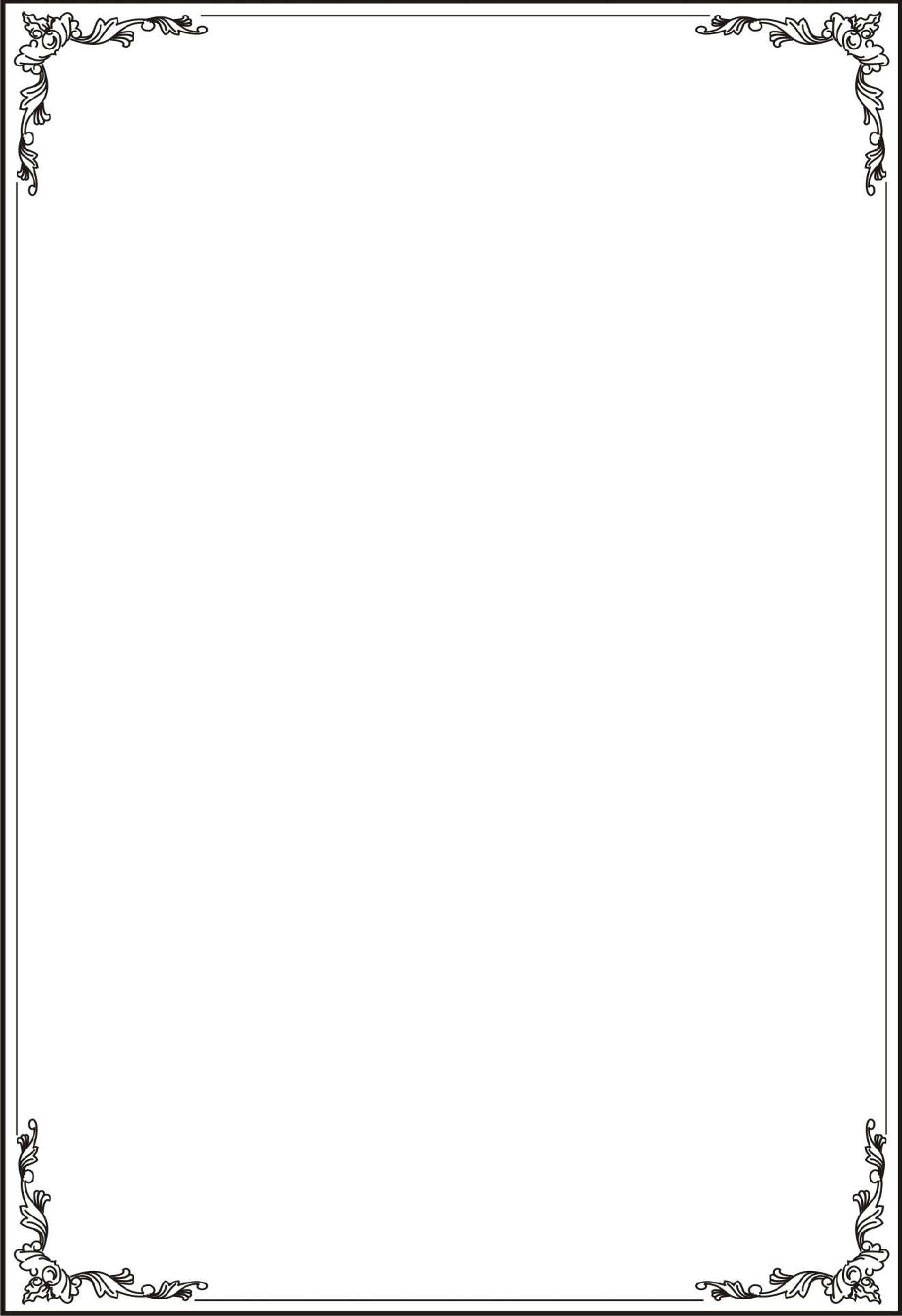
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**



**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**---o0o---**



**BÁO CÁO TIỂU LUẬN**

**CHUYÊN NGÀNH**

**TÌM HIỂU ANGULAR, ASP .NETCORE VÀ XÂY DỰNG TRANG WEB THƯƠNG MẠI ĐIỆN TỬ**

**GVHD : ThS.Nguyễn Trần Thi Văn**

**SVTH : Nguyễn Trọng Đạt 16110049**

**Nguyễn Duy Lâm 16110134**

**Tp. Hồ Chí Minh, tháng 12 năm 2019**

**NHIỆM VỤ THỰC HIỆN TIỂU LUẬN CHUYÊN NGÀNH**

Họ tên SV: Nguyễn Trọng Đạt ….…………MSSV: 16110049 …..

Họ tên SV: Nguyễn Duy Lâm ……………..MSSV: 16110134 …..

Chuyên ngành: Công nghệ phần mềm

Tên đề tài: Tìm hiểu Angular và .Netcore và xây dựng trang web thương mại điện tử.

**Nội dung thực hiện:**

*Lý thuyết:*

* Tổng quan Angular
* Định nghĩa
* Cấu trúc
* Nhu cầu,mục đích sử dụng
* Ứng dụng
* Tổng quan .NetCore
* Định nghĩa
* Cấu trúc
* Nhu cầu,mục đích sử dụng
* Ứng dụng
* Phát triển thêm Deploy bằng Docker
* *Thực hành:*
* Chức năng Web:

+ Quy trình mua sản phẩm.

+ CRUD.

+ Xem danh sách sản phẩm.

+ Quản lý sản phẩm.

+ Quản lý Account.

+….. ( bổ sung thêm chức năng )

**Thời gian thực hiện:** 15 tuần (Bắt đầu từ 9/9/2019 đến 14/12/2019)

Chữ ký của SV: ……………………………………………………………

Chữ ký của SV: ……………………………………………………………

*TP.HCM, Ngày…. tháng…. năm …*

TRƯỞNG KHOA CNTT GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên) (Ký và ghi rõ họ tên)

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên: Nguyễn Trọng Đạt …………………..MSSV: 16110049

Họ và tên Sinh viên: Nguyễn Duy Lâm …………………..MSSV: 16110134

Chuyên ngành: Công nghệ Thông tin

Tên đề tài: Tìm hiểu Angular và .Netcore và xây dựng trang web thương mại điện tử.

Họ và tên Giáo viên hướng dẫn: ThS. Nguyễn Trần Thi Văn

**NHẬN XÉT**

1. Về nội dung đề tài & 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.HCM, ngày.....tháng….năm 2019

Giáo viên hướng dẫn

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

Họ và tên Sinh viên: Nguyễn Trọng Đạt MSSV: 16110049

Họ và tên Sinh viên: Nguyễn Duy Lâm MSSV: 16110134

Ngành: Công nghệ Thông tin

Tên đề tài: Tìm hiểu Angular và .Netcore và xây dựng trang web thương mại điện tử.

Họ và tên Giáo viên hướng dẫn: ThS. Nguyễn Trần Thi Văn

**NHẬN XÉT**

1. Về nội dung đề tài & 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.HCM, ngày 9 tháng 12 năm 2019

Giáo viên phản biện

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn khoa Đào tạo Chất lượng cao, Ngành Công nghệ thông tin, chuyên ngành Công nghệ phầm mềm, trường Đại học Sư Phạm Kỹ Thuật Tp.HCM đã tạo điều kiện thuận lợi cho nhóm em thực hiện đề tài này.

Chúng em cũng xin được gửi lời cảm ơn chân thành nhất đến thầy **Nguyễn Trần Thi Văn**, người đã tận tình chỉ bảo, định hướng đi và hướng dẫn nhóm em thực hiện đề tài này.

Bên cạnh đó, nhóm em xin gửi lời cảm ơn đến các thầy cô, những người đã giảng dạy, tạo điều kiện cho em tích lũy được những kiến thức quý báu trong những năm học qua.

Dù đã cố gắng hoàn thành đề tài tiểu luận đúng yêu cầu, nhưng do thời gian hạn hẹp và khả năng còn hạn chế nên chắc chắn sẽ có những thiếu sót không tránh khỏi. Nhóm em mong nhận được sự thông cảm và tận tình chỉ bảo của các thầy cô.

TP. Hồ Chí Minh, tháng 12 năm 2019

**Nhóm sinh viên thực hiện**

Nguyễn Trọng Đạt

Nguyễn Duy Lâm

**DANH SÁCH CÁC HÌNH**

**Hình 1:** Kích thước Bundle của ứng dụng Hello World với Angular Ivy và không có Ivy.

**Hình 2:** Vị trí .NET Core trong mô hình phát triển .NET.

**Hình 3:** Mô hình MVVM.

**Hình 4:** Workflow MVVM.

**Hình 5:** Mô hình docker.

**Hình 6:** Workflow Docker.

**Hình 7:** Use case diagram.

**Hình 8:** Mô hình cơ sở dữ liệu.

**Hình 9:** Màn hình quản lý Tài Khoản.

**Hình 10:** Màn hình Quản lý Menu.

**Hình 11:** Màn hình quản lý loại MetaDataType.

**Hình 12:** Màn hình quản lý MetaDataValue.

**Hình 13:** Màn hình quản lý sản phẩm.

**Hình 14:** Màn hình quản lý Khuyến Mãi.

**Hình 15:** Màn hinh hiển thị sản phẩm client.

**Hình 16:** Màn hình giỏ hàng.

**Hình 17:** Màn hình hiển thị quy trình giao hàng bước 1.

**Hình 18:** Màn hình hiển thị quy trình giao hàng bước 2.

**DANH SÁCH CÁC BẢNG**

**Bảng 1:** Nhận diện tác nhân và các chức năng trong sơ đồ use case.

**Bảng 2:** Mô tả các Actor.

**Bảng 3:** Mô tả Use case diagram.

**Bảng 4:** Đặt tả use case đăng kí.

**Bảng 5:** Đặt tả use case đăng nhập.

**Bảng 6:** Đặt tả use case Thêm sản phẩm.

**Bảng 7:** Đặt tả use case Tìm kiếm sản phẩm.

**Bảng 8:** Đặt tả use case Giỏ hàng.

**Bảng 9:** Đặt tả use case Lịch sử mua hàng.

**Bảng 10:** Đặt tả use case Đánh giá sản phẩm.

**Bảng 11:** Đặt tả use case Thanh toán.

**Bảng 12:** Đặt tả use case Thêm tài khoản nhân viên.

**Bảng 13:** Đặt tả use case xem tài khoản nhân viên.

**Bảng 14:** Đặt tả use case xóa tài khoản nhân viên.

**Bảng 15:** Đặt tả use case sửa tài khoản nhân viên.

**Bảng 16:** Đặt tả use case tìm kiếm tài khoản nhân viên.

**Bảng 17:** Đặt tả use case Thêm menu.

**Bảng 18:** Đặt tả use case Xem menu.

**Bảng 19:** Đặt tả use case Xóa menu.

**Bảng 20:** Đặt tả use case Sửa menu.

**Bảng 21:** Đặt tả use case Tìm kiếm menu.

**Bảng 22:** Đặt tả use case Thêm sản phẩm.

**Bảng 23:** Đặt tả use case Xem sản phẩm.

**Bảng 24:** Đặt tả use case Xóa sản phẩm.

**Bảng 25:** Đặt tả use case Sửa sản phẩm.

**Bảng 26:** Đặt tả use case Tìm kiếm sản phẩm.

**Bảng 27:** Đặt tả use case Thêm đánh giá sản phẩm.

**Bảng 28:** Đặt tả use case Xem đánh giá sản phẩm.

**Bảng 29:** Đặt tả use case Xóa đánh giá sản phẩm.

**Bảng 30:** Đặt tả use case Sửa đánh giá sản phẩm.

**Bảng 31:** Đặt tả use case Tìm kiếm đánh giá sản phẩm.

**Bảng 32:** Đặt tả use case Thêm khuyến mãi.

**Bảng 33:** Đặt tả use case Xem khuyến mãi.

**Bảng 34:** Đặt tả use case Xóa khuyến mãi.

**Bảng 35:** Đặt tả use case Sửa khuyến mãi.

**Bảng 36:** Đặt tả use case Tìm kiếm khuyến mãi.

**Bảng 37:** Bảng AuthUsers.

**Bảng 38:** Bảng Products.

**Bảng 39:** Bảng Bills.

**Bảng 40:** Bảng Carts.

**Bảng 41:** Bảng CartDetails.

**Bảng 42:** Bảng DetailBills .

**Bảng 43:** Bảng Menus.

**Bảng 44:** Bảng MetaDataType.

**Bảng 45:** Bảng MetaDataValues.

**Bảng 46:** Bảng Promotions.

**Bảng 47:** Thông tin các đối tượng trên màn hình quản lý tài khoản

**Bảng 48:** Bảng xử lý màn hình quản lý tài khoản

**Bảng 49:** Bảng thông tin các đối tượng trên màn hình quản lý menu

**Bảng 50:** Bảng thiết kế xử lý màn hình quản lý menu

**Bảng 51:** Bảng thông tin các đối tượng trên màn hình quản lý MetaDataType

**Bảng 52:** Bảng thiết kế xử lý màn hình quản lý MetaDataType

**Bảng 53:** Bảng thông tin các đối tượng trên màn hình quản lý MetaDataValue

**Bảng 54:** Bảng thiết kế xử lý màn hình quản lý MetaDataValue

**Bảng 55:** Bảng thông tin các đối tượng trên màn hình quản lý sản phẩm

**Bảng 56:** Bảng thiết kế xử lý màn hình quản lý sản phẩm

**Bảng 57:** Bảng thông tin các đối tượng trên màn hình quản lý khuyến mãi

**Bảng 58:** Bảng thiết kế xử lý màn hình quản lý khuyến mãi

**Bảng 59:** Bảng thông tin các đối tượng trên màn hình hiển thị sản phẩm

**Bảng 60:** Bảng thiết kế xử lý màn hình hiển thị sản phẩm

**Bảng 61:** Bảng thông tin các đối tượng trên màn hình giỏ hàng

**Bảng 62:** Bảng thiết kế xử lý màn hình giỏ hàng

**Bảng 63:** Bảng thông tin các đối tượng trên màn hình quy trình giao hàng bước 1

**Bảng 64:** Bảng thiết kế xử lý màn hình quy trình giao hàng bước 1

**Bảng 65:** Bảng thông tin các đối tượng trên màn hình quy trình giao hàng bước 2

**Bảng 66:** Bảng thiết kế xử lý màn hình quy trình giao hàng bước 2

**Bảng 68:** Bảng thông tin các đối tượng trên màn hình quy trình giao hàng thành công

**Bảng 69:** Bảng thiết kế xử lý màn hình quy trình giao hàng thành công

**Bảng 70:** Bảng thông tin các đối tượng trên màn hình quản lý hóa đơn

**Bảng 71:** Bảng thiết kế xử lý màn hình quy trình hóa đơn

**Bảng 72:** Bảng test case giỏ hàng

MỤC LỤC

[PHẦN MỞ ĐẦU 1](#_Toc26771767)

[1. LÝ DO CHỌN ĐỀ TÀI 1](#_Toc26771768)

[2. MỤC TIÊU CỦA ĐỀ TÀI 2](#_Toc26771769)

[3. CÁCH TIẾP CẬN VÀ NGIÊN CỨU 2](#_Toc26771770)

[4. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC 2](#_Toc26771771)

[PHẦN NỘI DUNG 3](#_Toc26771772)

[1. CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 3](#_Toc26771773)

[1.1 ANGULAR CLI 8 3](#_Toc26771774)

[1.2 ANT-DESIGN(NG-ZORRO) 10](#_Toc26771775)

[1.3 TYPESCRIPT 11](#_Toc26771776)

[1.4 ASP .NETCORE 13](#_Toc26771777)

[1.5 MÔ HÌNH MVVM 17](#_Toc26771778)

[1.6 DOCKER 18](#_Toc26771779)

[2. CHƯƠNG 2: XÁC ĐỊNH MÔ HÌNH HÓA YÊU CẦU 25](#_Toc26771780)

[2.1 XÁC ĐỊNH YÊU CẦU 25](#_Toc26771781)

[2.2 THIẾT KẾ USECASE DIAGRAM 26](#_Toc26771782)

[2.3 ĐẶC TẢ USECASE 31](#_Toc26771783)

[3. CHƯƠNG 3: THIẾT KẾ CƠ SỞ DỮ LIỆU 55](#_Toc26771784)

[3.1 THIẾT KẾ DỮ LIỆU 55](#_Toc26771785)

[3.2 MÔ TẢ CHI TIẾT CÁC BẢNG 57](#_Toc26771786)

[4. CHƯƠNG 4: THIẾT KẾ GIAO DIỆN VÀ XỬ LÝ 64](#_Toc26771787)

[4.1 MÀN HÌNH QUẢN LÝ TÀI KHOẢN 64](#_Toc26771788)

[4.2 MÀN HÌNH QUẢN LÝ MENU 65](#_Toc26771789)

[4.3 MÀN HÌNH QUẢN LÝ METADATA TYPE 66](#_Toc26771790)

[4.4 MÀN HÌNH QUẢN LÝ METADATA VALUE 67](#_Toc26771791)

[4.5 MÀN HÌNH QUẢN LÝ SẢN PHẨM 68](#_Toc26771792)

[4.6 MÀN HÌNH QUẢN LÝ GIẢM GIÁ 69](#_Toc26771793)

[4.7 MÀN HÌNH HIỂN THỊ SẢN PHẨM 70](#_Toc26771794)

[4.8 MÀN HÌNH GIỎ HÀNG 71](#_Toc26771795)

[4.9 MÀN HÌNH QUY TRÌNH ĐẶT HÀNG BƯỚC 1 72](#_Toc26771796)

[4.10 MÀN HÌNH HIỂN THỊ QUY TRÌNH ĐẶT HÀNG BƯỚC 2 73](#_Toc26771797)

[4.11 MÀN HÌNH QUẢN LÝ GIAO HÀNG THÀNH CÔNG 74](#_Toc26771798)

[4.12 MÀN HÌNH QUẢN LÝ HÓA ĐƠN 75](#_Toc26771799)

[5. CHƯƠNG 5: CÀI ĐẶT VÀ KIỂM THỬ 76](#_Toc26771800)

[5.1 MÔI TRƯỜNG LẬP TRÌNH 76](#_Toc26771801)

[5.2 KIỂM THỬ 76](#_Toc26771802)

[KẾT LUẬN 79](#_Toc26771803)

[1. NHỮNG KẾT QUẢ ĐẠT ĐƯỢC 79](#_Toc26771804)

[2. ƯU ĐIỂM 79](#_Toc26771805)

[3. NHƯỢC ĐIỂM 79](#_Toc26771806)

[TÀI LIỆU THAM KHẢO 80](#_Toc26771807)

# PHẦN MỞ ĐẦU

## LÝ DO CHỌN ĐỀ TÀI

* Với sự phát triển mạnh mẽ của internet, công nghệ thông tin cùng với sự lớn mạnh của ngành thương mại điện tử nên mỗi doanh nghiệp cần có một thiết kế website bán hàng.
* Thu hút thêm nhiều khách hàng: sở hữu một website bán hàng sẽ giúp doanh nghiệp thu hút thêm nhiều khách hàng tiềm năng qua mạng internet.
* Đẩy mạnh hoạt động bán hàng mọi lúc, mọi nơi: Có một website bán hàng sẽ giúp đẩy mạnh hoạt động bán hàng mọi lúc, mọi nơi. Khi có một website thì chỉ cần khách hàng có kết nối internet với các thiết bị truy cập thì khách hàng có thể truy cập vào website bán hàng của bạn mọi nơi vào mọi thời điểm. Khách hàng sẽ không phải đến trực tiếp cửa hàng hay mua hàng theo giờ giấc hoạt động của cửa hàng mà có thể mua hàng bất cứ lúc nào. Điều này giúp tăng tối đa cơ hội bán hàng, giúp thúc đẩy doanh số và thuận tiện cho khách hàng trong việc mua sắm, làm cho họ hài lòng hơn.
* Giảm chi phí kinh doanh: Việc có một thiết kế website bán hàng sẽ giúp bạn tiết kiệm tối đa chi phí cho hoạt động kinh doanh. Khi bạn có một website bán hàng trên mạng, thậm chí bạn không cần phải mở cửa hàng mà vẫn có thể thực hiện được hoạt động mua bán. Điều này sẽ giúp bạn tiết kiệm được một khoản chi phí cho việc thuê cửa hàng, kho bãi cùng chi phí cho nhân lực. Ngoài ra, việc có website bán hàng cũng giúp việc thanh toán của khách hàng thuận tiện và dễ dàng hơn. Khách hàng có thể thanh toán qua thẻ ATM, qua các phương tiện thanh toán điện tử.
* Giúp triển khai các chiến lược marketing hiệu quả: Website sẽ giúp các chiến lược marketing thực hiện có hiệu quả hơn và thu được nguồn doanh thu hiệu quả. Nếu như trước đây có các hình thức marketing thông qua phát tờ rơi, quảng cáo qua báo, đài thì khả năng tiếp cận khách hàng hạn chế nhưng với phương tiện marketing online thông qua website sẽ giúp tiếp cận nhiều khách hàng, nhanh chóng và hiệu quả hơn. Thông qua website, doanh nghiệp có thể triển khai các chiến lược marketing online như: quảng cáo Google Adwords, SEO hay Email Marketing.
* Nâng tầm uy tín và giá trị thương hiệu: Khi có một website bán hàng riêng của doanh nghiệp cũng sẽ giúp nâng tầm uy tín và giá trị thương hiệu của doanh nghiệp. Bộ mặt doanh nghiệp trên internet chính là website và màu sắc, tính năng, thông tin trên website sẽ phản ánh đầy đủ hình ảnh của doanh nghiệp với khách hàng góp phần khẳng định giá trị thương hiệu của doanh nghiệp trong lòng công chúng.

## MỤC TIÊU CỦA ĐỀ TÀI

* Tìm hiểu được các công nghệ mới như là Angular 8 CLI và các bộ User Interface để áp dụng vào đề tài.
* Kết nối giữa ASP .NETCORE và Angular 8 CLi để xây dựng một website.
* Cố gắng phát triển thêm để deploy sản phẩm lên docker.
* Thực hiện machine learning vào đề tài.

## CÁCH TIẾP CẬN VÀ NGIÊN CỨU

* **Đối tượng nghiên cứu**
* Back-End: .Net Core.
* Front-End: Angular 8 CLI, Ant-Design(Ng-Zoro), Typescript.
* Phạm vi nghiên cứu:
* Với tiểu luận tốt nghiệp này, nhóm em sẽ tập trung tìm hiểu và thực hiện các nội dung sau:
* Giới thiệu sơ lược và trình bày những nội dung cơ bản, những điểm mạnh, điểm yếu hay lợi ích mang lại cho nhà phát triển phần mềm của .NET Core.
* Tìm hiểu các nội dung, kiến thức về Angular 8 CLI, Ant-Design(Ng-Zoro), Typescript.

## KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC

* Hiểu rõ kiến thức về .NET Core
* Hiểu rõ kiến thức về Angular 8 CLI, Ant-Design(Ng-Zoro), Typescript.
* Xây dựng thành công một website thương mại điện tử sử dụng Docker với việc áp dụng những kiến thức đã tìm hiểu.

# 

# PHẦN NỘI DUNG

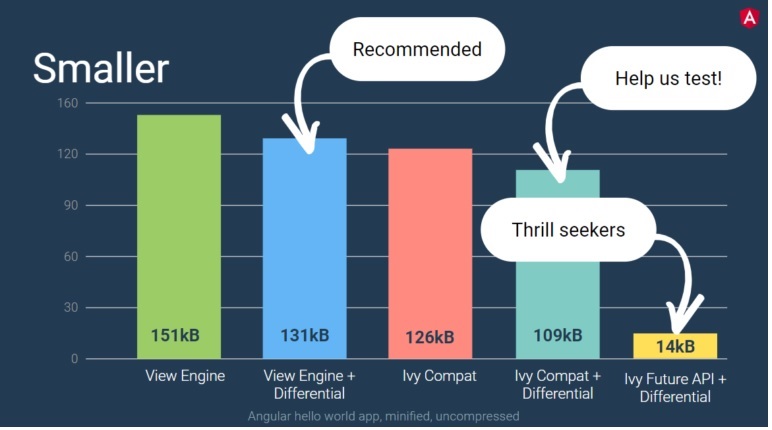
## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

### ANGULAR CLI 8

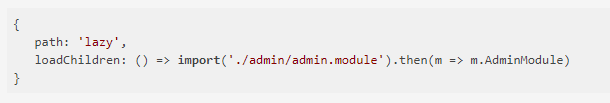
#### 1.1.1 Giới thiệu Angular 8 CLI

* Như chúng ta đã biết, Các ứng dụng Web hiện đại được các lập trình viên phát triển dựa trên các Framework mạnh mẽ như Angular. Các framework này cho phép các lập trình viên sử dụng các thư viện hiện đại và phức tạp của Javascript. Để thực hiện được điều này thì các nhà phát triển cần một môi trường để thiết lập các thư viện cần thiết cũng như là để phát triển các đoạn mã ứng dụng, toàn bộ các vấn đề này để cho đơn giả, nó đều được thiết lập để nằm trong cùng một thư mục, thư mục này sẽ chứa cả Code phần font-end của ứng dụng.
* Chúng ta đều biết Google vừa ra mắt phiên bản mới của Angular 8.0. Các nhà phát triển Angular và cộng đồng đã đợi bản update này từ lâu.
* Angular 8 sẽ phát hành với các tính năng của IVY engine. Angular 8 là phiên bản quan trọng từ Google trong năm 2019 nó tập trung vào các công cụ giúp Angular dễ dàng sử dụng cho các nhà phát triển xâ dựng các loại ứng dụng với sự tối ưu về hiệu năng.
* Ngoài ra, phiên bản này cũng chứa một số tính anwng mới được kỳ vọng từ các phiên bản trước. Angular 8 sẽ hỗ trợ phiên bản TypeScript 3.4 giúp code dễ và nhanh hơn rất nhiều.

#### 1.1.2 Tính năng mới trong Angular 8 CLI

* Hỗ trợ TypeScript 3.4.x:
* Angular 8 hỗ trợ bản TypeScript 3.4 hoặc cao hơn. Nếu bạn muốn dùng Angular 8 cho ứng dụng của mình thì cần upgrade TypeScript lên 3.4 hoặc cao hơn.
* IVY Rendering Engine:
* Tính năng quan trọng và được mong chờ nhất của Angular 8 là IVY render engine. IVY là trình biên dịch Angular hoạt động như một cơ chế render mới. Lợi ích của IVY là tạo ra các bundle nhỏ có thể thêm dễ dàng. IVY cơ bản là một cải tiến của Angular. Sau khi chuyển sang IVY thì ứng dụng đã có vẫn làm việc như trước đây nhưng giảm kích thước bundle. Trong Angular 8, Google giới thiệu bản preview cho IVY. Mục tiêu chính của bản này là nhận phản hồi từ các Developer liên quan tới Ivy. Nó được khuyến nghị không dùng cho môi trường production.
* Tạo ngconf 2019, Brad Green, Techincal Director của Angular Team tạo Google nói Ivy sẽ có một cải tiến đáng kể liên quan đến kích thước bundle và tương thích với tính năng Differnent Loading.
* Chúng ta sẽ có các lợi ích sau đây khi sử dụng Ivy:
* Nó sẽ giúp biên dịch nhanh hơn (dự kiến phát hành ở Angular 9)
  + Nhiều cải tiến kiên quan đến check type trong template và bắt lỗi trong thời gian build và giúp người dùng giảm gặp lỗi ở giai đoạn runtime.
  + Bundle size nhỏ hơn so với hiện tại
  + Nó cũng có thể tương thích ngược với các ứng dụng đã phát triển trước đây.
  + Code được tạo bởi Angular Complier sẽ dễ dàng để đọc hiểu hơn.
    - Bạn có thể debug cả template, tính năng này chắc chắn sẽ làm rất nhiều developer thích thú.

**Hình 1. Kích thước Bundle của ứng dụng với Angular Ivy và không có Ivy**

* Hỗ trợ Bazel
* Trong Angular 8, Google giới thiệu một công cụ build mới là Bazel. Thực tế nó không phải là công cụ mới hoàn toàn. Google sử dụng công cụ này nội bộ từ lâu và hiện tại họ phát hành nó như là một mã nguồn mở. Nhưng một thứ cần phải làm rõ là Bazel chưa thực sự sẵn sàng trên Angular 8. Nó được giới thiệu như là một tính năng nhỏ trong Angular 8 và được mong chờ là sẽ có bản chính thức trong Angular CLI 9. Chúng ta có thể có các lợi ích sau khi sử dụng tool này:
* Nó sẽ giúp build nhanh hơn. Nó sẽ tốn time hơn cho lần đầu build nhưng sẽ nhanh hơn nhiều nếu như build song song.
* + Dùng công cụ này chúng ta có thể build ứng dụng như là một incremental build (build chỉ những gì thay đổi) và deploy chỉ với những gì thay đổi hơn là toàn bộ ứng dụng.
* Chúng ta có thể ẩn file Bazel đi.
* Differential Loading cho việc tối ưu hóa hiệu năng
* Một trong những tính năng rất hay trong Angular CLI 8. Bởi vì nó giúp chúng ta chỉ ra trình duyệt nào sẽ được target và CLI sẽ build ứng dụng với các bundle JS liên quan cần thiết. Trình duyệt target mặc định trong tsconfig.json file giờ là es2015. Nghĩa là khi CLI build ứng dụng, nó sẽ build ứng dụng trong một trình duyệt cũ hơn như là IE9 sau đó chúng ta cần chỉ ra nó trong file browserlist. File này tòn tại trong thư mục gốc của ứng dụng và sử dụng.
* Thay đổi Lazy Loading trong Route
* Trong Angular từ khi khởi đầu cho đến nay, cơ chế router luôn hỗ trợ khái niệm lazy loading. Đến Angular 7 nó đã được hoàn thiện:
* Giá trị trước dấu # chỉ ra đường dẫn đến module file mà module đó tồn tại, còn sau dấu # chỉ ra tên của class Module. Phong cách này vẫn làm việc trong Angular 8, nhưng cách để viết lazy module được thay đổi. Chuỗi giá trị của loadChidren bị deprecated. Giờ loadChildren được khai báo như sau:
* Hỗ trợ Web Worker
* Như chúng ta đã biết JavaScript luôn thực thi đơn thread. Nó là điều tốt khi thực thi lượng lớn dữ liệu hoặc gọi liên tiếp nhau trong phương pháp bất đồng bộ. Nhưng theo kịch bản ứng dụng thực tế, khái niệm này không giúp gì nhiều. Đó là tại sao ngày này tất cả các trình duyệt web hỗ trợ web workder process. Cơ bản là web worker process là các script được thực thi bởi trình duyệt trong một thread tách biệt. Giao tiếp với các thread đó trong trình duyệt sẽ có thể gửi tin nhắn. Nhìn chung, web worker không liên quan đến Angular nhưng điểm chính là các script này cần được xem xét trong build process của ứng dụng. Mục tiêu chính là cung cấp một bundle cho tất cả web worker. Giờ Angular 8 đã thực hiện công việc này bởi Angular CLI.
* Theo chúng ta có thể cấu hình Angular CLI nếu chúng ta thêm web worker cho lần đầu tiên. Trong quá trình này, CLI sẽ loại bỏ các worker.ts từ tsconfig.json và thêm các cấu hình TypeScript với tên là tsconfig.worder.json để quản lý file worker.ts. CÁc thông tin này cũng được thêm vào angular.json.
* Thu thập phân tích dữ liệu sử dụng:
* Với Angular 8, Angluar CLI thu thập các dữ liệu phân tích cho Angular team có thể sắp xếp ưu tiên các tính năng và các cải tiến. Khi chúng ta update ứng dụng CLI, nó sẽ hỗ trợ tùy chọn ng analytics.
* Nếu chúng ta cho phép nó chạy toàn bộ, nó sẽ thu thập dữ liệu như các lệnh được sử dụng, các cờ được sử dụng, hệ điều hành, phiên bản Node, CPU count, Ram Size, thời gian thực thi, lỗi nếu có và Angular team có thể cải tiến và chỉnh sửa trong phiên bản tiếp.
* Loại bỏ @angular/http:
* Từ Angular 8, angular ngừng hỗ trợ @angular/http. Nó đã bị deprecated (báo ngừng hỗ trợ phiên bản sau) từ Angular 4 và cung cấp một thư viện hiệu quả và bảo mật hơn gọi là @angular/common/http.
* Nhưng ở Angular 7 vẫn cho phép chúng ta sử dụng @angular/http trong ứng dụng. Nhưng Angular 8 thì nó đã dừng hỗ trợ. Chúng ta cần chỉnh code lên @angular/common/http.
* Thay đổi ViewChild và ContentChild:
* Trong Angular 8 có một số thay đổi lớn với việc sử dụng ViewChild và ContentChild. Nó là những thứ không mong đợi của team Angular.
* Hiện tại Angular 8 nó bắt buộc cung cấp một cờ Boolean static để định nghĩa instance cho ViewChild và ConetntChild.
* Chúng ta đã có kinh nghiệm khi 1 issue thỉnh thoảng là các instance này có sẵn trong ngOnInit, nhưng đôi khi chúng ta không tìm thấy các instance này cho đến khi ngAfterContentInit hoặc ngAfterViewInit.
* Nó xuất hiện chính khi các phần thử được load vào DOM sau khi dữ liệu được binding. (giống \*ngIf hoặc \*ngFor) sau đó các phần thử được insert vào DOM trong ngAfterViewInit hoặc ngAfterContentInit.
* Điều này gây hiểu lầm và đó là vì sao chúng ta cần chỉ ra compoinent:



* Nếu giá trị static là true, angular sẽ cố tìm kiếm phần tử tở thời điểm khởi tạo component, ví dụ ngOnInit. Nó có thể hoàn thành khi phần tử không được sử dụng bất kỳ structure directive nào. Khi static value là false, angular sẽ tìm phần tử sau khi khởi tạo view.
* Hỗ trợ SVG Template:
* Hiện tại Angular 8 hỗ trợ template với đuôi mở rộng là SVG. Nghĩa là chúng ta có thể dùng file SVG để làm template như HTML mà không cần cấu hình gì khác cả.
* Nhưng câu hỏi là tại sao phải sử dụng .svg như là template thay vì sử dụng image trong HTML file. Lý do là khi sử dụng SVG template, chúng ta có thể sử dụng như là một directive và kết quả chúng ta có thể bind nó như HTML template. Với các tiếp cận này, chúng ta có thể tạo động các đồ hoạt tương tác trong ứng dụng.
* Hỗ trợ PNPM:
* Trong Angular 8, Angular CLI cũng hỗ trợ một package manager mới PNPM bao gồm NPM và Yarn. Trong câu lệnh ng add giờ cung cấp thêm 1 cái cờ nữa là –registry để thêm các packages từ bất cứ NPM private nào. Câu lệnh này đã có sẵn trên Angular CLI với ng update command.
* Hỗ trợ Codelyzer 5.0:
* Angular 8 hỗ trợ cấu hình TSLint bởi Codelyzer 5.0. Phiên bản này của Codelyzer, một vài quy tắc được đổi tên. Nhưng khi chúng ta upgrade Angular CLI, nó sẽ update file cấu hình TSLint. CLI update sẽ loại bỏ các lệnh import es6 từ file polyfills.ts. Bởi vì nó tự động được thêm vào nếu được yêu cầu bởi CLI.
* Hỗ trợ thư viện mới New Builders/ Architect API:
* Phiên bản mới của Angular CLI cho phép chúng ta sử dụng phiên bản mới của Builders được biết đến là Architect API. Angular sử dụng Builders API để thực hiện các hoạt động như server, build, test, lint và e2e. Chúng ta có thể sử dụng builders trong file angular.json.

#### 1.3 Ưu điểm và nhược điểm của Angular 8 CLI

* Ưu điểm:
* Angular được phát triển trên nền tảng hiện đại hơn, nhiều tính năng, và dễ dàng cho người mới học hơn.
* Với Angular 1, các nhà phát triển đã biết về sự khác nhau giữa Controllers, Services, Factories, Providers và các khái niệm khác có thể gây ra sự khó hiểu đặc biệt là các coder mới. Angular 8 là một framework với sự sắp xếp hợp lý hơn cho phép lập trình viên tập trung vào việc tạo lên các class JavaScript đơn giản. Views và các controllers sẽ được thay thế bởi Components, thứ mà có thể mô tả là một sự tinh chế mới của directives. Ngay cả các lập trình viên có kinh nghiệm với Angular cũng không chắc là có thể biết tất cả những khả năng của directives. Angular 8 component đã tăng khả năng đọc đáng kể, các API của chúng giảm đáng kể các biệt ngữ hơn là Angular 1.x directive.
* Angular 8 được viết với TypeScript, một superset của JavaScript, nó triển khai nhiều tính năng của ES2016 và các tính năng của TypeScript.
* Bằng cách tập trung tạo nên một framework dễ dàng hơn trong việc xử lý, Angular2 cho phép hệ sinh thái phát triển phong phú hơn.
* Các lập trình viên có thể sử dụng các trình soạn thảo tinh vi hơn với các tính năng gợi ý code hay tự động hoàn thành code.
* Những tính năng này giúp giảm thời gian học Angular2. May mắn hơn nữa cho các nhà phát triển ES5 truyền thống rằng không nhất thiết phải viết ES6 hay TypeScript, các coder có thể vẫn viết JavaScript thuần mà không cần biên dịch.
* Mặc dù được viết lại hoàn toàn, Angular 8 vẫn giữ lại các khái niệm cốt lõi và các quy tắc của Angular1.x. Điều đó có nghĩa là các lập trình viên giỏi vói Angular1 sẽ dễ dàng chuyển đổi sang Angular 8 hơn là các thư viện khác như ReactJS....
* Angular 8 được thiết kế cho mobile từ ban đầu. Bỏ qua một bên các vấn đề về sức mạnh xử lý, các thiết bị di động có các tính năng khác và sự giới hạn đó phân biệt chúng với máy tính truyền thống. Giao diện chạm, màn hình nhỏ, phần cứng đã được tính toán trong Angular 8. Máy tính để bàn cũng sẽ được cải tiến về hiệu suất và khả năng đáp ứng.
* Angular 8 giống React và các framework hiện đại khác, có thể tăng hiệu năng bằng cách render ra HTML trên server hoặc ngay cả trên browser. Phụ thuộc vào thiết kế cấu trúc giúp cho trải nghiệm người dùng tốt hơn.
* Yêu cầu về hiệu năng không bao giờ là dừng lại với việc tiền xử lý. Angular 8 có khả năng tích hợp với mobile bằng cách tích hợp với NativeScript, một thư viện mã nguồn mở là cậu nỗi giữa JavaScript và mobile. Bổ sung thêm, Ionic cũng đang được phát triển trên Angular8 cung cấp một cách khác giúp tăng hiệu quả của ứng dụng trên mobile.
* Phiên bản đầu tiên của Angular cung cấp cho các lập trình viên một thư viện mềm dẻo để phát triển ứng dụng. Đây là sự thay đổi đáng kể cho các lập trình viên web, và trong khi framework rất hữu ích thì nó là một bằng chứng cho sự linh hoạt đó. Qua thời gian, các best practice từ cộng đồng được tán thành rộng rãi.
* Angular 1.x đã cố gắng làm việc xung quanh các hạn chế trên trình duyệt liên quan đến JavaScript. Điều này được giải quyết bằng cách cung cấp một hệ thống các module sử dụng cơ chế Dependency Injection. Điều này thật tuyệt tuy nhiên có một vài vấn đề.
* Angular 2.x sử dụng hệ thống module ECMA Script 6 (ES6), và các công cụ đóng gói hiện đại như Webpack hoặc SystemJS. Module được tùy biến theo cách riêng của Angular để dễ dàng viết các Generic cho JS để cắm vào Angular. Hệ thống module mới giúp dễ dàng phát triển các dự án lớn hiệu quả hơn.

### 1.2 ANT-DESIGN(NG-ZORRO)

* Ng-Zoro là thư viện Angular UI ng-zorro-antd chứa một bộ các thành phần và bản trình diễn chất lượng cao để xây dựng giao diện người dùng tương tác phong phú.
* Các tính năng (Đoạn này bọn em xin được viết tiếng anh vì quá nhiều từ ngữ chuyên ngành):
* An enterprise-class UI design language for web applications.
* A set of high-quality Angular components out of the box.
* Written in TypeScript with complete defined types.
* Support OnPush mode, high performance.
* Support Customize theme.
* Support import Component individually.
* Môi trường hỗ trợ:
* Các trình duyệt hiện đại và Internet Explorer 11+.
* Server-side render.
* Electron
* Đặc tả
* ng-zorro-antd đồng bộ hóa đặc tả với Ant-Design thường xuyên, bạn có thể kiểm tra trực tuyến
* Hướng dẫn cài đặt
* Chúng ta nên sử dụng @angular/cli để cài đặt nó không chỉ giúp phát triển dễ dàng hơn mà còn cho phép bạn tận dụng hệ sinh thái phong phú của các gói.

+ Bước 1: ng new PROJECT\_NAME.

+ Bước 2: cd PROJECT\_NAME.

+ Bước 3: ng add ng-zorro-antd.

* Hoặc bạn có thể cài đặt bằng npm hoặc yarn

+ npm install ng-zorro-antd

### 1.3 TYPESCRIPT

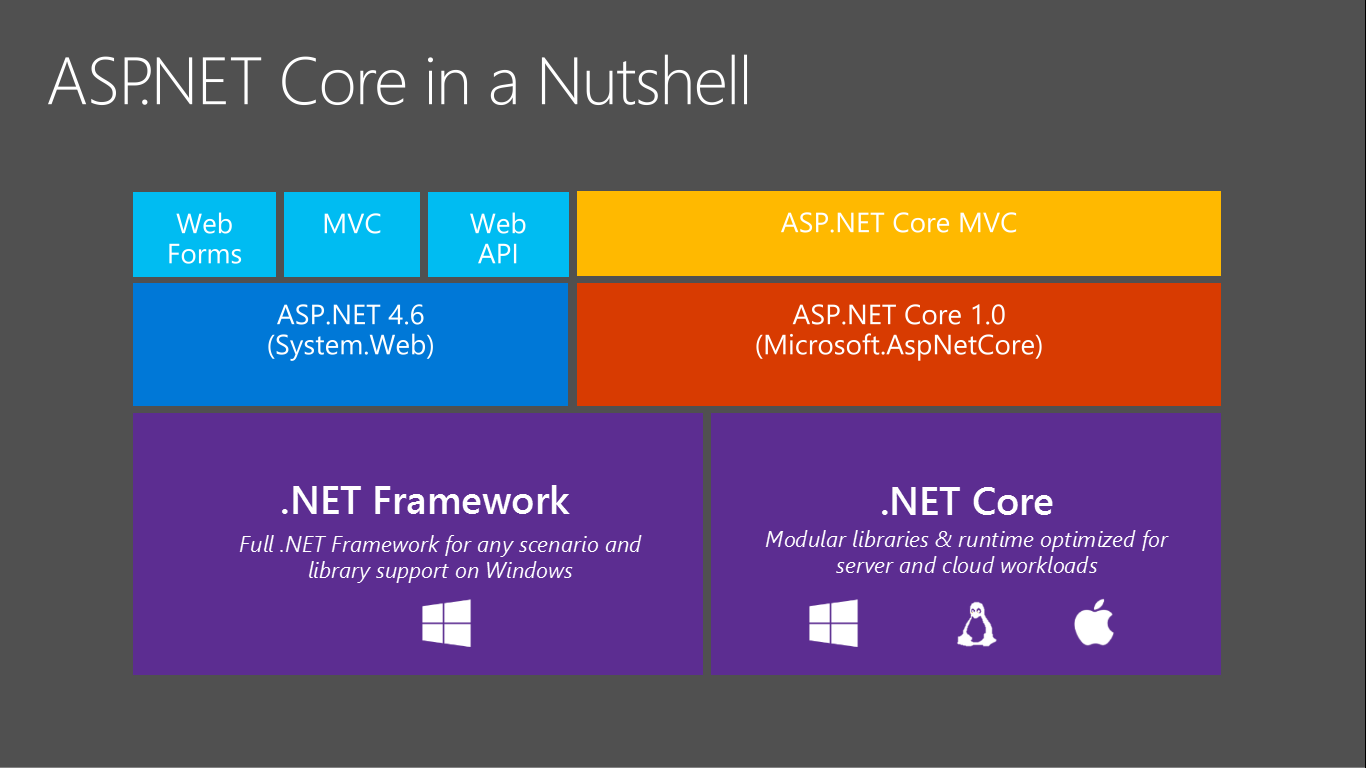
* Typescript là một dự án mã nguồn mở được Microsoft phát triển, được xem là một phiên bản nâng cao của Javascript.
* TypeScript là một ngôn ngữ giúp cung cấp quy mô lớn hơn so với JavaScript.
* Vì sao lại được xem là phiên bản nâng cao của Javascript? Vì nó được bổ sung những tùy chọn kiểu tĩnh và các lớp hướng đối tượng, nó bao hàm luôn ES6(ECMAScript 6 2015) - phiên bản mới nhất của Javascript.
* TypeScript thêm các namespace, class và module tùy chọn vào JavaScript. TypeScript hỗ trợ các công cụ cho các ứng dụng JavaScript quy mô lớn cho bất kỳ trình duyệt nào, cho bất kỳ máy chủ nào, trên bất kỳ hệ điều hành nào.
* Với những lập trình viên đã quen với Javascript thì chắc hẳn các bạn cũng hiểu được những khó khăn khi lập trình với nó.
* Tính mở trong việc sử dụng biến gây khó khăn trong việc kiểm soát sự thay đổi, cập nhật
* Các ứng dụng phức tạp đòi hỏi sử dụng rất nhiều file source, tách thành nhiều thư mục riêng thì việc đảm bảo tính thống nhất sẽ mất rất nhiều công sức vì phải thao tác bằng tay khá nhiều.
* Về cơ bản thì Javascript có hõ trợ OOP nhưng khi áp dụng thì lại rất khó khăn bởi cách triển khai code không hề đơn giản so với những ngôn ngữ bậc cao khác như Java, C#, Ruby ...
* Vậy ưu điểm của Typescript là gì và tại sao chúng ta lại sử dụng nó? Ưu điểm của Typescript:
* Dễ dàng hơn trong phát triển các dự án lớn, được hỗ trợ bởi các Javascript Framework lớn.
* Hỗ trợ OOP mạnh: Hầu hết các cú pháp hướng đối tượng đều được hỗ trợ bởi Typescript như kế thừa, đóng gói, constructor, abstract, interface, implement, override...v.v
* Cách tổ chức code rõ ràng hơn bởi được hỗ trợ các kỹ thuật mới nhất và hỗ trợ lập trình hướng đối tượng: Hỗ trợ cơ chế giúp kiến trúc hệ thống code hướng module, hỗ trợ namespace, giúp xây dựng các hệ thống lớn nơi mà nhiều dev có thể làm việc cùng nhau một cách dễ dàng hơn.
* Hỗ trợ các tính năng mới nhất của Javascript.
* Một lợi thế của Typescript nữa là mã nguồn mở vì vậy nó miễn phí và có cộng đồng hỗ trợ rất lớn.
* Hướng dẫn cài đặt Typescript
* Do đặc thù của Typescript là phải được biên dịch sang mã javascript trước khi thực thì. Nên Typescript không thể chạy trực tiếp trên trình duyệt.
* Để có thể sử dụng Typescript, bạn cần phải cài nodejs và npm. Bạn có thể tìm lại hướng dẫn cài đặt chi tiết hai thành phần này tại đây: Cài đặt Nodejs và npm.
* Sau đó cài đặt Typescript với câu lệnh sau:

+ npm install –g typescript

* Tham số -g tức là typescript sẽ được cài vào thư mục global của npm, bạn có thể sử dụng typescript cho nhiều dự án mà không phải cài đặt lại.

### 1.4 ASP .NETCORE

#### 1.4.1 Giới thiệu ASP.NET/ASP.NET Core

- ASP.NET Core là một framework mã nguồn mở, hiệu suất cao và đa nền tảng dùng để xây dựng các ứng dụng hiện đại có kết nối với Internet và dựa trên mô hình đám mây.

**Hình 2: Vị trí .NET Core trong mô hình phát triển .NET**

- ASP.NET Core được phát triển bởi tập đoàn Microsoft và cộng đồng lập trình viên. ASP.NET Core cũng là một framework kiểu module có khả năng thực thi trên framework .NET, Windows và .NET Core đa nền tảng.

- ASP.NET Core còn được xem là sự kết hợp giữa ASP.NET MVC và ASP.NET Web API tạo thành một mô hình lập trình đơn. Mặc dù được xây dựng mới, ASP.NET Core vẫn có tính tương thích cao với ASP.NET MVC.

- Hơn nữa, các ứng dụng ASP.NET Core hỗ trợ kiểu phiên bản “side by side”, tức là cùng chạy trên một máy tính với việc lựa chọn nhiều phiên bản ASP.NET Core khác nhau. Điều này là không thể với các phiên bản ASP.NET trước kia. Phiên bản ASP.NET Core mới nhất tính đến thời điểm viết bài này là ASP.NET 2.1.

- Với ASP.NET Core, bạn có thể:

* Xây dựng các ứng dụng web, các dịch vụ, ứng dụng IoT và các phần backend mobile.
* Sử dụng các công cụ phát triển ưa thích trên Windows, macOS và Linux.
* Triển khai trên đám mây hoặc tại chỗ.
* Chạy trên .NET Core hoặc .NET Framework.

#### 1.4.2 Tại sao dùng ASP.NET Core?

* Như bạn đã biết, có hàng triệu lập trình viên đã và đang sử dụng ASP.NET 4.x để xây dựng các ứng dụng Web. ASP.NET Core là một phiên bản thiết kế lại của ASP.NET 4.x, với nhiều thay đổi kiến trúc giúp framework nhẹ hơn và có tính module nhiều hơn. Do đó, các lập trình viên có thể tiếp tục xây dựng ứng dụng bằng ASP.NET Core với nền tảng hiệu suất và tính tương thích tốt hơn.
* ASP.NET mang lại các lợi ích như sau:
* Dùng để xây dựng giao diện Web (Web UI) cũng như các API Web.
* Tích hợp các framework phía client hiện đại và các quy trình làm việc phát triển.
* Hệ thống cấu hình sẵn có trên đám mây.
* Tích hợp sẵn nhúng phụ thuộc.
* Đường ống (pipeline) yêu cầu HTTP mang tính module, hiện suất cao và nhẹ ký.
* Có khả năng lưu trữ (host) ở IIS, Nginx, Apache, Docker, hoặc tự host ở các tiến trình riêng.
* Tạo mới phiên bản app side-by-side với .NET Core.
* Tạo công cụ đơn giản hóa phát triển web hiện đại.
* Khả năng xây dựng, chạy trên Windows, macOS, và Linux.
* Mã nguồn mở và tập trung vào cộng đồng phát triển mã nguồn.
* ASP.NET Core được cung cấp dưới dạng các gói NuGet. Bạn có thể sử dụng các gói này để tối ưu hóa ứng dụng khi chỉ nhúng những thành phần cần thiết. Trên thực tế, các ứng dụng ASP.NET Core 2.x cũng chỉ yêu cầu một gói NuGet đơn lẻ.

#### 1.4.3 Xây dựng các API và giao diện (UI) Web dùng ASP.NET Core MVVM

* ASP.NET Core MVVM cung cấp các tính năng để xây dựng API Web và ứng dụng web như sau:
* Mô hình MVVM (Model-View-View-Model) giúp tạo ra các API web và ứng dụng web có khả năng thực nghiệm.
* Razor Pages (phần mới ở ASP.NET Core 2.0) là một mô hình lập trình dựa theo cơ chế trang web giúp xây dựng giao diện web dễ và hiệu quả hơn.
* Razor cung cấp một cú pháp hiệu quả cho các trang Razor và view MVVM.
* Tag Helper cho phép mã nguồn phía server tham gia vào việc tạo và phát sinh phần tử HTML ở các tập tin Razor.
* Tích hợp hỗ trợ cho các định dạng đa dữ liệu và việc dàn xếp nội dung cho phép các API web tiếp cận nhiều khách hàng hơn, bao gồm trên các trình duyệt và thiết bị mobile khác nhau.
* Ràng buộc mô hình tự động ánh xạ dữ liệu từ các yêu cầu HTTP đến các tham số phương thức hành động.
* Xác thực mô hình tự động thực thi xác thực dữ liệu phía client và server.

#### 1.4.4 Phát triển phía client

* ASP.NET Core tích hợp liền mạch với các thư viện và framework phía client, bao gồm including Angular, React, và Bootstrap.

#### 1.4.5 ASP.NET Core nhắm đến .NET Framework

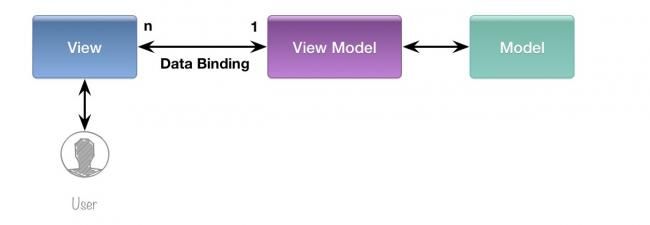
* ASP.NET Core có thể nhắm đến .NET Core hoặc .NET Framework. Các ứng dụng ASP.NET Core nhắm đến .NET Framework không phải là ứng dụng đa nền tảng khi chỉ chạy trên Windows. Hiện chưa có kế hoạch loại bỏ sự hỗ trợ .NET Framework ở ASP.NET Core. Nhìn chung, ASP.NET Core được tạo thành các thư viện .NET Standard (bản .NET chuẩn). Các ứng dụng viết bằng .NET Standard 2.0 cho thế chạy bất kỳ đâu nếu .NET Standard 2.0 được hỗ trợ.
* Có một số lợi ích khi nhắm đến .NET Core so với .NET Framework đó là:
* Đa nền tảng, chạy trên macOS, Linux và Windows.
* Nâng cao hiệu suất
* Tạo phiên bản side-by-side
* Các API mới
* Mã nguồn mở

#### 1.4.6 Ưu điểm và nhược điểm của ASP .NET

* Ưu điểm
* ASP.NET được xây dựng với bộ thư viện đa dạng và phong phú của .Net Framewrork, hỗ trợ vượt trội về XML và khả năng truy cập cơ sở dữ liệu qua ADO.net. Bên cạnh đó, ASP và ASPX có thể hoạt động trong cùng một ứng dụng mang lại hiệu năng cao cho website. Có thể nhận thấy ASP.NET mang một phong cách hoàn toàn mới, bạn có thể tách giao diện và code thành 2 phần khác biệt. Điều này giúp cho cho việc quản lý và bảo trì trở nên rõ ràng và đơn giản hơn.
* Thiết kế web bằng ASP.NET giúp cho hệ điều hành hoạt động trơn tru cũng như tăng tốc độ tải cho trang web của bạn. Với kiến trúc tương tự như các ứng dụng trên Windows, việc sử dụng cũng trở nên dễ dang hơn rất nhiều. Bên cạnh đó ASP.NET có khả năng tự động tạo ra mã HTML cho các Server để website có thể hoạt động trên hầu hết các trình duyệt được khách hàng sử dụng. Đặc biệt, ASP.NET là một ngôn ngữ mất phí nên bạn có thể hoàn toàn yên tâm về vấn đề bảo mật của nó. Với những ưu điểm vượt trội như vậy, ASP.NET luôn được rất nhiều nhà thiết kế web chuyên nghiệp ưa chuộng và sử dụng. Tuy nhiên, nó vẫn còn một số nhược điểm cần khắc phục mà bạn cần chú ý khi xây dựng web doanh nghiệp.
* Nhược điểm:
* Theo đánh giá từ các chuyên gia thiết kế website của thietkewebshop.vn và nhận định của các nhà phát triển trên toàn thế giới, ASP.NET thường có 2 nhược điểm lớn đó là không được hỗ trợ chạy trên hệ điều hành Linux và không được hỗ trợ Visual Studio trong suốt quá trình viết code.
* Tuy nhiên, với ASP.NET, người dùng có thể sử dụng bất kỳ một ngôn ngữ nào như C#, J# hay Visual Basic.net… sao cho phù hợp với website của bạn. Ngoài ra, ASP.NET sẽ hỗ trợ bạn biên dịch các website động và biến chúng thành các file DLL để server có thể thực thi các mệnh lệnh một các nhanh chóng và hiệu quả nhất.
* Ngoài ra, chi phí cho một website được thiết kế từ ASP.NET cũng cao hơn rất nhiều so với các mã nguồn mở miễn phí khác.

### 1.5 MÔ HÌNH MVVM

#### 1.5.1 Giới thiệu



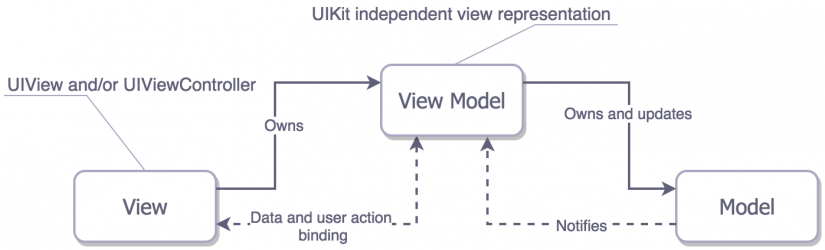
**Hình 3: Mô hình MVVM**

* Kể từ khi Microsoft giới thiệu hai nền tảng phát triển ứng dụng mới là WPF và Silverlight, đã có nhiều thay đổi trong việc xử lý sự kiện và binding dữ liệu, giữa các tầng của ứng dụng với nhau. Qua đó, hầu hết các công việc của tầng kết hợp với lớp presentation. Điều này làm nảy sinh ra nhu cầu phải có một mô hình phát triển ứng dụng mới phù hợp hơn. Và do đó, Model – View – ViewModel (MVVM) pattern ra đời và ngày càng trở nên phổ biến.
* Đa số các ứng dụng thuộc bất kì nền tảng nào cũng có thể chia thành hai phần: giao diện (View) và dữ liệu (Model). Vì việc tách riêng các phần này, cần phải có một phần trung gian nào đó nối kết hai phần này lại, và chúng tạo nên một mô hình (pattern).
* Quen thuộc và phổ biến nhất với chúng ta là mô hình MVC (Model – View – Controller) . Có thể nói MVC là một mô hình tiêu chuẩn bởi sự logic và hợp lý của nó. Điều này làm cho việc xuất hiện một mô hình phát triển ứng dụng mới có thể khiến bạn bỡ ngỡ.
* Trước khi tìm hiểu về mô hình MVVM này, ta cùng điểm qua một số tính năng mới trong xu hướng phát triển ứng dụng hiện nay.

#### *1.5.2* Data Binding

* Binding Data trong MVVM là điều không bắt buộc, một số implement chỉ đơn giản làm ViewModel như một lớp trung gian giữa Model-View, lớp này giữ nhiệm vụ format data hoặc mapping trạng thái của View. Tuy nhiên cách này theo mình khiến cho ViewModel trở thành Presenter và đưa kiến trúc này về MVP.

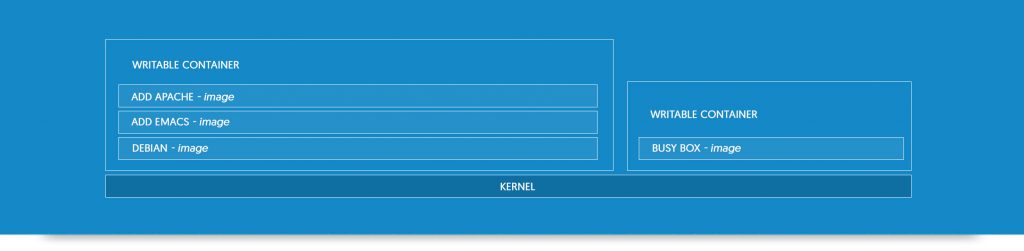
#### 1.5.3 MVVM được hiểu như thế nào?

* View: Tương tự như trong mô hình MVC, View là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn. Nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command.
* Model: Cũng tương tự như trong mô hình MVC. Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.
* ViewModel: Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực hiện data binding, command.
* Một điểm cần lưu ý là trong mô hình MVVM, các tầng bên dưới sẽ không biết được các thông tin gì về tầng bên trên nó. Như hình minh họa dưới đây:

**Hình 4: Workflow MVVM**

### 1.6 DOCKER

#### 1.6.1 Giới thiệu docker

* Docker là một nền tảng cho developers và sysadmin để develop, deploy và run application với container. Nó cho phép tạo các môi trường độc lập và tách biệt để khởi chạy và phát triển ứng dụng và môi trường này được gọi là container. Khi cần deploy lên bất kỳ server nào chỉ cần run container của Docker thì application của bạn sẽ được khởi chạy ngay lập tức.
* Docker là một tool để đóng gói ứng dụng thành các "gói chuẩn” phục vụ cho việc phát triển phần mềm.
* Docker cung cấp một khái niệm gọi là container, nó chứa một phần của phần mềm trong một hệ thống file đầy đủ, các container chứa mọi thứ cần để chạy bao gồm: mã nguồn, code biên dịch, công cụ hệ thống, thư viện hệ thống – bất kì cái gì cần thiết để cài đặt lên server.
* Nó đảm bảo cho phần mềm có thể chạy mà không bị phụ thuộc vào môi trường.

**Hình 5: Mô hình docker**

#### 1.6.2 Nếu không có docker thì sao?

* Nếu không có docker, để xây dựng môi trường phát triển, lập trình viên phải xây dựng các máy áo (bằng Virtual Box…).
* Lập trình viên có thể sử dụng công cụ như vagrant để có thể dễ quản lí và đồng bộ các môi trường phát triển. Như thế đòi hỏi máy lập trình viên cấu hình phải cao, các box sẽ tốn nhiều không gian đĩa, tốn RAM, chiếm dụng CPU.
* Nếu không có docker, để xây dựng môi trường deploy, nhà phát triển phải đảm bảo đóng gói các lệnh để build môi trường phát triển và môi trường deploy phải tương đồng.

#### 1.6.3 Ưu thế của docker

* Các container chạy trên một máy đơn cùng chia sẻ nhân hệ điều hành, so với giải pháp sử dụng máy ảo, Docker cần ít ram hơn. Các Images được xây dựng từ các hệ thống file và chia sẻ file nên việc tải về các image sẽ dễ dàng và ít tốn không gian đĩa hơn.
* Các docker container được dựa trên một tiêu chuẩn mở, cho phép các container chạy trên hầu hết phiên bản Linux và trên Mirosoft Windows và ở tầng trên của bất kì hạ tầng nào.
* Các container cô lập các ứng dụng khỏi hạ tầng bên dưới, và cung cấp một tầng nữa để bảo vệ ứng dụng. Khi có lỗ hổng của ứng dụng hoặc của container, việc tấn công của hacker chỉ ảnh hưởng đến container chứ không ảnh hưởng đến container khác hoặc cả server.

#### 1.6.4 Cài đặt docker

* Link download : <https://docs.docker.com/install/>. Chọn bản cài đặt tương ứng với hệ điều hành của bạn và tiến hành cài đặt theo hướng dẫn đối với Linux còn Windows và MacOS thì bạn chỉ cần tải bản cài về và cài đặt như mọi application khác.
* Mở command line:
* $ docker version
* $ docker info
* $ docker run hello-world

#### https://miro.medium.com/max/597/1*8-Gor5TMPqvkeNwSP3LmFw.png1.6.5 Một số khái niệm trong docker

**Hình 6: Workflow Docker**

* **Docker Client**: là cách mà bạn tương tác với docker thông qua command trong terminal. Docker Client sẽ sử dụng API gửi lệnh tới Docker Daemon.
* **Docker Daemon**: là server Docker cho yêu cầu từ Docker API. Nó quản lý images, containers, networks và volume.
* **Docker Volumes**: là cách tốt nhất để lưu trữ dữ liệu liên tục cho việc sử dụng và tạo apps.
* **Docker Registry**: là nơi lưu trữ riêng của Docker Images. Images được push vào registry và client sẽ pull images từ registry. Có thể sử dụng registry của riêng bạn hoặc registry của nhà cung cấp như : AWS, Google Cloud, Microsoft Azure.
* **Docker Hub**: là Registry lớn nhất của Docker Images ( mặc định). Có thể tìm thấy images và lưu trữ images của riêng bạn trên Docker Hub ( miễn phí).
* **Docker Repository**: là tập hợp các Docker Images cùng tên nhưng khác tags. VD: golang:1.11-alpine.
* **Docker Networking**: cho phép kết nối các container lại với nhau. Kết nối này có thể trên 1 host hoặc nhiều host.
* **Docker Compose**: là công cụ cho phép run app với nhiều Docker containers 1 cách dễ dàng hơn. Docker Compose cho phép bạn config các command trong file docker-compose.yml để sử dụng lại. Có sẵn khi cài Docker.
* **Docker Swarm**: để phối hợp triển khai container.
* **Docker Services**: là các containers trong production. 1 service chỉ run 1 image nhưng nó mã hoá cách thức để run image — sử dụng port nào, bao nhiêu bản sao container run để service có hiệu năng cần thiết và ngay lập tức.

## 2. CHƯƠNG 2: XÁC ĐỊNH MÔ HÌNH HÓA YÊU CẦU

### 2.1 XÁC ĐỊNH YÊU CẦU

**Bảng 1. Nhận diện tác nhân và các chức năng trong sơ đồ use case**

|  |  |
| --- | --- |
| Tác nhân | Chức năng |
| User | * Đăng kí tài khoản * Đăng nhập * Xem danh sách sản phẩm * Tìm kiếm sản phẩm * Lịch sử mua hàng. * Đánh giá sản phẩm * Rate sản phẩm * Bình luận sản phẩm * Giỏ hàng * Thêm/ sửa/ xoá/ xem sản phẩm trong giỏ hàng * Quên mật khẩu |
| Admin | * Quản lý sản phẩm ( Xem/ thêm/ sửa/ xóa sản phẩm). * Quản lý đánh giá sản phẩm ( Xem/ thêm/ sửa/ xóa các đánh giá sản phẩm). * Quản lý khuyến mãi (Xem/ thêm/ sửa/ xóa các khuyến mãi). * Quản lý account khách hàng (Xem/ thêm/ sửa/ xóa khách hàng). * Quản lý Lịch sử đăng nhập của khách hàng. * Quản lý Account nhân viên (Xem/ thêm/ sửa/ xóa account nhân viên). * Quản lý menu (Xem/ thêm/ sửa/ xóa menu) |

### 2.2 THIẾT KẾ USECASE DIAGRAM

**Hình 7: Use case diagram**

* **Mô tả use case diagram**

**Bảng 2: Mô tả các actor**

|  |  |  |
| --- | --- | --- |
| TT | Tên Actor | Ý nghĩa |
|  | Khách hàng | Người mua hàng |
|  | Admin | Người quản lý sản phẩm |

**Bảng 3: Mô tả Use case diagram**

|  |  |  |
| --- | --- | --- |
| TT | Tên use case | Ý nghĩa |
|  | Đăng kí | **Actor: Khách hàng**  **Mô tả tóm tắt:**Khách hàng đăng kí tài khoản để sử dụng dịch vụ.  **Các bước thực hiện**:  **+** Click vào Button đăng kí.  + Đăng kí bằng Email, số điện thoại, Facebook, Google |
|  | Đăng nhập | **Actor: Khách hàng**  **Mô tả tóm tắt:**Khách hàng đăng nhập để sử dụng dịch vụ.  **Các bước thực hiện**:  **+** Click vào Button đăng nhập.  + Đăng nhập bằng account đã đăng kí hoặc facebook, google. |
|  | Quên mật khẩu | **Actor: Khách hàng**  **Mô tả tóm tắt:**Khi khách hàng quên mật khẩu. Thì yêu cầu được cấp lại mật khẩu  **Các bước thực hiện**:  **+** Click vào Button quên mật khẩu.  + Nhận email cấp lại mật khẩu. |
|  | Lịch sử mua hàng | **Actor: Khách hàng**  **Mô tả tóm tắt:**Khách hàng xem lại các lịch sử mua hàng.  **Các bước thực hiện**: |
|  | Đánh giá sản phẩm | **Actor: Khách hàng**  **Mô tả tóm tắt:**Khách hàng có thể đánh giá sản phẩm dựa theo rate sao. |
|  | Thanh toán | **Actor: Khách hàng**  **Mô tả tóm tắt:**Khách hàng có thể thanh toán online hoặc khi nhận hàng. |
|  | Tìm kiếm sản phẩm | **Actor: Khách hàng**  **Mô tả tóm tắt:**Khách hàng xem lại các lịch sử mua hàng. |
|  | Xem danh sách sản phẩm | **Actor: Khách hàng**  **Mô tả tóm tắt:**Khách hàng xem danh sách các sản phẩm hiển thị trên website. |
|  | Quản lý sản phẩm | **Actor: Admin**  **Mô tả tóm tắt:**Admin có thể thêm, sửa, xóa sản phẩm. |
|  | Quản lý đánh giá sản phẩm | **Actor: Admin**  **Mô tả tóm tắt:**Admin có thể thêm, sửa, xóa đánh giá sản phẩm. |
|  | Quảng lý khuyến mãi | **Actor: Admin**  **Mô tả tóm tắt:**Admin có thể thêm, sửa, xóa khuyến mãi. |
|  | Quản lý account khách hàng | **Actor: Admin**  **Mô tả tóm tắt:**Admin có thể thêm, sửa, xóa account khách hàng. |
|  | Quản lý account nhân viên | **Actor: Admin**  **Mô tả tóm tắt:**Admin có thể thêm, sửa, xóa account nhân viên. |
|  | Quảng lý menu | **Actor: Admin**  **Mô tả tóm tắt:**Admin có thể thêm, sửa, xóa menu. |

### 2.3 ĐẶC TẢ USECASE

***Bảng 4: Đặt tả use case đăng kí***

|  |  |
| --- | --- |
| Name | Đăng kí |
| Description | Khách hàng đăng kí để có tài khoản để sử dụng dịch vụ. |
| Actor | Khách hàng |
| Pre-condition | Không có |
| Trigger | Khách hàng vào trang đăng kí |
| Basic Flow | **1** Khách hàng mở trang đăng kí.   1. Nhập Email vào thẻ input. 2. Nhập Password vào thẻ input. 3. Nhập tên khách hàng 4. Nhấn nút Đăng kí. 5. Kiểm tra các thông tin trên có chính xác hay không?   + Chính xác: Tới **bước 8**  + Không chính xác: Tới **bước 7**  + User không tồn tại: Tới **bước 7**   1. Thông báo “Đăng kí thành công!”. Quay về **Bước 2.** 2. Chuyển vào trang sản phẩm |
| Alternative flow | +Password or Email để trống.  Hệ thống kiểm tra các thẻ input, nếu có thẻ input nào bị trống thì hệ thống sẽ hiển thị thông báo "Vui lòng nhập đầy đủ Email và Password".  + Nếu không thực hiện thành công chuyển đến trang báo lỗi. |

**Bảng 5: Đặt tả use case đăng nhập**

|  |  |
| --- | --- |
| Name | Đăng nhập |
| Description | Khách hàng sử dụng chức năng để đăng nhập vào website. |
| Actor | Khách hàng, Admin |
| Pre-condition | Không có |
| Trigger | Khách hàng vào trang đăng nhập |
| Basic Flow | **1** Khách hàng mở trang đăng nhập.   1. Nhập Email vào thẻ input. 2. Nhập Password vào thẻ input. 3. Nhấn nút đăng nhập. 4. Kiểm tra các thông tin trên có chính xác hay không?   + Chính xác: Tới **bước 7**  + Không chính xác: Tới **bước 6**  + User không tồn tại: Tới **bước 6**   1. Thông báo “Invalid Credentials!”. Quay về **Bước 2.** 2. Chuyển vào trang chủ hoặc trang admin theo phân quyền . |
| Alternative flow | +Password or Email để trống.  Hệ thống kiểm tra các thẻ input, nếu có thẻ input nào bị trống thì hệ thống sẽ hiển thị thông báo "Vui lòng nhập đầy đủ Email và Password".  + Nếu không thực hiện thành công chuyển đến trang báo lỗi. |

**Bảng 6: Đặt tả use case Thêm sản phẩm**

|  |  |
| --- | --- |
| Name | Xem sản phẩm |
| Description | Khách hàng sử dụng chức năng để xem các sản phẩm có trên website. |
| Actor | Khách hàng |
| Pre-condition | Không có |
| Trigger | Khách hàng vào trang chủ |
| Basic Flow | 1. Khách hàng mở trang chủ. 2. Khách hàng có thể xem các sản phẩm trên trang web |
| Alternative flow | Không có |

**Bảng 7: Đặt tả use case Tìm kiếm sản phẩm**

|  |  |
| --- | --- |
| Name | Tìm kiếm sản phẩm |
| Description | Khách hàng sử dụng chức năng để tìm kiếm các sản phẩm có trên website. |
| Actor | Khách hàng |
| Pre-condition | Không có |
| Trigger | Khách hàng vào trang chủ. |
| Basic Flow | 1. Khách hàng mở trang chủ. 2. Khách hàng nhấn vào button tìm kiếm. 3. Nhập tên sản phẩm muốn tìm kiếm sau đó nhấn “Enter”. 4. Sản phẩm sẽ hiển thị theo đúng yêu cầu của khách hàng. 5. Kiểm tra có đúng hay không ?   + Chính xác: tới bước 4  + Không chính xác: tới bước 6   1. Thông báo “Không tìm thấy sản phẩm!”. Quay về **Bước 3.** 2. Chuyển vào trang sản phẩm chi tiết. |
| Alternative flow | + Input tìm kiếm trống. |

**Bảng 8: Đặt tả use case Giỏ hàng**

|  |  |
| --- | --- |
| Name | Giỏ hàng |
| Description | Khách hàng sử dụng chức năng để xem các sản phẩm có trong giỏ hàng vào website. |
| Actor | Khách hàng |
| Pre-condition | Cần phải đăng nhập |
| Trigger | Khách hàng vào trang chủ |
| Basic Flow | 1. Khách hàng mở trang login. 2. Khách hàng đăng nhập. 3. Khách hàng sẽ vào trang chủ. 4. Chọn sản phâm cần mua xong nhấn button “Add to cart” 5. Sản phẩm sẽ được lưu vào giỏ hàng. 6. Khách hàng nhấn vào button Cart trên thanh Menu. 7. Hiện lên 1 modal popup để khách hàng thực hiện các thao tác thêm, sửa, xóa các sản phẩm Khách hàng đã mua. |
| Alternative flow | + Khách hàng cần phải đăng nhập đê thực hiện các chức năng. |

**Bảng 9: Đặt tả use case Lịch sử mua hàng**

|  |  |
| --- | --- |
| Name | Lịch sử mua hàng |
| Description | Khách hàng sử dụng chức năng để xem lại các sản phẩm mà khách hàng đã mua. |
| Actor | Khách hàng |
| Pre-condition | Cần phải đăng nhập |
| Trigger | Khách hàng vào trang chủ |
| Basic Flow | 1. Khách hàng mở trang login. 2. Khách hàng đăng nhập. 3. Khách hàng sẽ vào trang chủ. 4. Chọn lịch sử mua hàng 5. Sẽ hiển thị các sản phẩm mà khách hàng đã mua |
| Alternative flow | + Khách hàng cần phải đăng nhập đê thực hiện các chức năng. |

**Bảng 10: Đặt tả use case Đánh giá sản phẩm**

|  |  |
| --- | --- |
| Name | Đánh giá sản phẩm |
| Description | Khách hàng sử dụng chức năng để đánh giá sản phẩm |
| Actor | Khách hàng |
| Pre-condition | Cần phải đăng nhập |
| Trigger | Khách hàng vào trang chủ |
| Basic Flow | 1. Khách hàng mở trang login. 2. Khách hàng đăng nhập. 3. Khách hàng sẽ vào trang chủ. 4. Chọn sản phẩm và nhấn vào sản phẩm 5. Khách hàng sẽ tới trang chi tiết sản phẩm 6. Khách hàng có thể rate sao sản phẩm. |
| Alternative flow | + Khách hàng cần phải đăng nhập đê thực hiện các chức năng. |

**Bảng 11: Đặt tả use case Thanh toán**

|  |  |
| --- | --- |
| Name | Thanh toán |
| Description | Khách hàng sử dụng chức năng để thanh toán sản phẩm. |
| Actor | Khách hàng |
| Pre-condition | Cần phải đăng nhập |
| Trigger | Khách hàng vào trang chủ |
| Basic Flow | 1. Khách hàng mở trang login. 2. Khách hàng đăng nhập. 3. Khách hàng sẽ vào trang chủ. 4. Chọn sản phâm cần mua xong nhấn button “Add to cart” 5. Sản phẩm sẽ được lưu vào giỏ hàng. 6. Khách hàng nhấn vào button Cart trên thanh Menu. 7. Hiện lên 1 modal popup để khách hàng thực hiện các thao tác thêm, sửa, xóa các sản phẩm Khách hàng đã mua. 8. Sau khi thực hiện xong bước 7. 9. Khách hàng nhấn “Thanh toán” để chuyển đến trang thanh toán. 10. Ở trang thanh toán khách hàng cần điền những thông tin cần thiết để thanh toán. 11. Sau khi nhập xong chọn thanh toán bằng Cod hoặc thẻ tín dụng. 12. Thanh toán thành công thì đơn hàng sẽ ghi lại trong lịch sử mua hàng. |
| Alternative flow | + Khách hàng cần phải đăng nhập đê thực hiện các chức năng.  + Khách hàng phải điền đầy đủ thông tin khi thanh toán. |

**Bảng 12: Đặt tả use case Thêm tài khoản nhân viên**

|  |  |
| --- | --- |
| Name | Quản lý tài khoản nhân viên (Thêm) |
| Description | Admin sử dụng chức năng này để thêm tài khoản nhân viên. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “+ Thêm” để thêm tài khoản nhân viên. 5. Admin sẽ điền các thông tin cơ bản vào các input : Mã tài khoản, Tên tài khoản, UserName, PassWord, Nhập lại PassWord, Email, Ngày Sinh, Giới tính, Số điện thoại, Địa chỉ, Tỉnh thành phố, Quận huyện. 6. Nhân nút “Lưu” để thêm nhân viên. |
| Alternative flow | + Admin cần phải đăng nhập.  + Admin nhập vào đầy đủ các Input không thiếu bất kì input nào. |

**Bảng 13: Đặt tả use case xem tài khoản nhân viên**

|  |  |
| --- | --- |
| Name | Quản lý tài khoản nhân viên (Xem) |
| Description | Admin sử dụng chức năng này để xem tài khoản nhân viên. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin chọn menu Tài Khoản. 5. Hiển thị danh sách tài khoản nhân viên. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 14: Đặt tả use case xóa tài khoản nhân viên**

|  |  |
| --- | --- |
| Name | Quản lý tài khoản nhân viên (Xóa) |
| Description | Admin sử dụng chức năng này để xóa tài khoản nhân viên. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Ở menu chọn Tài Khoản 5. Admin nhấn nút “Xóa” để xóa tài khoản nhân viên. 6. Sẽ hiện lên 1 modal để bạn lựa chọn 7. Nhân nút “Đồng ý” để xóa nhân viên. Hoặc “Hủy” để không xóa. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 15: Đặt tả use case sửa tài khoản nhân viên**

|  |  |
| --- | --- |
| Name | Quản lý tài khoản nhân viên (Sửa) |
| Description | Admin sử dụng chức năng này để sữa thông tin tài khoản nhân viên. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “Sửa” để sửa tài khoản nhân viên. 5. Admin sẽ sửa các thông tin cơ bản mà cần sửa vào các input : Mã tài khoản, Tên tài khoản, UserName, PassWord, Nhập lại PassWord, Email, Ngày Sinh, Giới tính, Số điện thoại, Địa chỉ, Tỉnh thành phố, Quận huyện. 6. Nhân nút “Lưu” để sửa nhân viên. |
| Alternative flow | + Admin cần phải đăng nhập.  + Không được để trống 1 input nào. |

**Bảng 16: Đặt tả use case tìm kiếm tài khoản nhân viên**

|  |  |
| --- | --- |
| Name | Quản lý tài khoản nhân viên (Tìm kiếm) |
| Description | Admin sử dụng chức năng này để tìm kiếm tài khoản nhân viên. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập bằng username và password. 3. Sau khi đăng nhập thành công sẽ chuyển vào dashboar. 4. Ở thanh menu chọn Tài Khoản 5. Admin nhập vào ôn input phía trên để tìm kiếm thông tin. 6. Sau khi nhập xong thì nhấn Tìm để thực hiện tìm kiếm. 7. Nếu đúng thì hiển thị Thông tin tài khoản. 8. Nếu sai thì quay lại bước 5. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 17: Đặt tả use case Thêm menu**

|  |  |
| --- | --- |
| Name | Quản lý menu (Thêm) |
| Description | Admin sử dụng chức năng này để thêm menu. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “+ Thêm” để thêm menu. 5. Admin sẽ điền các thông tin cơ bản vào các input. 6. Nhân nút “Lưu” để thêm menu. |
| Alternative flow | + Admin cần phải đăng nhập.  + Admin nhập vào đầy đủ các Input không thiếu bất kì input nào. |

**Bảng 18: Đặt tả use case xem menu**

|  |  |
| --- | --- |
| Name | Quản lý menu (Xem) |
| Description | Admin sử dụng chức năng này để xem menu. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin chọn menu. 5. Hiển thị danh sách menu. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 19: Đặt tả use case xóa menu**

|  |  |
| --- | --- |
| Name | Quản lý tài menu (Xóa) |
| Description | Admin sử dụng chức năng này để xóa menu. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Ở menu chọn Menu 5. Admin nhấn nút “Xóa” để xóa menu. 6. Sẽ hiện lên 1 modal để bạn lựa chọn 7. Nhân nút “Đồng ý” để xóa menu. Hoặc “Hủy” để không xóa. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 20: Đặt tả use case sửa menu**

|  |  |
| --- | --- |
| Name | Quản lý menu (Sửa) |
| Description | Admin sử dụng chức năng này để menu. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “Sửa” để sửa menu. 5. Admin sẽ sửa các thông tin cơ bản mà cần sửa 6. Nhân nút “Lưu” để sửa menu. |
| Alternative flow | + Admin cần phải đăng nhập.  + Không được để trống 1 input nào. |

**Bảng 21: Đặt tả use case tìm kiếm menu**

|  |  |
| --- | --- |
| Name | Quản lý menu (Tìm kiếm) |
| Description | Admin sử dụng chức năng này để tìm kiếm menu. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập bằng username và password. 3. Sau khi đăng nhập thành công sẽ chuyển vào dashboar. 4. Ở thanh menu chọn Menu 5. Admin nhập vào ôn input phía trên để tìm kiếm thông tin. 6. Sau khi nhập xong thì nhấn Tìm để thực hiện tìm kiếm. 7. Nếu đúng thì hiển thị Thông menu. 8. Nếu sai thì quay lại bước 5. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 22: Đặt tả use case Thêm sản phẩm**

|  |  |
| --- | --- |
| Name | Quản lý sản phẩm (Thêm) |
| Description | Admin sử dụng chức năng này để thêm sản phẩm. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “+ Thêm” để thêm sản phẩm. 5. Admin sẽ điền các thông tin cơ bản vào các input. 6. Nhân nút “Lưu” để thêm sản phẩm. |
| Alternative flow | + Admin cần phải đăng nhập.  + Admin nhập vào đầy đủ các Input không thiếu bất kì input nào. |

**Bảng 23: Đặt tả use case Xem sản phẩm**

|  |  |
| --- | --- |
| Name | Quản lý sản phẩm (Xem) |
| Description | Admin sử dụng chức năng này để xem sản phẩm nhân viên. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin chọn Sản Phẩm trên thanh menu. 5. Hiển thị danh sách sản phẩm. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 24: Đặt tả use case Xóa sản phẩm**

|  |  |
| --- | --- |
| Name | Quản lý sản phẩm (Xóa) |
| Description | Admin sử dụng chức năng này để xóa menu. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Ở menu chọn Menu 5. Admin nhấn nút “Xóa” để xóa sản phẩm. 6. Sẽ hiện lên 1 modal để bạn lựa chọn. 7. Nhân nút “Đồng ý” để xóa sản phẩm. Hoặc “Hủy” để không xóa. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 25: Đặt tả use case sửa Sản phẩm**

|  |  |
| --- | --- |
| Name | Quản lý sản phẩm (Sửa) |
| Description | Admin sử dụng chức năng này để sửa sản phẩm. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “Sửa” để sửa thông sản phẩm. 5. Admin sẽ sửa các thông tin cơ bản mà cần sửa 6. Nhân nút “Lưu” để sửa sản phẩm. |
| Alternative flow | + Admin cần phải đăng nhập.  + Không được để trống 1 input nào. |

**Bảng 26. Đặt tả use case tìm kiếm sản phẩm**

|  |  |
| --- | --- |
| Name | Quản lý sản phẩm (Tìm kiếm) |
| Description | Admin sử dụng chức năng này để tìm kiếm sản phẩm. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập bằng username và password. 3. Sau khi đăng nhập thành công sẽ chuyển vào dashboar. 4. Ở thanh menu chọn Menu 5. Admin nhập vào ô input phía trên để tìm kiếm thông tin. 6. Sau khi nhập xong thì nhấn Tìm để thực hiện tìm kiếm. 7. Nếu đúng thì hiển thị Thông tin sản phẩm. 8. Nếu sai thì quay lại bước 5. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 27: Đặt tả use case thêm đánh giá sản phẩm**

|  |  |
| --- | --- |
| Name | Quản lý đánh giá sản phẩm (Thêm) |
| Description | Admin sử dụng chức năng này để thêm đánh giá sản phẩm. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “+ Thêm” để thêm đánh giá sản phẩm. 5. Admin sẽ điền các thông tin cơ bản vào các input. 6. Nhân nút “Lưu” để thêm nhân viên. |
| Alternative flow | + Admin cần phải đăng nhập.  + Admin nhập vào đầy đủ các Input không thiếu bất kì input nào. |

**Bảng 28: Đặt tả use case xem tài khoản nhân viên**

|  |  |
| --- | --- |
| Name | Quản lý đánh giá sản phẩm (Xem) |
| Description | Admin sử dụng chức năng này để đánh giá sản phẩm. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin chọn Quản lý đánh giá. 5. Hiển thị danh sách đánh giá. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 29: Đặt tả use case xóa tài khoản nhân viên**

|  |  |
| --- | --- |
| Name | Quản lý Đánh giá sản phẩm (Xóa) |
| Description | Admin sử dụng chức năng này để xóa đánh giá sản phẩm. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Ở menu chọn Đánh giá sản phẩm 5. Admin nhấn nút “Xóa” để xóa đánh giá sản phẩm. 6. Sẽ hiện lên 1 modal để bạn lựa chọn 7. Nhân nút “Đồng ý” để xóa đánh giá. Hoặc “Hủy” để không xóa. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 30: Đặt tả use case sửa đánh giá sản phẩm**

|  |  |
| --- | --- |
| Name | Quản lý đánh giá sản phẩm (Sữa) |
| Description | Admin sử dụng chức năng này để đánh giá sản phẩm. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “Sửa” để sửa tài khoản nhân viên. 5. Admin sẽ sửa các thông tin cơ bản mà cần sửa vào các input. 6. Nhân nút “Lưu” để sửa nhân viên. |
| Alternative flow | + Admin cần phải đăng nhập.  + Không được để trống 1 input nào. |

**Bảng 31: Đặt tả use case tìm kiếm đánh giá sản phẩm**

|  |  |
| --- | --- |
| Name | Quản lý đánh giá sản phẩm (Tìm kiếm) |
| Description | Admin sử dụng chức năng này để tìm kiếm đánh giá sản phẩm. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập bằng username và password. 3. Sau khi đăng nhập thành công sẽ chuyển vào dashboar. 4. Ở thanh menu chọn Đánh giá sản phẩm 5. Admin nhập vào ôn input phía trên để tìm kiếm thông tin. 6. Sau khi nhập xong thì nhấn Tìm để thực hiện tìm kiếm. 7. Nếu đúng thì hiển thị đánh giá sản phẩm. 8. Nếu sai thì quay lại bước 5. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 32. Đặt tả use case thêm khuyến mãi**

|  |  |
| --- | --- |
| Name | Quản lý khuyến mãi (Thêm) |
| Description | Admin sử dụng chức năng này để thêm khuyến mãi. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “+ Thêm” để thêm khuyến mãi. 5. Admin sẽ điền các thông tin cơ bản vào các input. 6. Nhân nút “Lưu” để thêm khuyến mãi. |
| Alternative flow | + Admin cần phải đăng nhập.  + Admin nhập vào đầy đủ các Input không thiếu bất kì input nào. |

**Bảng 33: Đặt tả use case xem khuyến mãi**

|  |  |
| --- | --- |
| Name | Quản lý khuyến mãi (Xem) |
| Description | Admin sử dụng chức năng này để xem khuyễn mãi. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin chọn Quản lý khuyến mãi. 5. Hiển thị danh sách khuyến mãi. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 34: Đặt tả use case xóa khuyến mãi**

|  |  |
| --- | --- |
| Name | Quản lý khuyến mãi (Xóa) |
| Description | Admin sử dụng chức năng này để xóa khuyến mãi. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Ở menu chọn Đánh giá sản phẩm 5. Admin nhấn nút “Xóa” để xóa khuyến mãi. 6. Sẽ hiện lên 1 modal để bạn lựa chọn 7. Nhân nút “Đồng ý” để xóa khuyến mãi. Hoặc “Hủy” để không xóa. |
| Alternative flow | + Admin cần phải đăng nhập. |

**Bảng 35: Đặt tả use case sửa khuyến mãi**

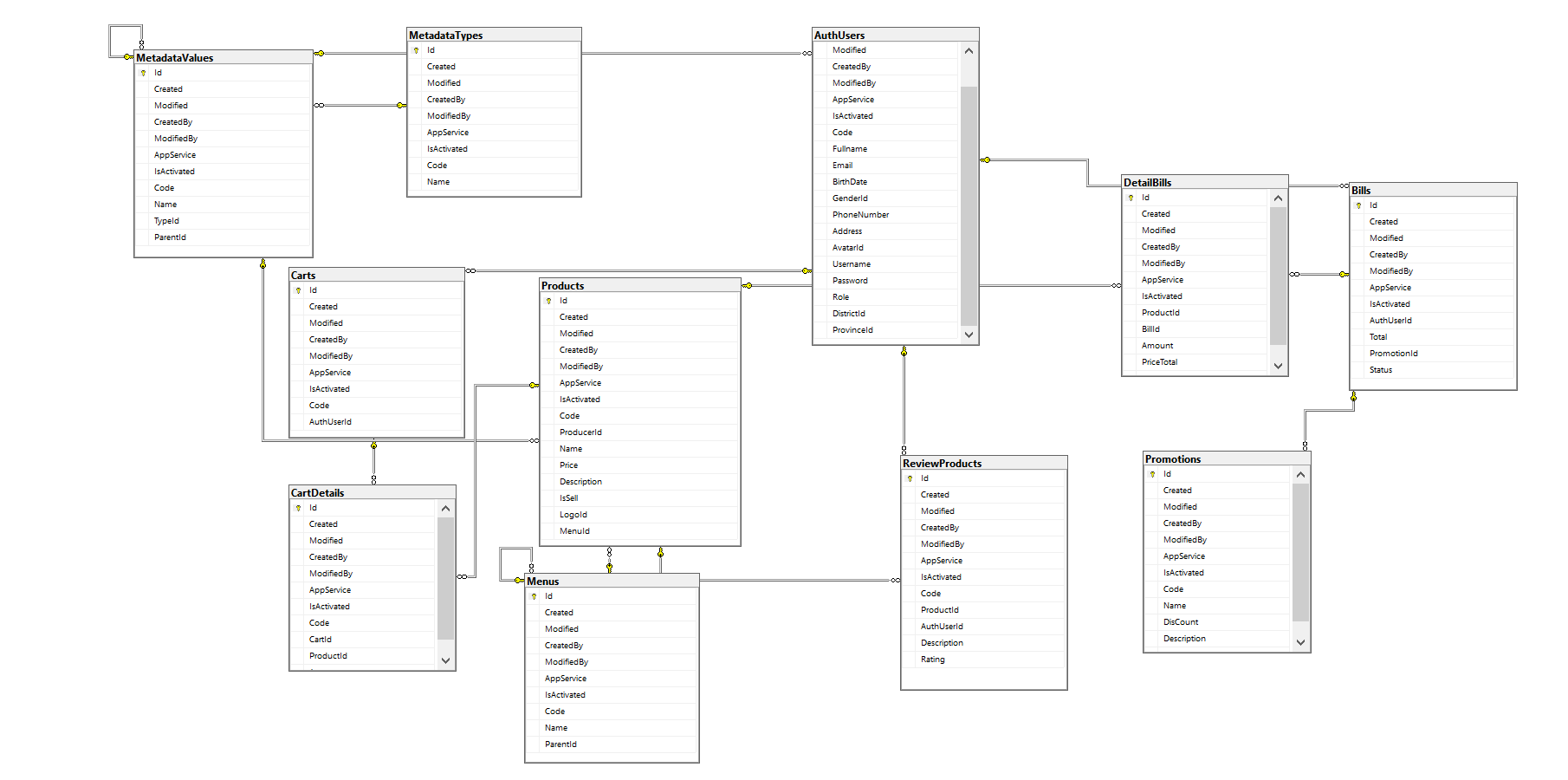
|  |  |
| --- | --- |
| Name | Quản lý đánh giá sản phẩm (sửa) |
| Description | Admin sử dụng chức năng này để sửa khuyến mãi. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập. 3. Admin sẽ vào dashboar. 4. Admin nhấn nút “Sửa” để sửa khuyến mãi. 5. Admin sẽ sửa các thông tin cơ bản mà cần sửa vào các input. 6. Nhân nút “Lưu” để sửa khuyến mãi. |
| Alternative flow | + Admin cần phải đăng nhập.  + Không được để trống 1 input nào. |

**Bảng 36: Đặt tả use case tìm kiếm khuyến mãi**

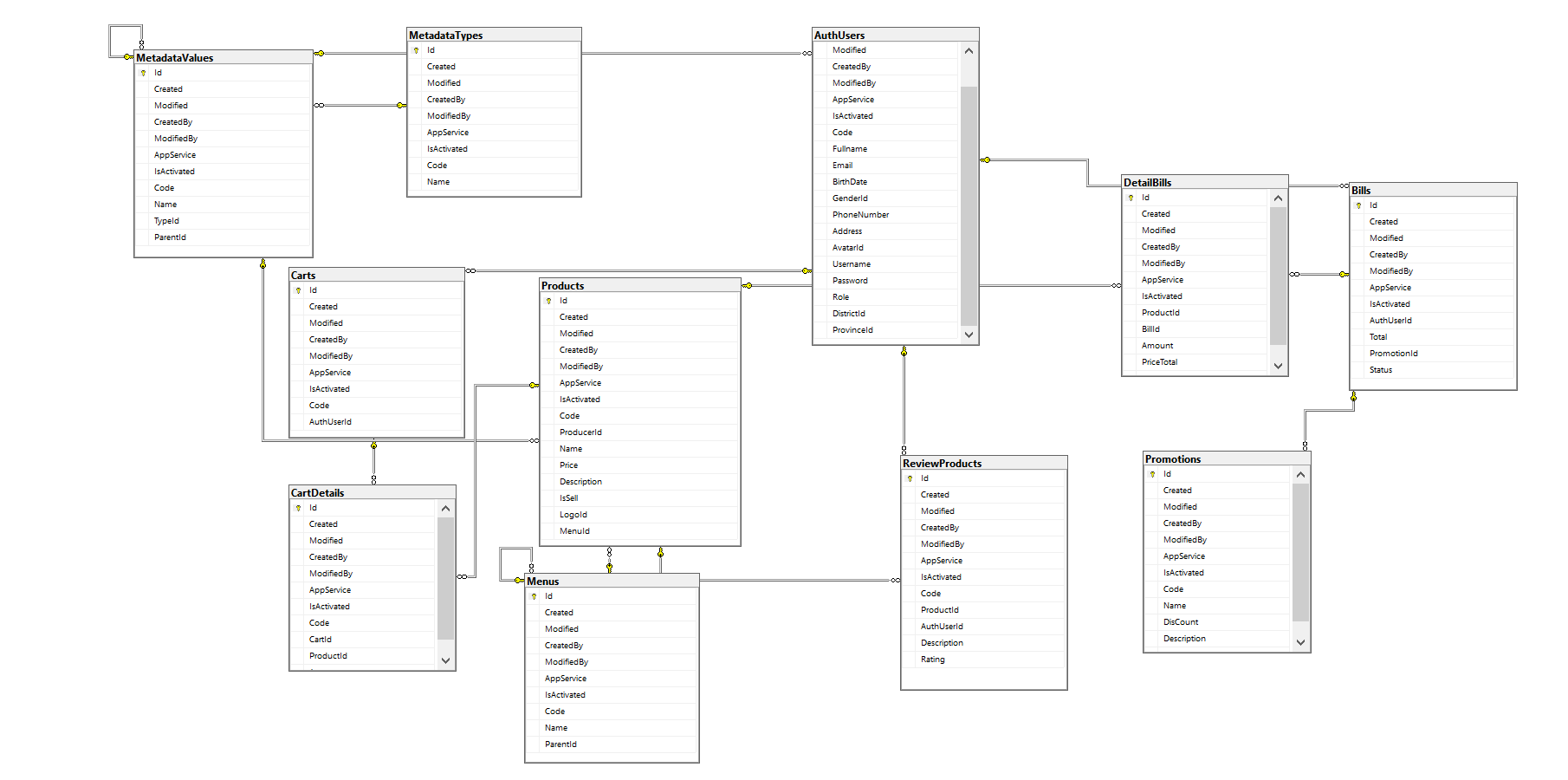
|  |  |
| --- | --- |
| Name | Quản lý khuyến mãi (Tìm kiếm) |
| Description | Admin sử dụng chức năng này để tìm kiếm khuyến mãi. |
| Actor | Admin |
| Pre-condition | Cần phải đăng nhập. |
| Trigger | Admin vào trang dashboar. |
| Basic Flow | 1. Admin mở trang đăng nhập. 2. Admin đăng nhập bằng username và password. 3. Sau khi đăng nhập thành công sẽ chuyển vào dashboar. 4. Ở thanh menu chọn Khuyến mãi 5. Admin nhập vào ôn input phía trên để tìm kiếm thông tin. 6. Sau khi nhập xong thì nhấn Tìm để thực hiện tìm kiếm. 7. Nếu đúng thì hiển thị khuyến mãi. 8. Nếu sai thì quay lại bước 5. |
| Alternative flow | + Admin cần phải đăng nhập. |

## CHƯƠNG 3: THIẾT KẾ CƠ SỞ DỮ LIỆU

### 3.1 THIẾT KẾ DỮ LIỆU



**Hình 8a: Mô hình cơ sở dữ liệu**

****

**Hình 8b: Mô hình cơ sở dữ liệu**

### 3.2 MÔ TẢ CHI TIẾT CÁC BẢNG

**Bảng 37: Bảng AuthUsers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | Id của user | uniqueidentifier | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo tài khoản | Datetimeoffset(7) | - |
| 3 | Modified | Ngày chỉnh sửa tài khoản | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo tài khoản | Nvarchar(255) |  |
| 5 | ModifiedBy | Người chỉnh sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated | Kiểm tra hoạt động | Bit |  |
| 8 | Code | Mã tài khoản | Nvarchar(255) |  |
| 9 | Fullname | Tên người dùng | Nvarchar(255) |  |
| 10 | Email | Email người dùng | Nvarchar(255) |  |
| 11 | BirthDate | Ngày sinh người dùng | Datetimeoffset(7) |  |
| 12 | GenderId | Giới tính | Uniqueidentifer |  |
| 13 | PhoneNumber | Số điện thoại | Nvarchar(255) |  |
| 14 | Address | Địa chỉ | Nvarchar(255) |  |
| 15 | AvartarId | Avarta | Uniqueidentifer |  |
| 16 | Username | Tài khoản đăng nhập | Nvarchar(255) |  |
| 17 | Password | Mật khẩu đăng nhập | Nvarchar(255) |  |
| 18 | Role | Quyền | Nvarchar(MAX) |  |
| 19 | DistricId | Huyện | Uniqueidentifer |  |
| 20 | ProvinceId | Tỉnh | Uniqueidentifer |  |

**Bảng 38: Bảng Products**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | Id | Id của sản phẩm | Uniqueidentifer | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo tài khoản | Datetimeoffset(7) |  |
| 3 | Modified | Ngày chỉnh sửa tài khoản | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo tài khoản | Nvarchar(255) |  |
| 5 | ModifiedBy | Người chỉnh sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated |  | bit |  |
| 8 | Code | Mã sản phẩm | Nvarchar(50) |  |
| 9 | ProducerId | Mã nhà sản xuất | Uniqueidentifer |  |
| 10 | Name | Tên sản phẩm | Nvarchar(255) |  |
| 11 | Price | Giá sản phẩm | Float |  |
| 12 | Description | Mô tả sản phẩm | Nvarchar(500) |  |
| 13 | isSell | Đang kinh doanh | Bit |  |
| 14 | LogoId | Hình ảnh | Nvarchar(500) |  |
| 15 | MenuId | Vị trí trong menu | Uniqueidentifer |  |

**Bảng 39: Bảng Bills**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | Id của Bill | uniqueidentifier | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo | Datetimeoffset(7) |  |
| 3 | Modified | Ngày sửa | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo | Nvarchar(255) |  |
| 5 | ModifiedBy | Người sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated |  | Bit |  |
| 8 | Code | Mã khách hàng | Nvarchar(50) |  |
| 9 | AuthUserId | Tổng hóa đơn | uniqueidentifier |  |
| 10 | Total | Mã khuyến mãi | Float |  |
| 11 | PromotionId | Trạng thái hóa đơn | uniqueidentifier |  |

**Bảng 40: Bảng Carts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | Id của Cart | uniqueidentifier | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo | Datetimeoffset(7) |  |
| 3 | Modified | Ngày chỉnh sửa | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo | Nvarchar(255) |  |
| 5 | ModifiedBy | Người chỉnh sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated |  | Bit |  |
| 8 | Code | Mã | Nvarchar(50) |  |
| 9 | AuthUserId | Trạng thái người dùng | uniqueidentifier |  |

**Bảng 41: Bảng CartDetails**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | Id của Bill | uniqueidentifier | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo | Datetimeoffset(7) |  |
| 3 | Modified | Ngày chỉnh sửa | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo | Nvarchar(255) |  |
| 5 | ModifiedBy | Người chỉnh sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated |  | Bit |  |
| 8 | Code | Mã cart | Nvarchar(50) |  |
| 9 | CartId | Id cart | uniqueidentifier |  |
| 10 | ProductId | Id sản phẩm | uniqueidentifier |  |
| 11 | Amount | Số lượng | int |  |

**Bảng 42: Bảng DetailBills**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | Id của BillDetails | uniqueidentifier | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo | Datetimeoffset(7) |  |
| 3 | Modified | Ngày chỉnh sửa | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo | Nvarchar(255) |  |
| 5 | ModifiedBy | Người chỉnh sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated |  | Bit |  |
| 8 | Code | Mã cart | Nvarchar(50) |  |
| 9 | BillId | Id hóa đơn | uniqueidentifier |  |
| 10 | ProductId | Id sản phẩm | uniqueidentifier |  |
| 11 | Amount | Số lượng | int |  |
| 12 | PriceTotal | Tổng giá | float |  |

**Bảng 43: Bảng Menus**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | Id của Menu | uniqueidentifier | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo | Datetimeoffset(7) |  |
| 3 | Modified | Ngày chỉnh sửa | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo | Nvarchar(255) |  |
| 5 | ModifiedBy | Người chỉnh sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated |  | Bit |  |
| 8 | Code | Mã menu | Nvarchar(50) |  |
| 9 | Name | Tên menu | Nvarchar(55) |  |
| 10 | ParentId | Chi nhánh cha | uniqueidentifier |  |

**Bảng 44: Bảng MetaDataType**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | Id của MetaDataType | uniqueidentifier | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo | Datetimeoffset(7) |  |
| 3 | Modified | Ngày chỉnh sửa | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo | Nvarchar(255) |  |
| 5 | ModifiedBy | Người chỉnh sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated |  | Bit |  |
| 8 | Code | Mã | Nvarchar(50) |  |
| 9 | Name | Tên | Nvarchar(55) |  |

**Bảng 45: Bảng MetaDataValues**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | Id của MetaDataValues | uniqueidentifier | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo | Datetimeoffset(7) |  |
| 3 | Modified | Ngày chỉnh sửa | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo | Nvarchar(255) |  |
| 5 | ModifiedBy | Người chỉnh sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated |  | Bit |  |
| 8 | Code | Mã | Nvarchar(50) |  |
| 9 | Name | Tên | Nvarchar(55) |  |
| 10 | TypeId | Loại Metadata | uniqueidentifier |  |
| 11 | ParentId | chi nhánh cha | uniqueidentifier |  |

**Bảng 46: Bảng Promotions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | Id của Promotions | uniqueidentifier | * PK(Primary Key) * Tăng tự động |
| 2 | Created | Ngày tạo | Datetimeoffset(7) |  |
| 3 | Modified | Ngày chỉnh sửa | Datetimeoffset(7) |  |
| 4 | CreatedBy | Người tạo | Nvarchar(255) |  |
| 5 | ModifiedBy | Người chỉnh sửa | Nvarchar(255) |  |
| 6 | AppService |  | Nvarchar(255) |  |
| 7 | isActivated |  | Bit |  |
| 8 | Code | Mã | Nvarchar(50) |  |
| 9 | Name | Tên | Nvarchar(55) |  |
| 10 | DisCount | Giá khuyến mãi | int |  |
| 11 | Description | Mô tả | Nvarchar(MAX) |  |
| 12 | BillID | Mã hóa đơn | uniqueidentifier |  |

## 4. CHƯƠNG 4: THIẾT KẾ GIAO DIỆN VÀ XỬ LÝ

### MÀN HÌNH QUẢN LÝ TÀI KHOẢN

**Hình 9: Màn hình quản lý Tài Khoản**

**Bảng 47: Thông tin các đối tượng trên màn hình quản lý tài khoản**

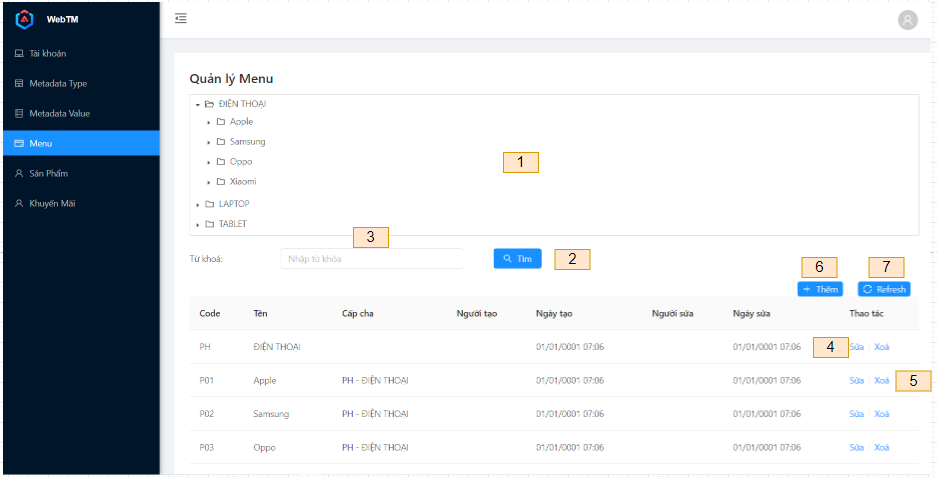
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Nút “Tìm” | Thẻ <button> |  |
| 2 | Khung tìm kiếm keyword | Thẻ <imput> |  |
| 3 | Nút “+ Thêm” | Thẻ <button> |  |
| 4 | Nút “ Refesh” | Thẻ <button> |  |
| 5 | Thao tác “Sửa” | Thẻ <a> |  |
| 6 | Thao tác “Xóa” | Thẻ <a> |  |
| 7 | Thông tin người dùng | Thẻ <a> |  |

* Thiết kế xử lý

**Bảng 48: Bảng xử lý màn hình quản lý tài khoản**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Tìm | Nhấn vào nút sẽ tìm theo keyword |  |
| 02 | Khung tìm kiếm keyword | Nhập các keyword để tìm kiếm. |  |
| 03 | + Thêm | Khi click sẽ hiện lên 1 modal popup. |  |
| 04 | Refesh | Khi click vào button sẽ reset lại những cái nhập. |  |
| 05 | Sửa | Khi click sẽ hiện 1 modal chưa thông tin của tài khoản. |  |
| 06 | Xóa | Khi click sẽ hiện 1 modal để bạn lựa chọn xóa hay không. |  |

### MÀN HÌNH QUẢN LÝ MENU



**Hình 10: Màn hình Quản lý Menu**

**Bảng 49: Bảng thông tin các đối tượng trên màn hình quản lý menu**

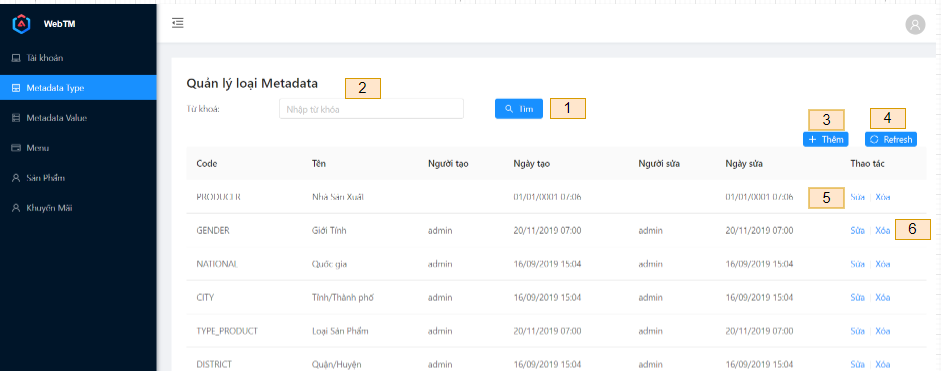
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Hiển thị cây menu mẫu | Thẻ <div> |  |
| 2 | Nút “Tìm” | Thẻ <button> |  |
| 3 | Khung tìm kiếm keyword | Thẻ <input> |  |
| 4 | Thao tác “Sửa” | Thẻ <a> |  |
| 5 | Thao tác “Xóa” | Thẻ <a> |  |
| 6 | Nút “Thêm” | Thẻ <button> |  |
| 7 | Nút “Refesh” | Thẻ <button> |  |

* Thiết kế xử lý

**Bảng 50: Bảng thiết kế xử lý màn hình quản lý menu**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Hiển thị cây menu | Khi có dữ liệu |  |
| 02 | Tìm | Nhấn vào nút sẽ tìm theo keyword. Lọc dữ liệu trong table. |  |
| 03 | Khung tìm kiếm | Nhập các keyword để tìm kiếm. |  |
| 04 | Sửa | Khi click sẽ hiện 1 modal chưa thông tin của tài khoản. |  |
| 05 | Xóa | Khi click sẽ hiện 1 modal để bạn lựa chọn xóa hay không. |  |
| 06 | Thêm | Khi click sẽ hiện lên 1 modal popup. |  |
| 07 | Refesh | Khi click vào button sẽ reset lại những cái nhập. |  |

### MÀN HÌNH QUẢN LÝ METADATA TYPE



**Hình 11: Màn hình quản lý loại MetaDataType**

**Bảng 51:** **Bảng thông tin các đối tượng trên màn hình quản lý MetaDataType**

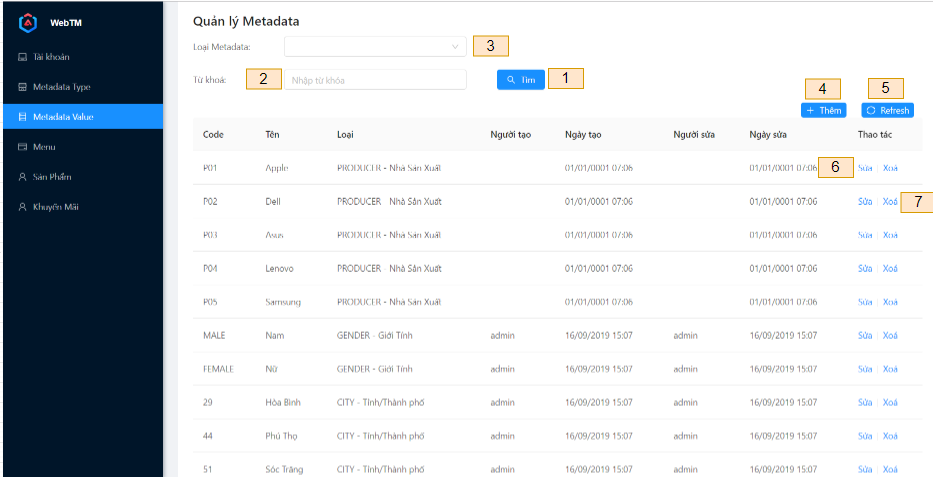
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Nút “Tìm” | Thẻ <button> |  |
| 2 | Khung tìm kiếm keyword | Thẻ <input> |  |
| 3 | Nút “Thêm” | Thẻ <button> |  |
| 4 | Nút “Refesh” | Thẻ <button> |  |
| 5 | Thao tác “Sửa” | Thẻ <a> |  |
| 6 | Thao tác “Xóa” | Thẻ <a> |  |

* Thiết kế xử lý

**Bảng 52: Bảng thiết kế xử lý màn hình quản lý MetaDataType**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Tìm | Nhấn vào nút sẽ tìm theo keyword. Lọc dữ liệu trong table. |  |
| 02 | Khung tìm kiếm | Nhập các keyword để tìm kiếm. |  |
| 03 | Thêm | Khi click sẽ hiện 1 modal để bạn nhập thông tin. |  |
| 04 | Refesh | Khi click vào button sẽ reset lại những cái nhập. |  |
| 05 | Sửa | Khi click sẽ hiện 1 modal để bạn sửa thông tin. |  |
| 06 | Xóa | Khi click sẽ hiện 1 modal để bạn lựa chọn xóa hay không. |  |

### MÀN HÌNH QUẢN LÝ METADATA VALUE



**Hình 12: Màn hình quản lý MetaDataValue**

**Bảng 53:** **Bảng thông tin các đối tượng trên màn hình quản lý MetaDataValue**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Nút “Tìm” | Thẻ <button> |  |
| 2 | Khung tìm kiếm keyword | Thẻ <input> |  |
| 3 | Combobox tìm kiếm | Thẻ <select> |  |
| 4 | Nút “Thêm” | Thẻ <button> |  |
| 5 | Nút “Refesh” | Thẻ <button> |  |
| 6 | Thao tác “Sửa” | Thẻ <a> |  |
| 7 | Thao tác “Xóa” | Thẻ <a> |  |

* Thiết kế xử lý

**Bảng 54: Bảng thiết kế xử lý màn hình quản lý MetaDataValue**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Tìm | Nhấn vào nút sẽ tìm theo keyword. Lọc dữ liệu trong table. |  |
| 02 | Khung tìm kiếm | Nhập các keyword để tìm kiếm. |  |
| 03 | Lọc dữ liệu | Chọn vào combobox và cho các giá trị trong mặc định trong combobox để lọc dữ liệu. |  |
| 04 | Thêm | Khi click sẽ hiện 1 modal để bạn nhập thông tin. |  |
| 05 | Refesh | Khi click vào button sẽ reset lại những cái nhập. |  |
| 06 | Sửa | Khi click sẽ hiện 1 modal để bạn sửa thông tin. |  |
| 07 | Xóa | Khi click sẽ hiện 1 modal để bạn lựa chọn xóa hay không. |  |

### MÀN HÌNH QUẢN LÝ SẢN PHẨM

**Hình 13: Màn hình quản lý sản phẩm**

**Bảng 55: Bảng thông tin các đối tượng trên màn hình quản lý sản phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Nút “Tìm” | Thẻ <button> |  |
| 2 | Khung tìm kiếm keyword | Thẻ <input> |  |
| 3 | Combobox tìm kiếm | Thẻ <select> |  |
| 4 | Nút “Thêm” | Thẻ <button> |  |
| 5 | Nút “Refesh” | Thẻ <button> |  |
| 6 | Thao tác “Sửa” | Thẻ <a> |  |
| 7 | Thao tác “Xóa” | Thẻ <a> |  |
| 8 | Thanh menu loại sản phẩm | Thẻ <div> |  |

* Thiết kế xử lý

**Bảng 56: Bảng thiết kế xử lý màn hình quản lý sản phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Tìm | Nhấn vào nút sẽ tìm theo keyword. Lọc dữ liệu trong table. |  |
| 02 | Khung tìm kiếm | Nhập các keyword để tìm kiếm. |  |
| 03 | Lọc dữ liệu | Chọn vào combobox và cho các giá trị trong mặc định trong combobox để lọc dữ liệu. |  |
| 04 | Thêm | Khi click sẽ hiện 1 modal để bạn nhập thông tin. |  |
| 05 | Refesh | Khi click vào button sẽ reset lại những cái nhập. |  |
| 06 | Sửa | Khi click sẽ hiện 1 modal để bạn sửa thông tin. |  |
| 07 | Xóa | Khi click sẽ hiện 1 modal để bạn lựa chọn xóa hay không. |  |

### MÀN HÌNH QUẢN LÝ GIẢM GIÁ

**Hình 14: Màn hình quản lý Khuyến Mãi**

**Bảng 57: Bảng thông tin các đối tượng trên màn hình quản lý khuyến mãi**

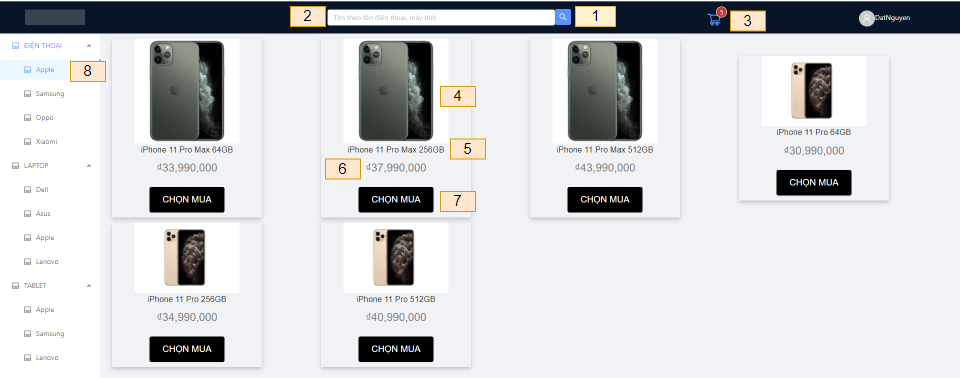
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Nút “Tìm” | Thẻ <button> |  |
| 2 | Khung tìm kiếm keyword | Thẻ <input> |  |
| 3 | Nút “Thêm” | Thẻ <button> |  |
| 4 | Nút “Refesh” | Thẻ <button> |  |
| 5 | Thao tác “Sửa” | Thẻ <a> |  |
| 6 | Thao tác “Xóa” | Thẻ <a> |  |

* Thiết kế xử lý

**Bảng 58: Bảng thiết kế xử lý màn hình quản lý khuyến mãi**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Tìm | Nhấn vào nút sẽ tìm theo keyword. Lọc dữ liệu trong table. |  |
| 02 | Khung tìm kiếm | Nhập các keyword để tìm kiếm. |  |
| 03 | Thêm | Khi click sẽ hiện 1 modal để bạn nhập thông tin. |  |
| 04 | Refesh | Khi click vào button sẽ reset lại những cái nhập. |  |
| 05 | Sửa | Khi click sẽ hiện 1 modal để bạn sửa thông tin. |  |
| 06 | Xóa | Khi click sẽ hiện 1 modal để bạn lựa chọn xóa hay không. |  |

### MÀN HÌNH HIỂN THỊ SẢN PHẨM



**Hình 15: Màn hình hiển thị sản phẩm client**

**Bảng 59: Bảng thông tin các đối tượng trên màn hình hiển thị sản phẩm**

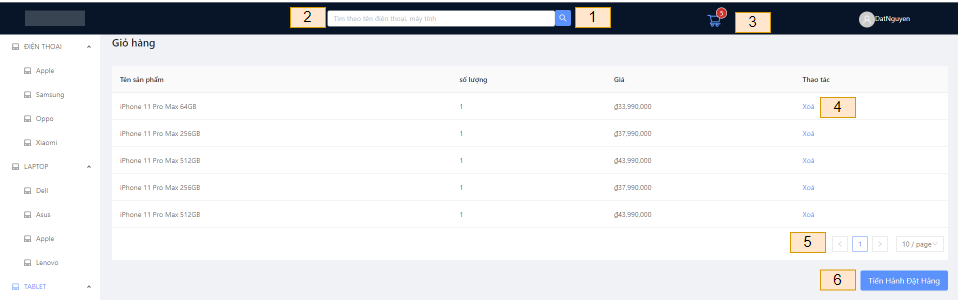
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Nút “Tìm kiếm” | Thẻ <button> |  |
| 2 | Khung tìm kiếm keyword | Thẻ <input> |  |
| 3 | Nút “Giỏ Hàng” | Thẻ <a> |  |
| 4 | Hình sản phẩm | Thẻ <img> |  |
| 5 | Tên sản phẩm | Thẻ <a> |  |
| 6 | Giá sản phẩm | Thẻ <p> |  |
| 7 | Nút chọn mua | Thẻ <button> |  |
| 8 | Thanh menu | Thẻ <ul> <li> |  |

* Thiết kế xử lý

**Bảng 60: Bảng thiết kế xử lý màn hình hiển thị sản phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Tìm | Nhấn vào nút sẽ tìm theo keyword |  |
| 02 | Khung tìm kiếm | Nhập các keyword để tìm kiếm. |  |
| 03 | Giỏ hàng | Khi click sẽ hiện 1 modal hiển thị các sản phẩm đã chọn mua. |  |
| 04 | Hình sản phẩm | Click vào hình sẽ chuyển tới trang sản phẩm chi tiết |  |
| 05 | Tên sản phẩm | Click vào tên sản phẩm sẽ chuyển tới trang sản phẩm chi tiết |  |
| 07 | Nút chọn mưa | Khi click thì sẽ thêm 1 sản phẩm vào giỏ hàng |  |

### MÀN HÌNH GIỎ HÀNG



**Hình 16: Màn hình giỏ hàng**

**Bảng 61: Bảng thông tin các đối tượng trên màn hình giỏ hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Nút “Tìm kiếm” | Thẻ <button> |  |
| 2 | Khung tìm kiếm keyword | Thẻ <input> |  |
| 3 | Nút “Giỏ Hàng” | Thẻ <a> |  |
| 4 | Thao tác “Xóa” | Thẻ <a> |  |
| 5 | Phân trang | Thẻ <ul> |  |
| 6 | Tiến hành đặt hàng | Thẻ <p> |  |

* Thiết kế xử lý

**Bảng 62: Bảng thiết kế xử lý màn hình giỏ hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Tìm | Nhấn vào nút sẽ tìm theo keyword |  |
| 02 | Khung tìm kiếm | Nhập các keyword để tìm kiếm. |  |
| 03 | Giỏ hàng | Khi click sẽ hiện 1 modal hiển thị các sản phẩm đã chọn mua. |  |
| 04 | Thao tác “Xóa” | Click sẽ xóa một sản phẩm khỏi giỏ hàng. |  |
| 05 | Phân trang | Khi số lượng giỏi hàng quá nhiều thì sẽ qua trang tiếp theo |  |
| 06 | Tiến hành đặt hàng | Khi click thì chuyển đến trang tiến hành đặt hàng. |  |

### MÀN HÌNH QUY TRÌNH ĐẶT HÀNG BƯỚC 1

**Hình 17: Màn hình hiển thị quy trình giao hàng bước 1**

**Bảng 63: Bảng thông tin các đối tượng trên màn hình quy trình giao hàng bước 1**

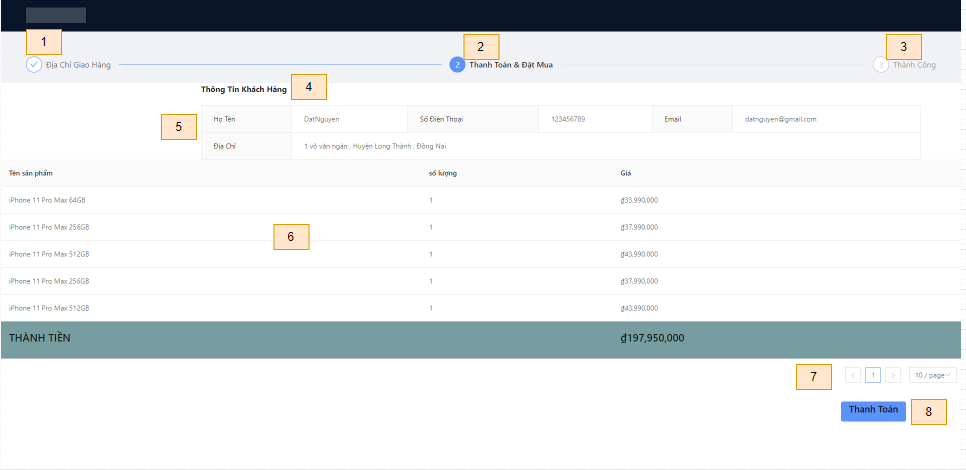
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Địa chỉ giao hàng | Thẻ <div> |  |
| 2 | Thanh toán & đặt mua | Thẻ <div> |  |
| 3 | Thành Công | Thẻ <div> |  |
| 4 | Tiêu đề | Thẻ <p> |  |
| 5 | Bảng thông tin | Thẻ <table> |  |
| 6 | Nút “Giao đến địa chỉ này” | Thẻ <button> |  |

* Thiết kế xử lý

**Bảng 64: Bảng thiết kế xử lý màn hình quy trình giao hàng bước 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Địa chỉ giao hàng | Khi nhấn nút thanh toán ở giỏ hàng |  |
| 02 | Thanh toán & đặt mua | Khi hoàn thành bước “Địa chỉ giao hàng”. |  |
| 03 | Thành Công | Khi hoàn thành bước “Thanh toán & đặt mua”. |  |
| 06 | Nút “Giao đến địa chỉ này” | Click vào “Giao đến địa chỉ này”. Sẽ đi tớ bước “Thanh toán & đặt mua” |  |

### MÀN HÌNH HIỂN THỊ QUY TRÌNH ĐẶT HÀNG BƯỚC 2



**Hình 18: Màn hình hiển thị quy trình giao hàng bước 2**

**Bảng 65: Bảng thông tin các đối tượng trên màn hình quy trình giao hàng bước 2**

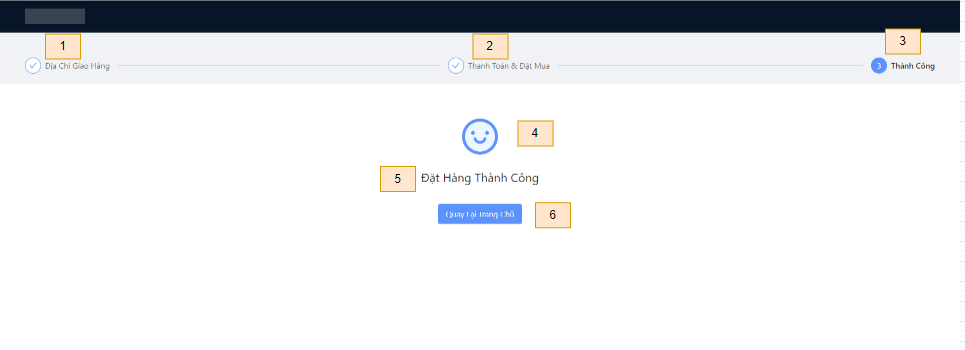
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Địa chỉ giao hàng | Thẻ <div> |  |
| 2 | Thanh toán & đặt mua | Thẻ <div> |  |
| 3 | Thành Công | Thẻ <div> |  |
| 4 | Tiêu đề | Thẻ <p> |  |
| 5 | Bảng thông tin | Thẻ <table> |  |
| 6 | Bảng thông tin sản phẩm mua | Thẻ <table> |  |
| 7 | Phân trang | Thẻ <ul> |  |
| 8 | Nút “Thanh toán” | Thẻ <button> |  |

* Thiết kế xử lý

**Bảng 66: Bảng thiết kế xử lý màn hình quy trình giao hàng bước 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Địa chỉ giao hàng | Khi nhấn nút thanh toán ở giỏ hàng |  |
| 02 | Thanh toán & đặt mua | Khi hoàn thành bước “Địa chỉ giao hàng”. |  |
| 03 | Thành Công | Khi hoàn thành bước “Thanh toán & đặt mua”. |  |
| 07 | Phân trang | Khi số lượng mua nhiều thì sẽ phân trang. |  |
| 08 | Nút “Thanh toán” | Nhấn vào nút “Thanh toán” |  |

### MÀN HÌNH QUẢN LÝ GIAO HÀNG THÀNH CÔNG



**Hình 18: Màn hình hiển thị quy trình giao hàng thành công**

**Bảng 68: Bảng thông tin các đối tượng trên màn hình quy trình giao hàng thành công**

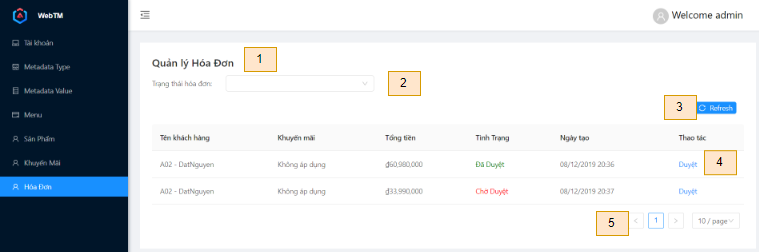
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Địa chỉ giao hàng | Thẻ <div> |  |
| 2 | Thanh toán & đặt mua | Thẻ <div> |  |
| 3 | Thành Công | Thẻ <div> |  |
| 4 | Icon | Thẻ <i> |  |
| 5 | Thông báo | Thẻ <p> |  |
| 6 | Nút “Quay về trang chủ” | Thẻ <button> |  |

* Thiết kế xử lý

**Bảng 69: Bảng thiết kế xử lý màn hình quy trình giao hàng thành công**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 01 | Địa chỉ giao hàng | Khi nhấn nút thanh toán ở giỏ hàng |  |
| 02 | Thanh toán & đặt mua | Khi hoàn thành bước “Địa chỉ giao hàng”. |  |
| 03 | Thành Công | Khi hoàn thành bước “Thanh toán & đặt mua”. |  |
| 06 | Nút “Quay về trang chủ” | Click vào nút sẽ quay về trang chủ |  |

### 4.12 MÀN HÌNH QUẢN LÝ HÓA ĐƠN



**Hình 19: Màn hình quản lý hóa đơn**

**Bảng 70: Bảng thông tin các đối tượng trên màn hình quản lý hóa đơn**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên đối tượng trên giao diện** | **Loại đối tượng trong mã** | **Ghi chú** |
| 1 | Tiêu đề | Thẻ <h2> |  |
| 2 | Combobox trạng thái hóa đơn | Thẻ < select> |  |
| 3 | Nút refesh | Thẻ <button> |  |
| 4 | Thao tác “Duyệt” | Thẻ <a> |  |
| 5 | Phân trang | Thẻ <ul> |  |

* Thiết kế xử lý

**Bảng 71: Bảng thiết kế xử lý màn hình quản lý hóa đơn**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên xử lý** | **Điều kiện thực hiện** | **Ghi chú** |
| 02 | Combobox Trạng thái hóa đơn | Khi click vào combobox sẽ xuất hiện các thẻ cho admin chọn. |  |
| 03 | Refesh | Click vào nút sẽ reset lại bảng dữ liệu |  |
| 04 | Duyệt | Click vào thao tác “Duyệt” sẽ thay đổi trạng thái của hóa đơn. |  |

## 

## CHƯƠNG 5: CÀI ĐẶT VÀ KIỂM THỬ

### 5.1 MÔI TRƯỜNG LẬP TRÌNH

* Hệ điều hành: Window 10
* Visual studio 2015 ++
* Visual code
* Sql Server

### 5.2 KIỂM THỬ

* Nhóm đã lên kế hoạch kiểm thử (lập Test Plan, Test Case chi tiết) đối với website. Vì số lượng chỉ có hai thành viên nên nhóm tổ chức kiểm thử chéo. Thành viên đảm nhiệm lập trình chức năng này sẽ được kiểm thử bởi thành viên còn lại và ngược lại. Dưới đây là một số test case tiêu biểu đã kiểm thử:

#### 5.2.1 Test case màn hình Login

**Bảng 72: Bảng test case màn hình login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test case description** | **Test steps** | **Expected Result** |
| LOG\_01 | Thực hiện đăng đăng nhập một account chưa đăng kí. | 1. Truy cập vào hệ thống. 2. Thực hiện đăng nhập một account chưa đăng kí. | Thông báo lên form “Tài khoản không tồn tại” |
| LOG\_02 | Kiểm tra đăng nhập với password sai. | 1. Truy cập vào hệ thống. 2. Thực hiện đăng nhập với password sai. | Thông báo lên form “Mật khẩu sai vui lòng nhập lại” |
| LOG \_02 | Kiểm tra Guest đăng nhập thành công | 1. Nhập đúng username với pass 2. Nhấn button “Login” | 1. Thông báo đăng nhập thành công 2. Hiển thị màn hình chính của admin. |
| LOG\_03 | Kiểm tra password validation | Nhập password sai validation. | Thông báo lỗi:  “Mật khẩu của bạn chứa kí tự tuần tự |

#### 5.2.2 Test case Thêm sản phẩm

**Bảng 73: Bảng test case Thêm sản phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test case description** | **Test steps** | **Expected Result** |
| Add\_01 | Không nhập mã sản phẩm | 1. Đăng nhập vào trang admin 2. Nhấn button thêm 3. Hiện thị modal để nhập thông tin sản phẩm cần thêm 4. Nhập tất cả thiếu mã sản phẩm | Thông báo lên form “Yêu cầu nhập mã sản phẩm” |
| Add\_02 | Không import ảnh | 1. Đăng nhập vào trang admin 2. Nhấn button thêm 3. Hiện thị modal để nhập thông tin sản phẩm cần thêm 4. Nhập tất cả thiếu ảnh. | Thông báo lên form “Yêu cầu import ảnh” |
| Add \_03 | Nhập vào giá sản phẩm bằng chữ | 1. Đăng nhập vào trang admin 2. Nhấn button thêm 3. Hiện thị modal để nhập thông tin sản phẩm cần thêm 4. Nhập vào ô giá “sadas” | Thông báo lên form “Giá không hợp lệ”. |

#### 5.2.3 Test case Giỏ hàng

**Bảng 74: Bảng test case giỏ hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test case description** | **Test steps** | **Expected Result** |
| Cart\_01 | Khi khách hàng không đăng nhập. | 1. Vào trang sản phẩm 2. Tiến hành “Chọn mua hàng”. | Thông báo lên form “Yêu cầu đăng nhập”. |
| Cart\_02 | Không có hàng hóa trong giỏ hàng | 1. Đăng nhập thành công. 2. Click vào giỏ hàng. | Sẽ không hiển thị button “Tiến hành mua hàng.” |

# KẾT LUẬN

### NHỮNG KẾT QUẢ ĐẠT ĐƯỢC

* Hiểu rõ được Angular 8 CLI, ANT-DESGIN(NG-ZORRO), Typescript.
* Hiểu được ASP .NetCore và áp dụng mô hình MVVM.
* Xây dựng thành công một website thương mại điện tử với những công nghệ đã tìm hiểu.

### ƯU ĐIỂM

* Áp dụng được nhưng kiến thức tìm hiểu vào trong tiểu luận.
* Giao diện thân thiện với người dùng.
* Xây dựng được một website bán hàng cơ bản.
* Việc áp dụng mô hình MVVM làm giảm lượng dữ liệu truyền lên tránh việc Overload.
* Phía Admin đã hoàn thiện các chức năng cơ bản như ( Thêm, Sửa, Xóa, Tìm kiếm) và có thể duyệt hóa đơn.
* Phí người dùng có thể xem danh sách sản phẩm và sản phẩm chi tiết. Đặt hàng và mua hàng.

### NHƯỢC ĐIỂM

* Giao diện phía người dùng chưa được tốt.
* Các chức năng còn hạn chế.
* Thuật toán chưa được tối ưu hóa.

### HƯỚNG PHÁT TRIỂN

* Khắc phục những nhược điểm nêu trên.
* Nhóm sẽ cải thiện giao diện để đạt chuẩn UX/UI.
* Nhóm sẽ thiết kế giao diện để chuẩn responsive trên các thiết bị mobile.
* Bên cạnh đó, cải tiến code để tốc độ xử lý các request và trả response nhanh hơn.
* Xác nhận đơn hàng qua mail.
* Deploy Docker.

# TÀI LIỆU THAM KHẢO

1. https://angular.io/docs
2. <https://tedu.com.vn/lap-trinh-angular-2-can-ban/angular-80-%E2%80%93-co-gi-moi-cach-nang-cap-ra-sao-202.html>.
3. <https://ng.ant.design/docs/getting-started/en>.
4. <https://www.typescriptlang.org/docs/home.html>.
5. <https://www.howkteam.vn/course/lap-trinh-website-voi-aspnet-mvc-co-ban/gioi-thieu-tong-quan-cong-nghe-web-aspnet-mvc-123>
6. <https://www.dammio.com/2018/09/03/asp-net-core-phan-1-gioi-thieu-ve-asp-net-core>
7. <https://viblo.asia/p/gioi-thieu-ve-docker-jOxGdqrXRdm>
8. <https://xuanthulab.net/gioi-thieu-ve-docker-lam-quen-voi-docker-tao-container.html>
9. <https://viblo.asia/p/co-ban-ve-mvvm-model-view-viewmodel-pattern-Do754wdWlM6>
10. <http://stackoverflow.com/>