lần trước
4	Button	Thêm địa chỉ giao hàng mới
5	Textbox	Nhập họ tên người nhận hàng

6	Textbox	Nhập số điện thoại người nhận hàng
7	Combobox	Chọn Tỉnh/Thành phố giao hàng
8	Combobox	Chọn Quận/Huyện giao hàng
9	Combobox	Chọn Phường/Xã giao hàng
10	Textarea	Nhập địa chỉ giao hàng
11	Radio	Chọn loại địa chỉ giao hàng
12	Button	Xác nhận giao đến địa chỉ đã nhập

3.3.2.12. Màn hình Hoàn tất đơn đặt hàng

- **Ý nghĩa:** Cho phép người dùng xem lại địa chỉ giao hàng, xem lại sản phẩm muốn mua và tiến hành đặt hàng.
- **Chi tiết màn hình**

Hình 3. 29: Màn hình Hoàn tất đơn đặt hàng

- **Các đối tượng trong màn hình**

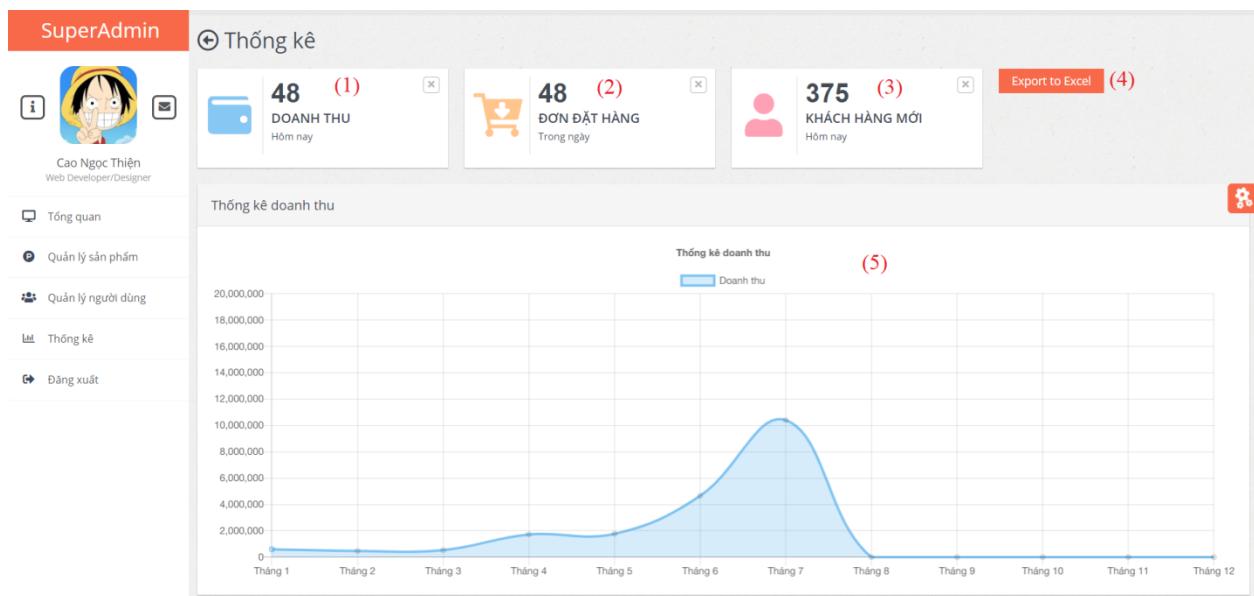
Bảng 3. 26: Các đối tượng màn hình Hoàn tất đơn đặt hàng

STT	Loại	Ý nghĩa
1	Radio	Hình thức giao hàng
2	Radio	Hình thức thanh toán
3	Label	Thông tin địa chỉ giao hàng
4	Button	Chuyển đến trang Địa chỉ giao hàng để chỉnh sửa địa chỉ

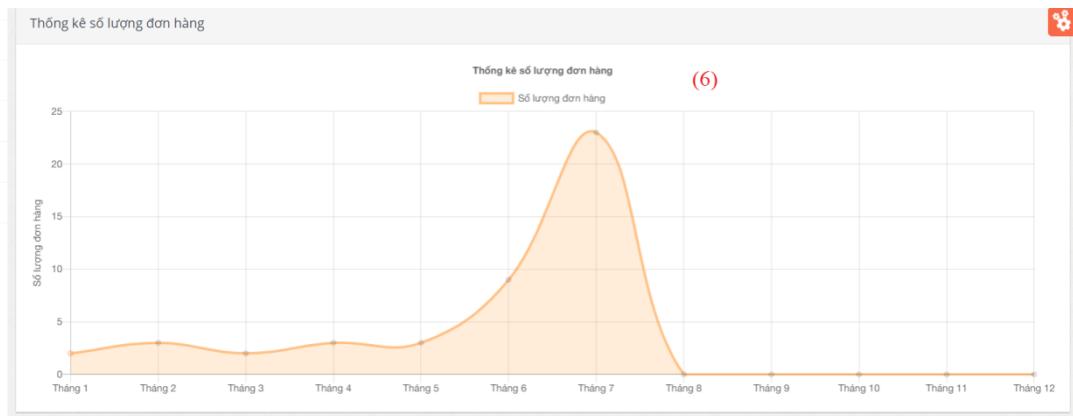
5	Label	Thông tin sản phẩm muốn mua
6	Button	Chuyển đến Giỏ hàng để thay đổi sản phẩm muốn mua
7	Label	Tổng tiền phải trả
8	Button	Hoàn tất đơn đặt hàng

3.3.2.13. Màn hình Thống kê

- **Ý nghĩa:** Cho phép người dùng xem thống kê doanh thu, đơn đặt hàng, khách hàng mới theo từng ngày và theo tháng.
- **Chi tiết màn hình**



Hình 3. 30: Màn hình Thống kê



Hình 3. 31: Màn hình Thống kê

- **Các đối tượng trong màn hình**

Bảng 3. 27: Các đối tượng màn hình Thông kê

STT	Loại	Ý nghĩa
1	Label	Thông kê doanh thu trong ngày
2	Label	Thông kê đơn đặt hàng trong ngày
3	Label	Thông kê khách hàng mới trong ngày
4	Button	Xuất ra file Excel thống kê
5	Chart	Biểu đồ thống kê doanh thu theo tháng
6	Chart	Biểu đồ thống kê số lượng đơn hàng theo tháng

CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ

4.1. CÀI ĐẶT

4.1.1. ASP.Net Core

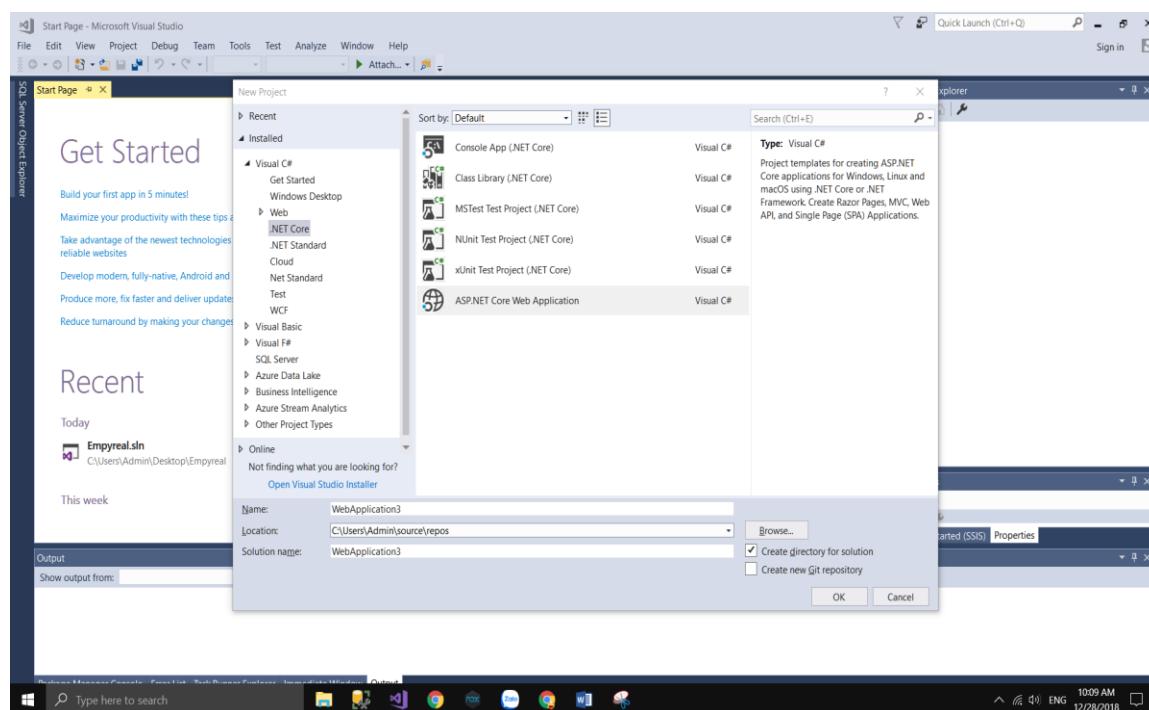
4.1.1.1. Cài đặt

Bảng 4. 1: Cài đặt ASP.NET Core

STT	Tên	Đường dẫn
1	Visual studio 2017	https://visualstudio.microsoft.com/vs/
2	.NET Core SDK	https://dotnet.microsoft.com/download
3	.NET Core Runtime	https://dotnet.microsoft.com/download

4.1.1.2. Khởi tạo dự án ASP.Net Core Web APP

Chọn File => New => Project => Installed => Visual C# => .NET Core => ASP.NET Core WEB Application.

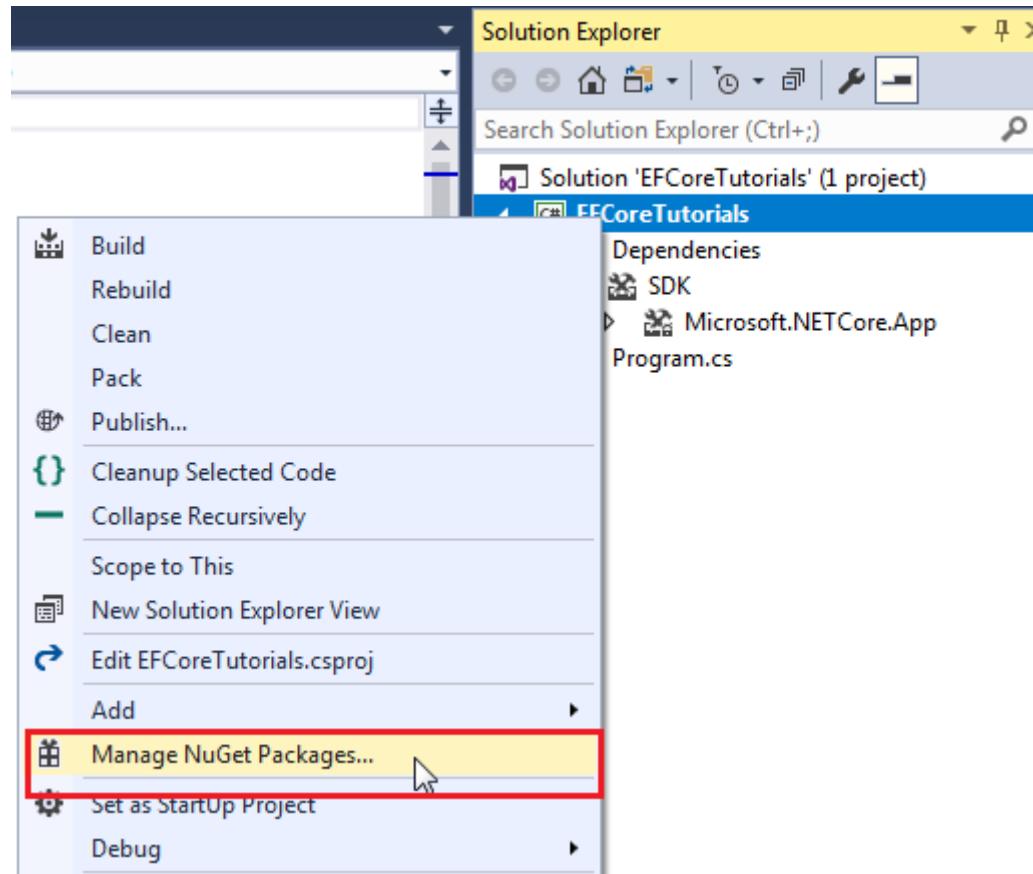


Hình 4. 1: Khởi tạo .NET Core

4.1.2. Entity Framework Core

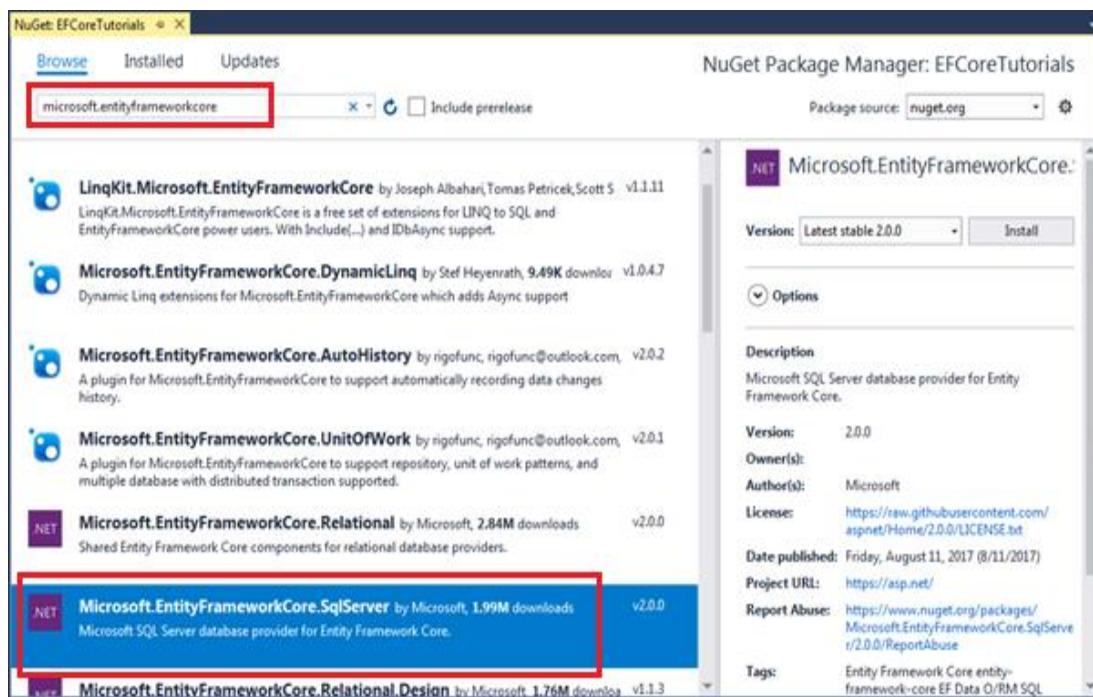
4.1.2.1. Cài đặt

- Cài đặt bằng NuGet Packages.

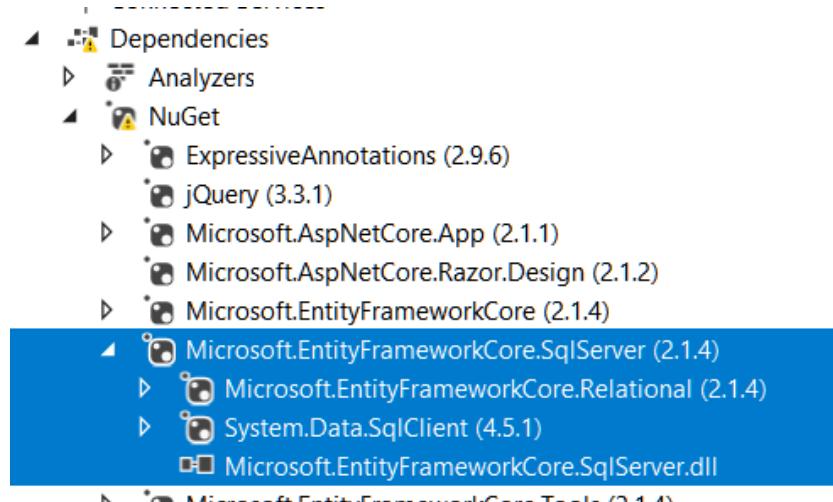


Hình 4. 2: NuGet Packages

- Tìm kiếm từ khóa: *Microsoft.entityframeworkcore*

**Hình 4. 3: Microsoft.entityframeworkcore**

- Chọn Project và nhấn Button Install.

**Hình 4. 4: Cài đặt Entity Framework Core thành công**

4.1.2.2. Sử dụng

Có 2 cách:

- Scaffold-DbContext Command:

- Sử dụng Scaffold-DbContext Command để tạo Model & DbContext từ cơ sở dữ liệu có sẵn trong SqlSever.

- Tool -> NuGet Package Manager -> Package Manager Console và Chạy bằng lệnh:

```
PM> Scaffold-DbContext "Server=.\SQLEXPRESS;Database=SchoolDB;
Trusted_Connection=True;"
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
```

- Server=.\SQLEXPRESS : Địa chỉ vật lý của Sqlsever.
- Database=SchoolDB : Tên cơ sở dữ liệu.
- Thêm –f nếu DbContext đã tồn tại trong Visual Studio.

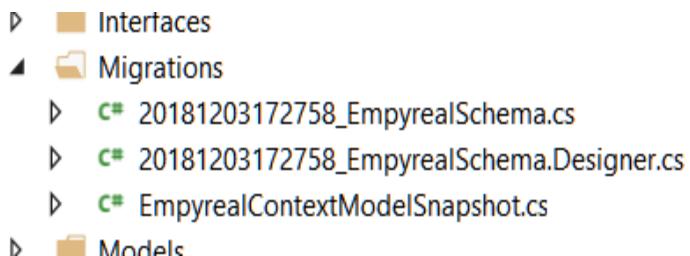
- Migration:

- Migration giúp đồng bộ Cơ sở dữ liệu với EF Core để lưu trữ dữ liệu.
- Truy cập *Package Manager Console* để thực hiện *Migration*:

Tools -> NuGet Package Manager -> Package Manager Console

Bảng 4. 2: Cú pháp Migration

STT	Cú pháp	Nhiệm vụ
1	add-migration <migration name>	Tạo 1 migration theo tên
2	Remove-migration	Xóa migration được thêm cuối cùng
3	Update-database	Cập nhật cơ sở dữ liệu theo migration được thêm cuối cùng



Hình 4. 5: Tạo Migration thành công

4.1.3. Repository

- Bao gồm các phương thức Lấy dữ liệu, thêm/xóa/sửa cơ bản của 1 entity

```

public interface IGenericRepository<T> where T : class
{
    6 references | 0 exceptions
    T Get(Expression<Func<T, bool>> predicate);
    9 references | 0 exceptions
    IEnumerable<T> GetAll();
    4 references | 0 exceptions
    Task Add(T entity);
    1 reference | 0 exceptions
    Task Update(T entity);
    1 reference | 0 exceptions
    Task Delete(T entity);
}

```

Hình 4. 6: Repository

4.1.4. Unit Of Work

- Bao gồm tất cả Services của các Entity

```

2 references | 0 exceptions
IGenericRepository<Cart> CartRepository { get; }
2 references | 0 exceptions
IGenericRepository<CartDetail> CartDetailRepository { get; }
2 references | 0 exceptions
IGenericRepository<Catalog> CatalogRepository { get; }
2 references | 0 exceptions
IGenericRepository<Order> OrderRepository { get; }
2 references | 0 exceptions
IGenericRepository<OrderDetail> OrderDetailRepository { get; }
7 references | 0 exceptions
IGenericRepository<Product> ProductRepository { get; }
2 references | 0 exceptions
IGenericRepository<User> UserRepository { get; }
7 references | 0 exceptions
IGenericRepository<ProductDetail> ProductDetailRepository { get; }
6 references | 0 exceptions
IGenericRepository<Image> ImageRepository { get; }
2 references | 0 exceptions
IGenericRepository<Comment> CommentRepository { get; }
4 references | 0 exceptions
IGenericRepository<Rate> RateRepository { get; }
2 references | 0 exceptions
IGenericRepository<CommentSub> CommentSubRepository { get; }

```

Hình 4. 7: Unit Of Work

4.1.5. Dependency Injection

- Thêm Denpency Injection vào Statup.cs

```
/* Denpency Injection*/
services.AddSingleton<IUnitOfWork, UnitOfWork>();

// Others Services
services.AddSingleton<ICatalogService, CatalogService>();
services.AddSingleton<IUserService, UserService>();
services.AddSingleton<IProductService, ProductService>();
services.AddSingleton<IProductDetailService, ProductDetailService>();
services.AddSingleton<IImageService, ImageService>();
services.AddSingleton<IProductTypeService, ProductTypeService>();
services.AddSingleton<IProductPriceService, ProductPriceService>();
services.AddSingleton<IHistoryService, HistoryService>();
services.AddSingleton<ICartService, CartService>();
services.AddSingleton<ICartDetailService, CartDetailService>();
services.AddSingleton<IRateService, RateService>();
services.AddSingleton<ICommonService, CommonService>();
services.AddSingleton<IProvinceService, ProvinceService>();
services.AddSingleton<IDistrictService, DistrictService>();
services.AddSingleton<IWardService, WardService>();
services.AddSingleton<IOrderService, OrderService>();
services.AddSingleton<IOrderDetailService, OrderDetailService>();
services.AddSingleton<IStatisticalService, StatisticalService>();
```

Hình 4. 8: Dependency Injection

4.2. KIỂM THỬ

4.2.1. Kế hoạch kiểm thử

Nhận dạng chiến lược để kiểm tra và đảm bảo sản phẩm thỏa mãn đặc tả thiết kế, yêu cầu của phần mềm cũng như của khách hàng. Ngoài ra nhận dạng được tính chất, mức độ rủi ro hay hạn chế để chọn phương pháp kiểm thử hợp lý.

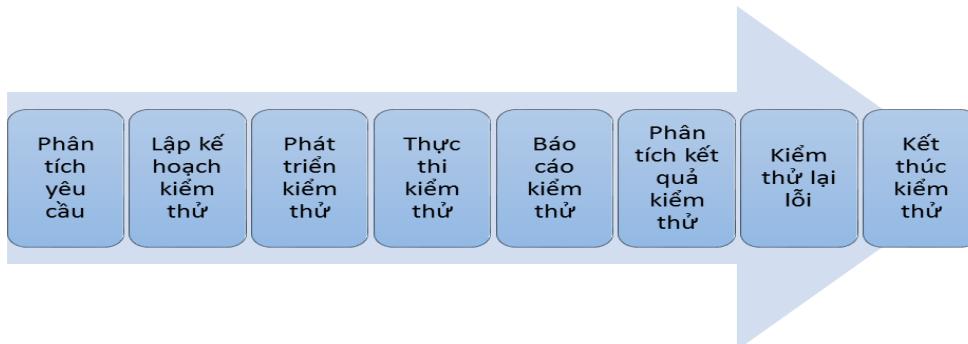
Nhóm sẽ dựa vào các testcase đã được viết để phân chia công việc giữa các thành viên. Và quá trình kiểm thử sẽ dựa vào mẫu excel có sẵn, người tester sẽ điền tên các testcase vào, quy trình thực hiện và kết quả nhận được là pass hay fail.

Chuẩn bị kế hoạch theo dõi, báo cáo, sửa lỗi khi kiểm thử không đạt yêu cầu. Với kết quả nhận được, nhóm sẽ phân chia lỗi ở các cấp độ nghiêm trọng nhất định và phân chia công việc cho các thành viên tiến hành sửa lỗi. Các lỗi nghiêm trọng sẽ được ưu

tiên sửa trước. Và quá trình sẽ lặp đi lặp lại cho đến khi chương trình hoàn thiện, không còn lỗi.

4.2.2. Quy trình kiểm thử

Quy trình kiểm thử phần mềm sẽ thực hiện theo trình tự 8 bước như sau:



Hình 4. 9: Quy trình kiểm thử

4.2.2.1. Phân tích yêu cầu - Requirement Analysis

Đọc thật kỹ và nắm rõ các yêu cầu đặt ra đối với phần mềm và chuẩn bị lập kế hoạch kiểm thử để đảm bảo rằng các testcase sẽ bao trùm các trường hợp lỗi sẽ xảy ra.

4.2.2.2. Lập kế hoạch kiểm thử - Test Planning

Xác định phạm vi kiểm thử, chiến lược kiểm thử. Sẽ sử dụng loại test nào? Test ở phạm vi, vấn đề nào? Như thế nào là đạt? Là lỗi? Ngày nào bắt đầu kiểm thử? Nhận dạng, xác định các rủi ro trong quá trình kiểm thử và kế hoạch đối phó.

4.2.2.3. Phát triển kiểm thử - Testcase Development

Thiết kế danh sách testcase từ các yêu cầu chức năng và phi chức năng theo như kế hoạch kiểm thử đã được làm trước đó. Đảm bảo testcase bao phủ mọi yêu cầu.

Danh sách testcase sẽ được kiểm định và đánh giá lại bởi nhà phân tích nghiệp vụ (Business Analyst). Các đánh giá sẽ được trả về lại cho nhóm tester để tiến hành điều chỉnh, bổ sung nếu cần thiết. Nhóm tester sẽ tiến hành điều chỉnh các testcase.

4.2.2.4. Thực thi kiểm thử - Test Execution

Khi các testcase đã được phê duyệt, phần mềm đã đáp ứng đầy đủ thì tiến hành thực hiện kiểm thử theo testcase đã làm trước đó. Lưu lại kết quả trong quá trình thực thi kiểm thử.

4.2.2.5. Báo cáo kiểm thử - Test Report

Các tester chạy từng testcase, khi gặp một testcase không pass thì phải ghi chú lại. Với mỗi trường hợp test thất bại, ngoài việc ghi chép dữ liệu, tester phải chụp ảnh màn hình để làm bằng chứng lỗi.

Viết báo cáo các lỗi tìm được, tạo bảng tổng kết đánh giá hoạt động kiểm lỗi. Xác định ở mỗi testcase có đạt tiêu chí thành công hay không? Hoàn thành hay chưa?

Tester phải liên tục duy trì báo cáo tài liệu lưu vết trong quá trình thực hiện kiểm thử.

4.2.2.6. Phân tích kết quả kiểm thử - Test Result Analysis

Phân tích kết quả của báo cáo kiểm thử. Nếu có testcase nào lỗi thì tiến hành sửa lỗi. Có thể tiến hành chỉnh sửa hoặc bổ sung testcase.

4.2.2.7. Kiểm thử lại lỗi – Repeat Testing

Tiến hành thực hiện lại kiểm thử đối với các lỗi đã tìm ra trước đó hoặc cái testcase mới phát sinh. Nếu lỗi thì lại thực hiện kiểm thử lại cho đến khi không tìm thấy lỗi nữa.

4.2.2.8. Kết thúc kiểm thử - Finish Testing

Khi tất cả testcase đều đã pass, không còn testcase nào phát sinh, thỏa mãn được yêu cầu của phần mềm. Thì quá trình kiểm thử kết thúc.

PHẦN KẾT LUẬN

1. KẾT QUẢ ĐẠT ĐƯỢC

Sau thời gian nghiên cứu và tiến hành ứng dụng lý thuyết đã nghiên cứu, nhóm đã xây dựng được một Website bán hàng với mục tiêu: Hoàn thành tốt các chức năng được đề ra, hiểu về lý thuyết, vận hành được website bằng công nghệ .Net Core và xử lý được các nghiệp vụ của một website bán hàng.

2. ƯU ĐIỂM

- Website quản lý và thực thi được quá trình mua-bán giữa doanh nghiệp với người.
- Giao diện website đẹp, dễ sử dụng.
- Người dùng có thể đăng nhập, đăng ký dễ dàng và nhanh chóng.
- Các chức năng xử lý đơn giản, chia thành từng phần nhỏ nên dễ dàng quản lý và bảo trì.
- Website thực thi nhanh.

3. NHƯỢC ĐIỂM

- Chưa phát huy hết sức mạnh của thời gian thực (realtime).
- Việc quản lý người dùng và sản phẩm cần được phát triển: nhu cầu bảo mật, kiểm soát được số lượng hàng tồn, các thống kê thông minh để người dùng có thể quản lý tài sản của mình tốt hơn.

4. HƯỚNG PHÁT TRIỂN

- Website bán hàng và công nghệ .Net Core là một bước tiến rất lớn trên con đường sự nghiệp của chúng em. Chúng em đã học hỏi rất nhiều thứ về kĩ thuật cũng như nghiệp vụ khi thiết kế, xây dựng và phát triển website này. Công nghệ ASP.Net Core rất “thông minh”, rất hiện đại và đầy sự thú vị.
- Trong tương lai, nhóm sẽ tối đa hóa khả năng thời gian thực vào website để sử dụng các chức năng thông báo, trò chuyện. ngoài ra nhóm cũng sẽ phát triển và tối ưu hóa chức năng hiện tại để website hoạt động nhanh và tốt nhất có thể. Ngoài ra, nhu cầu bảo mật về dữ liệu cũng rất quan trọng, nhóm sẽ song song phát triển vấn đề này cùng với những thứ ở trên.

- Trong quá trình tìm hiểu, nhóm cũng đã gặp rất nhiều khó khăn, nhưng nhờ vào Thầy Vinh và các bạn khác nên nhóm cũng có thể hoàn thành đề tài này. Trải qua tất cả, “Lửa thử vàng. Gian nan thử sức”, nhóm chúng em rất vui và tự hào về sản phẩm này.
- Nhóm chân thành cảm ơn Thầy Lê Văn Vinh và các bạn đã giúp đỡ nhóm trong thời gian qua.

TÀI LIỆU THAM KHẢO

1. [Asp.net Core những thay đổi lớn cho lập trình viên .NET](#)
2. [\[Giới thiệu .NET core và ASP.NET core\]Phần 2:Phân biệt .NET Framework, .NET Core và Mono](#)
3. [About .NET Core, Microsoft](#)
4. [Entity Framework Core 2.0: Những điều bên trong bộ mã nguồn mở EF Core 2.0](#)
5. [Entity Framework Tutorial: Entity Framework Core](#)
6. [ASP .NET MVC - Phần 1: Tao ứng dụng ASP .NET MVC đầu tiên](#)
7. [Tổng quan về ASP.NET MVC](#)
8. [ASP.NET MVC – Bảng Razor Syntax Reference \(gồm Razor v2\)](#)
9. [DEPENDENCY INJECTION VÀ INVERSION OF CONTROL – Phần 1,2,3](#)
10. [Repository Pattern và Unit of Work với Entity Framework trong ASP.NET MVC](#)
11. [Repository và Unit of Work Pattern](#)
12. [\[Series\] Hướng dẫn sử dụng ASP.NET Identity \(phần 1\)](#)