**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

Đề tài:

**XÂY DỰNG WEBSITE BÁN ĐIỆN THOẠI ÁP DỤNG ELASTICSEARCH**

**Sinh viên thực hiện : Đinh Quý Hải**

**Lớp : ĐH – HTTT1 – K10**

**Mã sinh viên :**

**Giáo viên hướng dẫn : Ts. Ngô Đức Vĩnh**

(Tài liệu lưu hành nội bộ)

**Hà Nội, tháng 11, năm 2019**

# LỜI NÓI ĐẦU

Khi mạng Internet phát triển bùng nổ như hiện nay, có rất nhiều những doanh nghiệp, doanh nhân đã sử dụng Internet để bán những mặt hàng mà họ muốn kinh doanh. Hay nói cách khác những nhà bán buôn, bán lẻ đó, họ đã biết cách mở rộng hình thức kinh doanh của mình thông qua việc xây dựng các Website bán hàng. Song song với việc kinh doanh trực tiếp tại cửa hàng, mà những nhà bán buôn, bán lẻ cũng bán các loại mặt hàng trên Internet thông qua những Website họ xây dựng.

Những Website bán hàng vô cùng đa dạng, có rất nhiều cách hay công nghệ để xây dựng một website. Một trong những nền tảng rất được ưa chuộng hiện nay là ASP.Net Core cùng với hệ thống Search Engine là ElasticSearch. ASP.Net Core là nền tảng lập trình web được viết trên ngôn ngữ lập trình C# cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. ElasticSearch là hệ thống tìm kiếm cho phép người dùng tìm kiếm thông tin một cách nhanh chóng, hỗ trợ thống kê báo cáo.

Dựa vào những nền tảng trên, đề tài “**Xây dựng Website bán điện thoại áp dục ElasticSearch**” của em sẽ được xây dựng bằng ngôn ngữ **C#** trên nền tàng là **ASP.Net Core** với mô hình MVC, kết hợp với ElasticSearh trong tìm kiếm sản phẩm.

Trong quá trình làm đề tài, em vẫn còn những khúc mắc và thiếu xót trong đề tài. Em mong thầy có thể góp ý cho em có thể hoàn thiện đề tài hơn.

Em xin trân thành cảm ơn!

MỤC LỤC

[LỜI NÓI ĐẦU 2](#_Toc27084093)

[MỤC LỤC 3](#_Toc27084094)

[DANH MỤC HÌNH ẢNH SỬ DỤNG TRONG ĐỒ ÁN 6](#_Toc27084095)

[DANH MỤC BẢNG BIỂU SỬ DỤNG TRONG ĐỒ ÁN 7](#_Toc27084096)

[CHƯƠNG 1. KHẢO SÁT HỆ THỐNG 8](#_Toc27084097)

[1.1. Giới thiệu đề tài 8](#_Toc27084098)

[1.2. Hiện trạng 8](#_Toc27084099)

[1.3. Lý do chọn đề tài 9](#_Toc27084100)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 11](#_Toc27084101)

[2.1. Các tác nhân của hệ thống 11](#_Toc27084102)

[2.2. Các ca sử dụng của hệ thống 11](#_Toc27084103)

[2.3. Biểu đồ ca sử dụng 12](#_Toc27084104)

[2.2.1. Biểu đồ phân rã tác nhân Khách hàng 13](#_Toc27084105)

[2.2.2. Biểu đồ phân rã tác nhân Quản trị viên 14](#_Toc27084106)

[2.4. Đặc tả ca sử dụng 14](#_Toc27084107)

[2.4.1. Đăng ký 14](#_Toc27084108)

[2.4.2. Đăng nhập 15](#_Toc27084109)

[2.4.3. Xem thông tin sản phẩm 16](#_Toc27084110)

[2.4.4. Tìm kiếm 17](#_Toc27084111)

[2.4.5. Đánh giá 17](#_Toc27084112)

[2.4.6. Bình luận 18](#_Toc27084113)

[2.4.7. Quản lý giỏ hàng 18](#_Toc27084114)

[2.4.8. Thanh toán 19](#_Toc27084115)

[2.4.9. Quản lý khách hàng 20](#_Toc27084116)

[2.4.10. Quản lý sản phẩm 21](#_Toc27084117)

[2.4.11. Quản lý đơn hàng 23](#_Toc27084118)

[2.5. Biểu đồ trình tự ca sử dụng 24](#_Toc27084119)

[2.5.1. Biểu đồ trình tự cho ca sử dụng “Đăng ký” 25](#_Toc27084120)

[2.5.2. Biểu đồ trình tự cho ca sử dụng “Đăng nhập” 25](#_Toc27084121)

[2.5.3. Biểu đồ trình tự cho ca sử dụng “Xem thông tin sản phẩm” 26](#_Toc27084122)

[2.5.4. Biểu đồ trình tự cho ca sử dụng “Tìm kiếm” 27](#_Toc27084123)

[2.5.5. Biểu đồ trình tự cho ca sử dụng “Đánh giá” 28](#_Toc27084124)

[2.5.6. Biểu đồ trình tự cho ca sử dụng “Bình luận” 28](#_Toc27084125)

[2.5.7. Biểu đồ trình tự cho ca sử dụng “Quản lý giỏ hàng” 29](#_Toc27084126)

[2.5.8. Biểu đồ trình tự cho ca sử dụng “Thanh toán” 30](#_Toc27084127)

[2.5.9. Biểu đồ trình tự cho ca sử dụng “Quản lý khách hàng” 31](#_Toc27084128)

[2.5.10. Biểu đồ trình tự cho ca sử dụng “Quản lý sản phẩm” 32](#_Toc27084129)

[2.5.11. Biểu đồ trình tự cho ca sử dụng “Quản lý đơn hàng” 33](#_Toc27084130)

[2.6. Thiết kế cơ sở dữ liệu 34](#_Toc27084131)

[2.7.1. Mô hình thực thể liên kết 34](#_Toc27084132)

[2.7.2. Chi tiết bảng cơ sở dữ liệu 35](#_Toc27084133)

[2.7. Thiết kế giao diện 43](#_Toc27084134)

[2.8.1. Giao diện trang chủ 43](#_Toc27084135)

[2.8.2. Giao diện đăng nhập 44](#_Toc27084136)

[2.8.3. Giao diện đăng ký 45](#_Toc27084137)

[2.8.4. Giao diện giỏ hàng 45](#_Toc27084138)

[2.8.5. Giao diện chi tiết sản phẩm 46](#_Toc27084139)

[2.8.6. Giao diện trang Quản trị 46](#_Toc27084140)

[2.8.7. Giao diện quản lý người dùng 48](#_Toc27084141)

[2.8.8. Giao diện quản lý sản phẩm 49](#_Toc27084142)

[2.8.9. Giao diện quản lý đặt hàng 50](#_Toc27084143)

[CHƯƠNG 3. CÀI ĐẶT 51](#_Toc27084144)

[3.1. Một số nền tảng công nghệ 51](#_Toc27084145)

[**3.1.1.** **Elasticsearch** 51](#_Toc27084146)

[3.1.2. ASP.Net Core 61](#_Toc27084147)

[3.1.3. JQuery 66](#_Toc27084148)

[3.1.4. Ajax 67](#_Toc27084149)

[3.1.5. Bootstrap 67](#_Toc27084150)

[3.2. Một số kết quả đạt được 67](#_Toc27084151)

[3.2.1. Giao diện trang chủ 67](#_Toc27084152)

[3.2.2. Giao diện đăng nhập 69](#_Toc27084153)

[3.2.3. Giao diện đăng ký 69](#_Toc27084154)

[3.2.4. Giao diện giỏ hàng 70](#_Toc27084155)

[3.2.5. Giao diện chi tiết sản phẩm 71](#_Toc27084156)

[3.2.6. Giao diện quản lý đơn hàng người dùng 72](#_Toc27084157)

[3.2.7. Giao diện chi tiết đơn hàng người dùng 72](#_Toc27084158)

[3.2.8. Giao diện trang quản lý 72](#_Toc27084159)

[3.2.9. Giao diện quản lý người dùng 72](#_Toc27084160)

[3.2.10. Giao diện quản lý sản phẩm 73](#_Toc27084161)

[3.2.11. Giao diện quản lý đơn hàng 73](#_Toc27084162)

[KẾT LUẬN 74](#_Toc27084163)

[TÀI LIỆU THAM KHẢO 75](#_Toc27084164)

# DANH MỤC HÌNH ẢNH SỬ DỤNG TRONG ĐỒ ÁN

[Hình 1 Biểu đồ usecase hệ thống của Website bán điện thoại 12](#_Toc27084052)

[Hình 2 Biểu đồ phân rã usecase Khách hàng 13](#_Toc27084053)

[Hình 3 Biểu đồ phân rã usecase Quản trị viên 14](#_Toc27084054)

[Hình 4 Biểu đồ trình tự chức năng “**Đăng ký**” 25](#_Toc27084055)

[Hình 5 Biểu đồ trình tự chức năng “**Đăng nhập**” 25](#_Toc27084056)

[Hình 6 Biểu đồ trình tự chức năng “**Xem thông tin sản phẩm**” 26](#_Toc27084057)

[Hình 7 Biểu đồ trình tự chức năng “**Tìm kiếm**” 27](#_Toc27084058)

[Hình 8 Biểu đồ trình tự chức năng “**Đánh giá**” 28](#_Toc27084059)

[Hình 9 Biểu đồ trình tự chức năng “**Bình luận**” 28](#_Toc27084060)

[Hình 10 Biểu đồ trình tự chức năng “**Quản lý giỏ hàng**” 29](#_Toc27084061)

[Hình 11 Biểu đồ trình tự chức năng “**Thanh toán**” 30](#_Toc27084062)

[Hình 12 Biểu đồ trình tự chức năng “**Quản lý khách hàng**” 31](#_Toc27084063)

[Hình 13 Biểu đồ trình tự chức năng “**Quản lý sản phẩm**” 32](#_Toc27084064)

[Hình 14 Biểu đồ trình tự chức năng “**Quản lý đơn hàng**” 33](#_Toc27084065)

[Hình 16 Biểu đồ mô hình thực thể liên kết 34](#_Toc27084066)

[Hình 17 Giao diện trang chủ 43](#_Toc27084067)

[Hình 18 Giao diện chức năng “**Đăng nhập**” 44](#_Toc27084068)

[Hình 19 Giao diện chức năng “**Đăng ký**” 45](#_Toc27084069)

[Hình 20 Giao diện giỏ hàng 45](#_Toc27084070)

[Hình 21 Giao diện sản phẩm 46](#_Toc27084071)

[Hình 22 Giao diện trang quản trị 47](#_Toc27084072)

[Hình 23 Giao diện quản lý người dùng 48](#_Toc27084073)

[Hình 24 Giao diện quản lý sản phẩm 49](#_Toc27084074)

[Hình 25 Giao diện quản lý đơn hàng 50](#_Toc27084075)

[Hình 30 ElasticSearch 51](#_Toc27084076)

[Hình 31 Sơ đồ nguyên lý hoạt động của ElasticSearch 52](#_Toc27084077)

[Hình 26 Giao diện website mặc định khi tạo project với ASP.net core 62](#_Toc27084078)

[Hình 27 Mô hình lập trình Convention 64](#_Toc27084079)

[Hình 28 Mô Hình phi trạng thái 65](#_Toc27084080)

[Hình 29 Mô hình trạng thái 66](#_Toc27084081)

[Hình 32 Giao diện trang chủ 68](#_Toc27084082)

[Hình 33 Giao diện chức năng “**Đăng nhập**” 69](#_Toc27084083)

[Hình 34 Giao diện chức năng “**Đăng ký**” 69](#_Toc27084084)

[Hình 35 Giao diện giỏ hàng 70](#_Toc27084085)

[Hình 36 Giao diện chi tiết sản phẩm 71](#_Toc27084086)

[Hình 37 Giao diện quản lý đơn hàng người dùng 72](#_Toc27084087)

[Hình 38 Giao diện chi tiết đơn hàng người dùng 72](#_Toc27084088)

[Hình 39 Giao diện trang quản lý 72](#_Toc27084089)

[Hình 40 Giao diện quản lý người dùng 72](#_Toc27084090)

[Hình 41 Giao diện quản lý sản phẩm 73](#_Toc27084091)

[Hình 42 Giao diện quản lý đơn hàng 73](#_Toc27084092)

# DANH MỤC BẢNG BIỂU SỬ DỤNG TRONG ĐỒ ÁN

[Bảng 1 Thông tin bảng User 35](#_Toc27084038)

[Bảng 2 Thông tin bảng quyền 36](#_Toc27084039)

[Bảng 3 Thông tin bảng đơn hàng 36](#_Toc27084040)

[Bảng 4 Thông tin bảng chi tiết đặt hàng 37](#_Toc27084041)

[Bảng 5 Thông tin bảng chi tiết đánh giá 37](#_Toc27084042)

[Bảng 6 Thông tin bảng chi tiết bình luận 38](#_Toc27084043)

[Bảng 7 Thông tin bảng Sản phẩm 39](#_Toc27084044)

[Bảng 8 Thông tin bảng loại 39](#_Toc27084045)

[Bảng 9 Thông tin bảng Nhà cung cấp 40](#_Toc27084046)

[Bảng 10 Thông tin bảng màu sắc 40](#_Toc27084047)

[Bảng 11 Thông tin bảng quyền người dùng 40](#_Toc27084048)

[Bảng 12 Thông tin bảng thông tin cá nhân 41](#_Toc27084049)

[Bảng 13 Thông tin bảng hình ảnh 41](#_Toc27084050)

[Bảng 13 Thông tin bảng thương hiệu 42](#_Toc27084051)

# CHƯƠNG 1. KHẢO SÁT HỆ THỐNG

## Giới thiệu đề tài

Ngày nay thị trường về điện thoại di động ngày càng tang, nhiều hãng điện thoại ra đời với cấu hình và chất lượng tốt, công nghệ hiện dại.

**Website bán điện thoại** là website giới thiệu đến khách hàng những sản phẩm tốt nhất. Khi khách hàng vào website có thể tìm kiếm và xem thông tin những mặt hàng mà website giới thiệu, nếu khách hàng muốn mua thì có thể cho mặt hàng đó vào giỏ hàng và mua hàng.

## Hiện trạng

Hiện nay trào lưu kinh doanh qua mạng ngày càng trở nên phổ biến. Internet phát triển phát triển kéo theo nhiều dịch vụ phát triển theo đi liền với nó.

Và việc tiếp cận với internet hiện nay không còn mấy xa lạ với con người được trở thành phương tiện truyền thông được nhiều người sử dụng nhất trên thê giới.

Không chỉ dụng internet làm phương tiện truyền thông tin trao đổi, giải trí, xem tin tức, tìm kiếm thông tin… mọi người còn dùng internet để kinh doanh các mặt hàng bình thường đến đặc biệt của mình.

Ở Việt Nam cũng có một số trang web kinh doanh qua mạng <https://cellphones.com.vn/>, <https://hoanghamobile.com/>.... Và nhiều trang web nổi tiếng khác. Các mặt hàng điệ thoại được bán trên trang web này. Tuy nhiên không phải việc kinh doanh qua mạng trở thành 1 trào lưu phát triển mạnh mẽ và thu hút các nhà kinh doanh, mọi lứa tuổi mà vì 1 số lí do như không mất chi phí thuê mặt bằng kinh doanh, phí nhân công bán hàng. Kinh doanh trực tuyến không phải có một nguồn vốn lớn mới có thể kinh doanh mà chỉ cần nguồn vốn đủ để tích luỹ hàng hay một số dịch vụ khác là bạn có thể bắt tay vào việc kinh doanh của mình. Khách hàng chỉ cần mấy thao tác “click chuột vào sản phẩm mà mình yêu thích và liện hệ với chủ cửa hàng là có thể chờ đơn hàng chuyển tới nhà của mình.

Việc kinh doanh qua mạng giúp nhiều công ty có nhiều cơ hội và thách thức hơn trong việc mở rộng thị trường không chỉ thời gian mà cả không gian, trong nước cũng như ngoài nước.

Bên những thuận lợi thì việc kinh doanh qua mạng vân cố những vấn đề nãy sinh. Nó đòi hỏi phải có hiểu biết về công nghệ thông tin để thiết kế và tạo một trang web để thực hiện hoạt động bán hàng và quảng bá sản phẩm của mình.

Một thực tế cho thấy ở Việt Nam cho dù internet và việc mua bán hàng trực tuyến phát triển khá nhanh chóng nhưng rất nhiều người dùng vẫn đang lo sợ với phương thức kinh doanh khá mới mẻ này.

Khách hàng có tâm lí lo ngại khi mua các mặt hàng vì họ không chắc chắn rằng nó tốt như các mẫu sản phẩm tốt như quảng cáo của các nhà kinh doanh hay không. Còn phía bên bán hàng cũng phải tốn 1 chi phí không nhỏ trong việc mua hàng hay sản xuất ra một sản phẩm nào đó mà không thể biết được phản ứng của khách hàng, hay giao hàng mà không có người nhận, hay địa chỉ không chính xác…

Hình thức kinh doanh nào cũng có nhưng ưu điểm và nhược điểm của nó tuy nhiên sự thành công của một hình thức kinh doanh nó còn phụ thuộc vào ý tưởng và cách kinh doanh của người quản lí.

Đối với ngành kinh doanh qua mạng giúp nói chung và bán điệ thoại nói riêng ngày càng đa dạng hơn và phù hợp với xu hướng tất yếu của thời đại hiện nay - thời đại công nghệ thông tin.

## Lý do chọn đề tài

Như chúng ta cũng thấy trong thị trường hiện nay thì việc cạnh tranh về kinh doanh ngày càng trở nên quyết liệt và hầu hết những nhà kinh doanh, những công ty lớn dều rất chú tâm đến việc làm thỏa mãn khách hàng một cách tốt nhất.

So với kinh doanh truyền thống thì TMDT chi phí thấp hơn, hiệu quả đạt cao hơn. Hơn thế nữa, với lợi thế của công nghệ Internet nên việc truyền tải thông tin về sản phẩm nhanh chóng, thuận tiên. Kết hợp với bộ phận giao hàng tận nơi, là thông qua bưu điện và ngân hàng để thanh toán tiền, càng tăng thêm thuận lợi để loại hình này phát triển.

Biết được những nhu cầu đó **Website bán điện thoại được xây dựng trên nền tang ASP.Net Core kết hợp với ElasticSearch** nhằm đáp ứng cho mọi người tiêu dùng trên toàn quốc và thông qua hệ thống Website này họ có thể đặt mua các mặt hàng hay sản phẩm cần thiết.

Do đó, với sự ra đời các website bán hàng qua mạng, mọi người có thể mua mọi thứ hàng hóa mọi lúc mọi nơi mà không cần phải tới tận nơi để mua. Trên thế giới có rất nhiều trang web bán hàng trực tuyến nhưng vẫn chưa được phổ biến rộng rãi. Và khái niệm thương mại điện tử còn khá xa lạ. Trước thực tế đó em đã chọn đề tài: “**Website bán điện thoại áp dụng ElasticSearch**” cho đồ án tốt nghiệp của mình.

# CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Các tác nhân của hệ thống

Hệ thống của Website bán điện thoại có 2 tác nhân chính trong hệ thống:

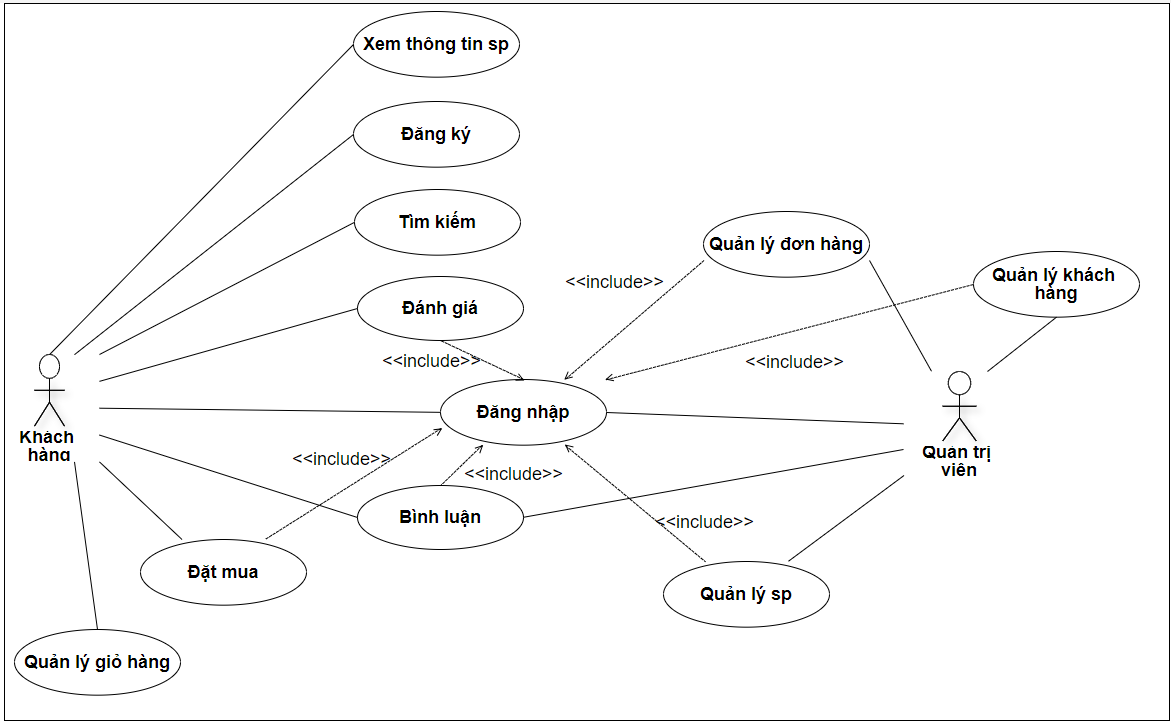
* Khách hàng
* Quản trị viên

## Các ca sử dụng của hệ thống

Hệ thống có các ca sử dụng ứng với mỗi tác nhân chính như sau:

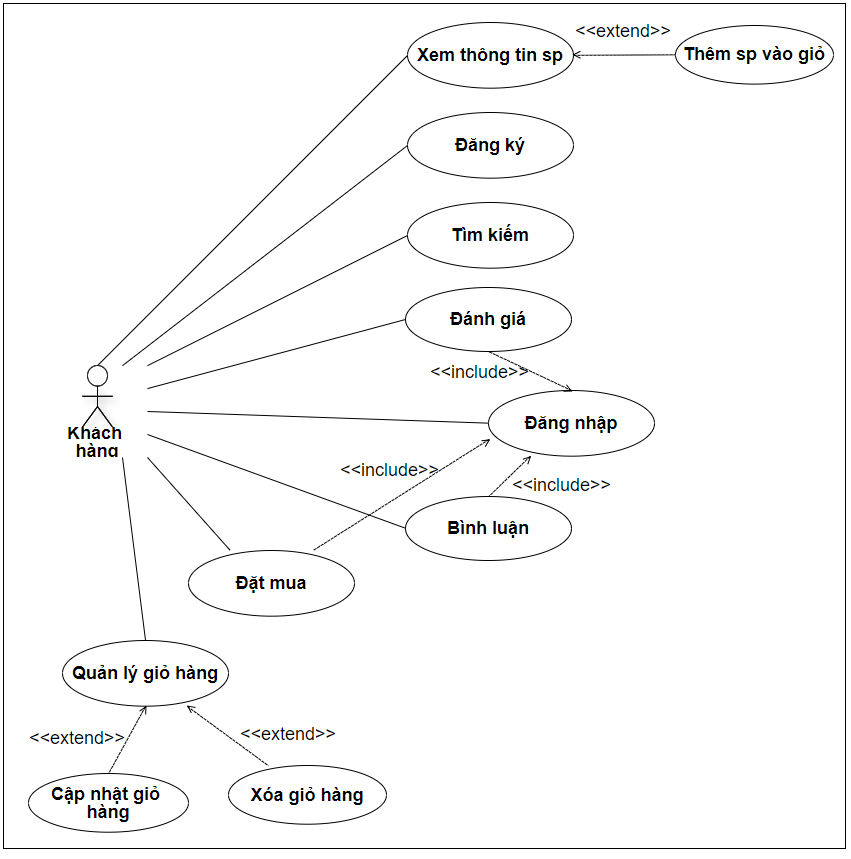
* Khách hàng
* Đăng ký
* Đăng nhập
* Xem thông tin sp
* Tìm kiếm
* Đánh giá
* Bình luận
* Thanh toán
* Quản lý giỏ hàng
* Quản trị viên
* Đăng nhập
* Quản lý đơn hàng
* Quản lý khách hàng
* Quản lý sản phẩm

## Biểu đồ ca sử dụng



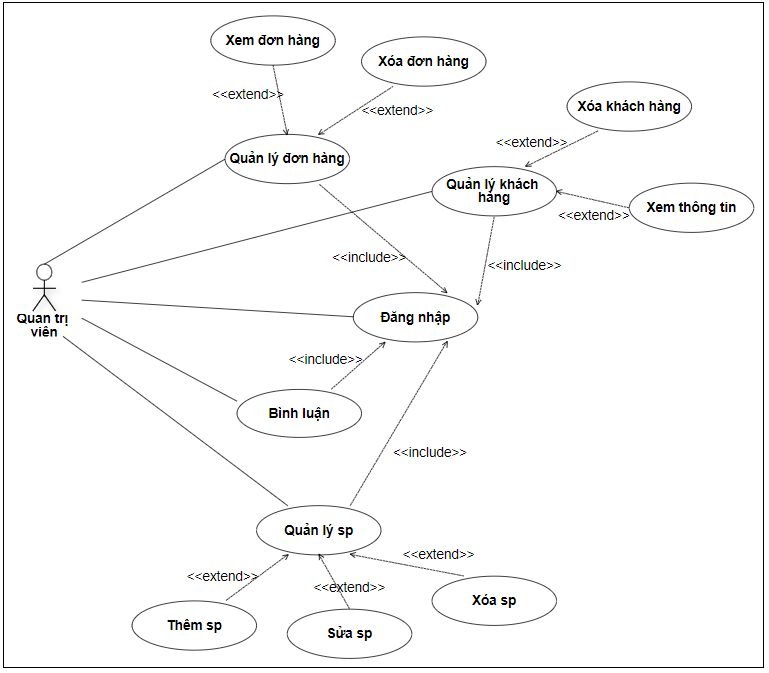
Hình 1 Biểu đồ usecase hệ thống của Website bán điện thoại

### Biểu đồ phân rã tác nhân Khách hàng

****

Hình 2 Biểu đồ phân rã usecase Khách hàng

### Biểu đồ phân rã tác nhân Quản trị viên



Hình 3 Biểu đồ phân rã usecase Quản trị viên

## Đặc tả ca sử dụng

### Đăng ký

1. Mô tả tóm tắt

* Tên ca sử dụng: Đăng ký
* Mục đích: Mô tả công việc khách hàng làm thế nào có thể đăng để có tài khoản trên website.
* Tác nhân: Khách hàng.

1. Tiền điều kiện: Không có
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi tác nhân nhấn chọn đăng ký.
* Sau đó hệ thống sẽ yêu cầu khách hàng điền đầy đủ thông tin của mình vào mẫu gồm có: tên khách hàng, ngày sinh, địa chỉ, số điện thoại và một số thông tin khác.
* Sau khi điền xong thông tin, khách hàng nhấn đăng ký
* Hệ thống sẽ kiểm tra thông tin điền đã hợp lệ hay chưa. Nếu hợp lệ hệ thống sẽ thông báo đăng ký thành công và cấp cho khách hàng mã khách hàng của mình. Nếu không hợp lệ thì thực hiện luồng A1.
* Hệ thống sẽ ghi lại tài khoản của khách hàng.
* Các luồng rẽ nhánh:
* Luồng A1: Nhập thông tin không hợp lệ.
* Hệ thống hiển thị thông báo lỗi.
* Khách hàng có thể chọn hoặc là nhập lại thông tin sao cho hợp lệ hoặc là hủy bỏ đăng ký, khi đó ca sử dụng sẽ kết thúc.

1. Hậu điều kiện: Không có

### Đăng nhập

1. Mô tả tóm tắt

* Tên ca sử dụng: Đăng nhập
* Mục đích: Mô tả cách thức User đăng nhập vào hệ thống.
* Tác nhân: Quản trị viên, Khách hàng.

1. Tiền điều kiện: Không có
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng bắt đầu khi User chọn đăng nhập vào hệ thống.
* Hệ thống sẽ hiển thị giao diện để User điền thông tin: tài khoản, mật khẩu.
* User sẽ điền tài khoản được cấp và mật khẩu của mình vào.
* User chọn nút đăng nhập. Nếu thành công, hệ thống sẽ thông báo đăng nhập thành công. Nếu lỗi sẽ thực hiện luồng A1.
* Luồng rẽ nhánh: A1, đăng nhập không thành công:
* Hệ thống thông báo tài khoản hoặc mật khẩu nhập sai.
* User nhập lại tài khoản hoặc mật khẩu nếu muốn tiếp tục đăng nhập.
* User không muốn tiếp tục đăng nhập và chọn thoát.
* Ca sử dụng kết thúc.

1. Hậu điều kiện: Nếu đăng nhập thành công, User sẽ đăng nhập được vào hệ thống.

### Xem thông tin sản phẩm

1. Mô tả tóm tắt

* Tên ca sử dụng: Xem thông tin sản phẩm
* Mục đích: Hiển thị thông tin sản phẩm cho user.
* Tác nhân: Quản trị viên, khách hàng.

1. Tiền điều kiện: Không có
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng bắt đầu khi user click vào một sản phẩm để xem chi tiết thông tin sản phẩm.
* Hệ thống sẽ hiển thị giao diện thông tin chi tiết của sản phẩm.
* Nếu tác nhân là khách hàng thì sẽ hiển thị nút “thêm vào giỏ”
* Nếu khách hàng click vào nút “thêm vào giỏ”, luồng sự kiện con thêm vào giỏ được thực hiện.
* Thêm vào giỏ:
* Sản phẩm sẽ được đưa vào giỏ hàng.
* Hệ thống sẽ tự động cập nhật số lượng trong giỏ hàng trên thanh menu, bên phải icon giỏ hàng

1. Hậu điều kiện: Không có

### Tìm kiếm

1. Mô tả tóm tắt

* Tên ca sử dụng: Tìm kiếm
* Mục đích: Mô tả cách thức user tìm kiếm sản phẩm.
* Tác nhân: Quản trị viên, khách hàng.

1. Tiền điều kiện: Không có
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng bắt đầu khi user muốn tìm kiếm sản phẩm trên website.
* Hệ thống sẽ hiển thị giao diện để user có thể tìm kiếm sản phẩm, trên giao diện có một vùng tìm kiếm để user có thể gõ từ khóa muốn tìm kiếm.
* User có thể gõ từ tìm kiếm theo các cách: theo tên hàng, nhà cung cấp, thương hiệu, sản phẩm mới nhất, loại hàng.
* Sau đó user nhấn nút “Tìm kiếm”. Nếu lỗi sẽ thực hiện luồng A1.
* Luồng rẽ nhánh: A1, tìm kiếm không thành công:
* Từ khóa user tìm kiếm không hợp lệ hoặc là sản phẩm hoặc là sản phẩm đó không có trong website.
* Ca sử dụng kết thúc khi user không muốn tìm kiếm.

1. Hậu điều kiện: Không có.

### Đánh giá

1. Mô tả tóm tắt

* Tên ca sử dụng: Đánh giá
* Mục đích: Mô tả cách thức user đánh giá sản phẩm.
* Tác nhân: khách hàng.

1. Tiền điều kiện: Đăng nhập thành công vào hệ thống
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng bắt đầu khi user muốn đánh giá sản phẩm
* Hệ thống sẽ hiển thị giao diện để user có thể đánh giá sản phẩm, trên giao diện có hiển thị 5 ngôi sao.
* User có thể click vào các ngôi sao để thể hiện độ thích hoặc không thích dựa vào số lượng ngôi sao.

1. Hậu điều kiện: Sản phẩm được đánh giá

### Bình luận

1. Mô tả tóm tắt

* Tên ca sử dụng: Bình luận
* Mục đích: Mô tả cách thức user bình luận sản phẩm.
* Tác nhân: Quản trị viên, khách hàng.

1. Tiền điều kiện: Đăng nhập thành công vào hệ thống
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng bắt đầu khi user muốn bình luận sản phẩm
* Hệ thống sẽ hiển thị giao diện để user có thể đánh giá sản phẩm, bên dưới trang chi tiết sản phẩm sẽ hiển thị cho user phần bình luận.
* User có thể viết bình luận cho sản phẩm mà user muốn bình luận.

1. Hậu điều kiện: Sản phẩm được bình luận.

### Quản lý giỏ hàng

1. Mô tả tóm tắt

* Tên ca sử dụng: Quản lý giỏ hàng.
* Mục đích: Ca sử dụng này cho phép Khách hàng quản lý giỏ hàng của mình
* Tác nhân: Khách hàng.

1. Tiền điều kiện: Không có
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi Khách hàng muốn cập nhật hoặc xóa sản phẩm trong giỏ hàng
* Hệ thống hiển thị các lựa chọn.
* Hệ thống yêu cầu Khách hàng chọn công việc mà họ muốn thực hiện.
* Nếu Khách hàng chọn “cập nhật sản phẩm trong giỏ”, luồng sự kiện con cập nhật sản phẩm trong giỏ sẽ được thực hiện.
* Nếu Khách hàng chọn “xóa sản phẩm trong giỏ”, luồng sự kiện con xóa sản phẩm trong giỏ được thực hiện.
* Cập nhật sản phẩm trong giỏ:
* Hệ thống hiển thị phần cập nhật số lượng sản phẩm trong giỏ
* Khách hàng có thể tăng giảm số lượng của sản phẩm trong giỏ
* Xóa sản phẩm trong giỏ:
* Hệ thống hiển thị nút xóa sản phẩm trong giỏ hàng.
* Khách hàng có thể xóa sản phẩm trong giỏ
* Trong trường hợp Khách hàng xóa hết sản phẩm trong giỏ thì hệ thống sẽ tự động quay lại trang chủ của hệ thống

1. Hậu điều kiện

Sản phẩm trong giỏ hàng được cập nhật hoặc xóa thành công

### Thanh toán

1. Mô tả tóm tắt

* Tên ca sử dụng: Thanh toán.
* Mục đích: Ca sử dụng này cho phép Khách hàng thanh toán
* Tác nhân: Khách hàng.

1. Tiền điều kiện: Đăng nhập thành công vào hệ thống
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi Khách hàng muốn thanh toán.
* Hệ thống hiển thị các lựa chọn hình thức thanh toán.
* Hệ thống yêu cầu Khách hàng chọn hình thức thanh toán mà họ muốn thực hiện.
* Hệ thống sẽ hiển thị lại thông tin của khách hàng để xác nhận
* Khách hàng xác nhận lại và thanh toán

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thì khách hàng đặt hàng thành công

### Quản lý khách hàng

1. Mô tả tóm tắt

* Tên ca sử dụng: Quản lý khách hàng.
* Mục đích: Ca sử dụng này cho phép Quản trị viên duy trì thông tin khách hàng trong hệ thống. Quản trị viên có thể thêm và xóa khách hàng từ hệ thống.
* Tác nhân: Quản trị viên.

1. Tiền điều kiện: Đăng nhập thành công vào hệ thống
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi Quản trị viên muốn thêm hoặc xóa khách hàng từ hệ thống.
* Hệ thống hiển thị các lựa chọn.
* Hệ thống yêu cầu Quản trị viên chọn công việc mà họ muốn thực hiện.
* Nếu Quản trị viên chọn “thêm một khách hàng”, luồng sự kiện con thêm một khách hàng sẽ được thực hiện.
* Nếu Quản trị viên chọn “xóa một khách hàng”, luồng sự kiện con xóa một khách hàng được thực hiện.
* Thêm một khách hàng:
* Hệ thống yêu cầu Quản trị viên nhập thông tin khách hàng, bao gồm: Tên khách hàng, email, địa chỉ, số điện thoại…
* Quản trị viên nhập thông tin được yêu cầu.
* Hệ thống tạo ra mã số tự động và gán cho khách hàng.
* Một khách hàng mới được thêm vào hệ thống.
* Hệ thống cung cấp cho Quản trị viên mã số mới của khách hàng.
* Xóa một khách hàng:
* Hệ thống yêu cầu Quản trị viên nhập mã số khách hàng.
* Quản trị viên nhập vào mã số sản phẩm.
* Hệ thống kiểm tra mã số khách hàng vừa nhập có tồn tại hay không. Nếu không tồn tại thì thực hiện luồng A1.
* Hệ thống hiển thị thông tin về khách hàng có mã số vừa nhập.
* Hệ thống nhắc Quản trị viên xác nhận việc xóa khách hàng.
* Quản trị viên xác nhận việc xóa.
* Khách hàng đó sẽ bị xóa khỏi hệ thống.
* Các luồng rẽ nhánh
* Luồng A1: nhập mã số khách hàng không tồn tại.
* Hệ thống thông báo lỗi
* Hệ thống yêu cầu Quản trị viên nhập vào mã số khách hàng khác hoặc hủy bỏ thao tác xóa.
* Nếu Quản trị viên nhập mã số khách hàng khác thì quay lại luồng xóa.
* Nếu Quản trị viên chọn hủy, ca sử dụng kết thúc.

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thông tin về khách hàng sẽ được thêm vào hay xóa khỏi hệ thống. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa đổi.

### Quản lý sản phẩm

1. Mô tả tóm tắt

* Tên ca sử dụng: Quản lý sản phẩm.
* Mục đích: Ca sử dụng này cho phép Quản trị viên duy trì thông tin hàng trong hệ thống. Quản trị viên có thể thêm, sửa và xóa thông tin sản phẩm từ hệ thống.
* Tác nhân: Quản trị viên.

1. Tiền điều kiện: Đăng nhập thành công vào hệ thống
2. Các luồng sự kiện

* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi Quản trị viên muốn thêm, sửa hoặc xóa thông tin sản phẩm từ hệ thống.
* Hệ thống hiển thị các lựa chọn.
* Hệ thống yêu cầu Quản trị viên chọn công việc mà họ muốn thực hiện.
* Nếu Quản trị viên chọn “thêm một sản phẩm”, luồng sự kiện con thêm một sản phẩm sẽ được thực hiện.
* Nếu Quản trị viên chọn “sửa thông tin sản phẩm”, luồng sự kiện con sửa thông tin một sản phẩm sẽ được thực hiện.
* Nếu Quản trị viên chọn “xóa thông tin một sản phẩm”, luồng sự kiện con xóa một sản phẩm được thực hiện.
* Thêm một sản phẩm:
* Hệ thống yêu cầu Quản trị viên nhập thông tin sản phẩm, bao gồm: Tên sản phẩm, nhà cung cấp, số lượng hiện có, đơn giá, một số thông tin khác,…
* Quản trị viên nhập thông tin được yêu cầu.
* Hệ thống tạo ra mã số tự động và gán cho sản phẩm.
* Một sản phẩm mới được thêm vào hệ thống.
* Hệ thống cung cấp cho Quản trị viên mã số mới của sản phẩm.
* Sửa một sản phẩm:
* Hệ thống yêu cầu Quản trị viên nhập mã số của sản phẩm.
* Quản trị viên nhập vào mã số của sản phẩm.
* Hệ thống kiểm tra mã số sản phẩm vừa nhập có tồn tại hay không. Nếu không tồn tại thì thực hiện luồng A1.
* Hệ thống hiển thị thông tin về sản phẩm có mã số vừa nhập.
* Quản trị viên sửa những thông tin cần thiết về sản phẩm: tên, số lượng hiện có, đơn giá, nhà cung cấp,… và xác nhận việc sửa đổi.
* Hệ thống cập nhật những thay đổi vừa được thực hiện.
* Xóa một sản phẩm:
* Hệ thống yêu cầu Quản trị viên nhập mã số sản phẩm.
* Quản trị viên nhập vào mã số sản phẩm.
* Hệ thống kiểm tra mã số sản phẩm vừa nhập có tồn tại hay không. Nếu không tồn tại thì thực hiện luồng A1.
* Hệ thống hiển thị thông tin về sản phẩm có mã số vừa nhập.
* Hệ thống nhắc Quản trị viên xác nhận việc xóa sản phẩm.
* Quản trị viên xác nhận việc xóa.
* Sản phẩm đó sẽ bị xóa khỏi hệ thống.
* Các luồng rẽ nhánh
* Luồng A1: nhập mã số sản phẩm không tồn tại.
* Hệ thống thông báo lỗi
* Hệ thống yêu cầu Quản trị viên nhập vào mã số sản phẩm khác hoặc hủy bỏ thao tác xóa hoặc sửa.
* Nếu Quản trị viên nhập mã số sản phẩm khác thì quay lại luồng xóa hoặc sửa.
* Nếu Quản trị viên chọn hủy, ca sử dụng kết thúc.

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, thông tin về sản phẩm sẽ được thêm vào, sửa đổi hay xóa khỏi hệ thống. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa đổi.

### Quản lý đơn hàng

1. Mô tả tóm tắt

* Tên ca sử dụng: Quản lý đơn hàng.
* Mục đích: Ca sử dụng này cho phép Quản trị viên duy trì thông tin đơn hàng trong hệ thống. Quản trị viên có thể xem và xóa thông tin đơn hàng từ hệ thống.
* Tác nhân: Quản trị viên.

1. Tiền điều kiện: Đăng nhập thành công vào hệ thống
2. Các luồng sự kiện

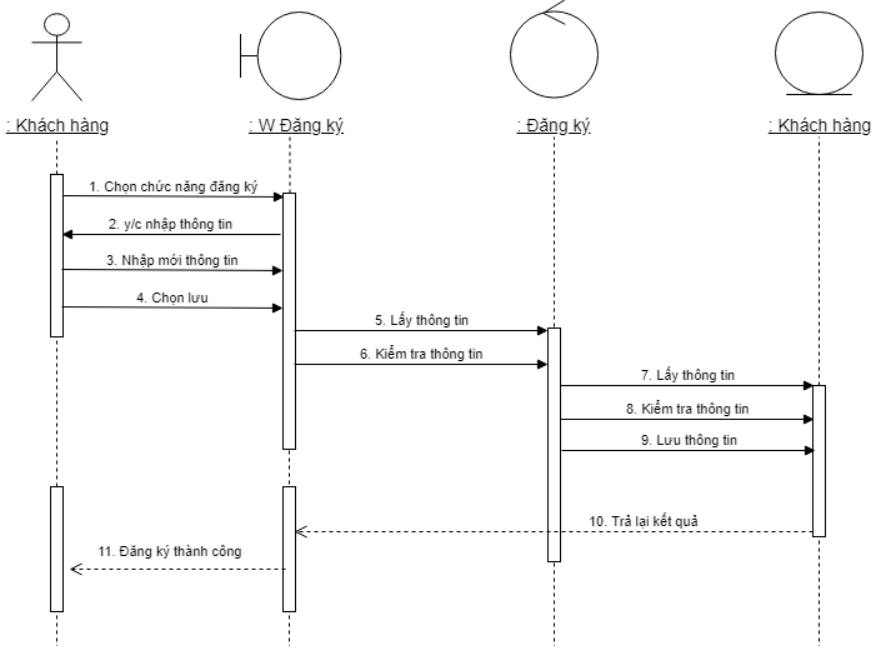
* Luồng sự kiện chính: Ca sử dụng này bắt đầu khi Quản trị viên muốn xem hoặc xóa thông tin đơn hàng từ hệ thống.
* Hệ thống hiển thị các lựa chọn.
* Hệ thống yêu cầu Quản trị viên chọn công việc mà họ muốn thực hiện.
* Nếu Quản trị viên chọn “xem một đơn hàng”, luồng sự kiện con xen một đơn hàng sẽ được thực hiện.
* Nếu Quản trị viên chọn “xóa một đơn hàng”, luồng sự kiện con xóa đơn hàng sẽ được thực hiện.
* Xem một đơn hàng:
* Hệ thống yêu cầu Quản trị viên nhập mã số của đơn hàng.
* Quản trị viên nhập vào mã số của đơn hàng.
* Hệ thống kiểm tra mã số đơn hàng vừa nhập có tồn tại hay không. Nếu không tồn tại thì thực hiện luồng A1.
* Hệ thống hiển thị thông tin về đơn hàng có mã số vừa nhập.
* Xóa một đơn hàng:
* Hệ thống yêu cầu Quản trị viên nhập mã số đơn hàng.
* Quản trị viên nhập vào mã số đơn hàng.
* Hệ thống kiểm tra mã số đơn hàng vừa nhập có tồn tại hay không. Nếu không tồn tại thì thực hiện luồng A1.
* Hệ thống hiển thị thông tin về đơn hàng có mã số vừa nhập.
* Hệ thống nhắc Quản trị viên xác nhận việc xóa đơn hàng.
* Quản trị viên xác nhận việc xóa.
* Đơn hàng đó sẽ bị xóa khỏi hệ thống.
* Các luồng rẽ nhánh
* Luồng A1: nhập mã đơn hàng không tồn tại.
* Hệ thống thông báo lỗi
* Hệ thống yêu cầu Quản trị viên nhập vào mã số đơn hàng khác hoặc hủy bỏ thao tác xóa hoặc sửa.
* Nếu Quản trị viên nhập mã số đơn hàng khác thì quay lại luồng xóa hoặc xem.
* Nếu Quản trị viên chọn hủy, ca sử dụng kết thúc.

1. Hậu điều kiện

Nếu ca sử dụng này được thực hiện thành công, đơn hàng sẽ được hiển thị thông tin hoặc xóa khỏi hệ thống. Trong các trường hợp khác, hệ thống ở trong trạng thái chưa đổi.

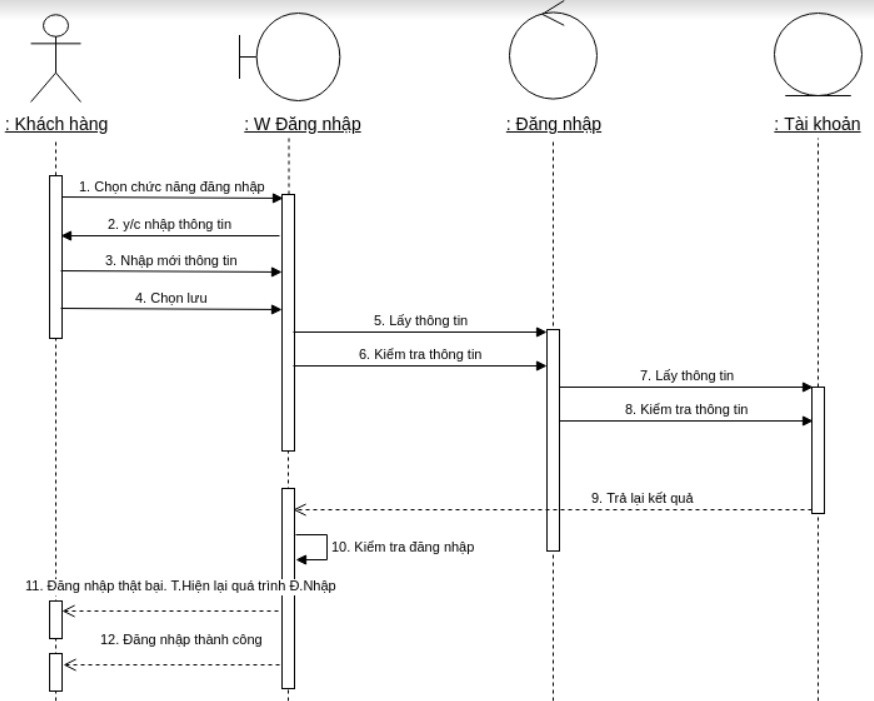
## Biểu đồ trình tự ca sử dụng

### Biểu đồ trình tự cho ca sử dụng “Đăng ký”



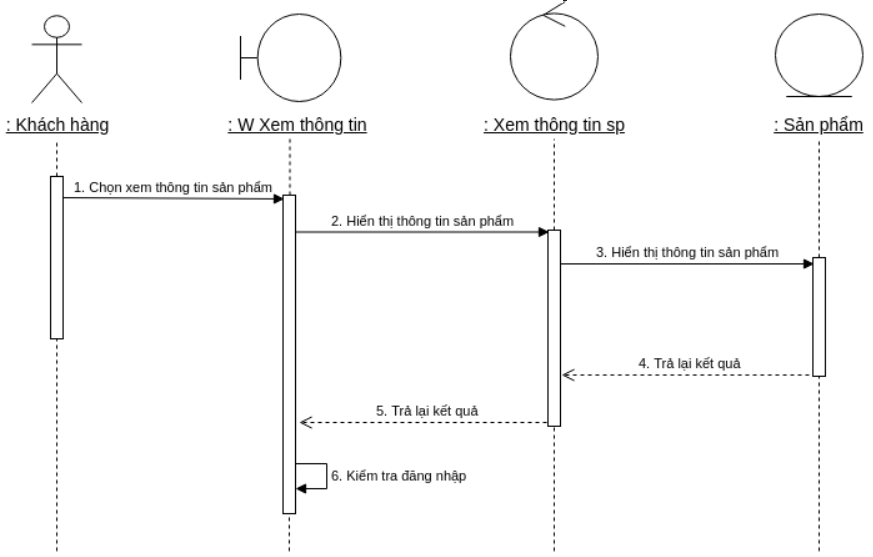
Hình 4 Biểu đồ trình tự chức năng “**Đăng ký**”

### Biểu đồ trình tự cho ca sử dụng “Đăng nhập”



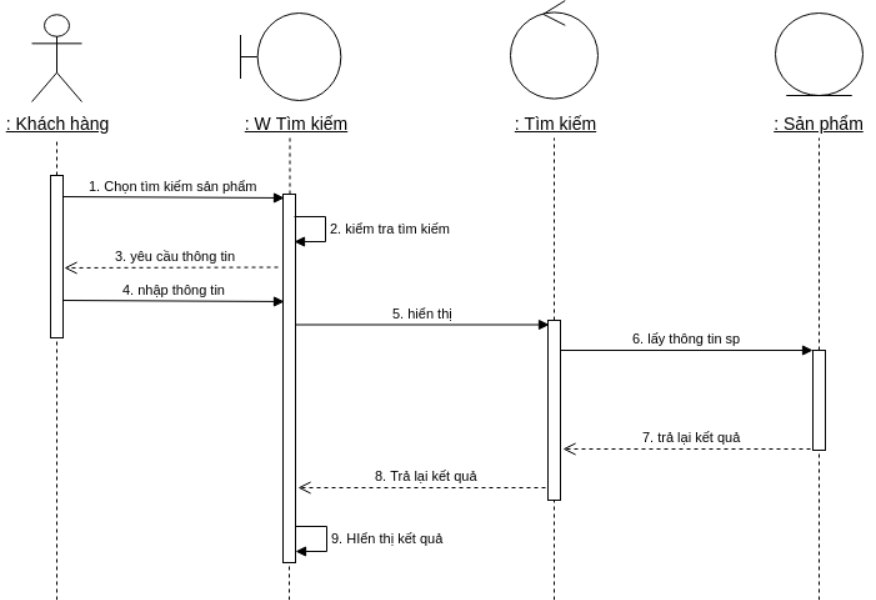
Hình 5 Biểu đồ trình tự chức năng “**Đăng nhập**”

### Biểu đồ trình tự cho ca sử dụng “Xem thông tin sản phẩm”



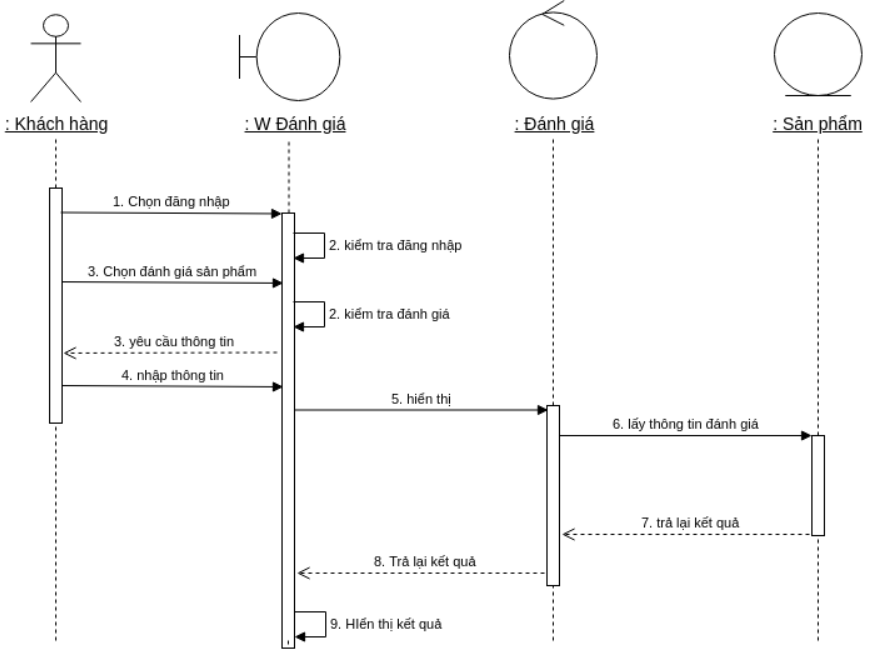
Hình 6 Biểu đồ trình tự chức năng “**Xem thông tin sản phẩm**”

### Biểu đồ trình tự cho ca sử dụng “Tìm kiếm”



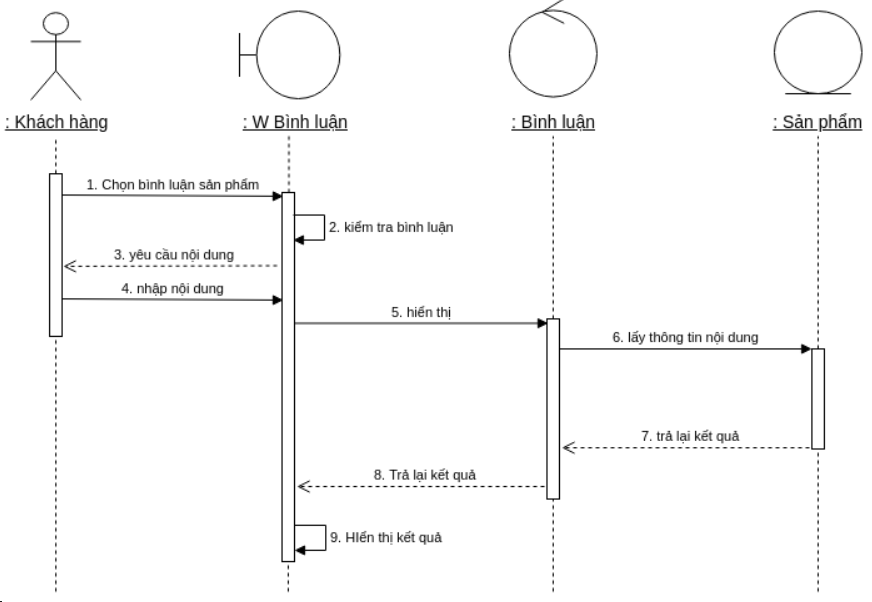
Hình 7 Biểu đồ trình tự chức năng “**Tìm kiếm**”

### Biểu đồ trình tự cho ca sử dụng “Đánh giá”



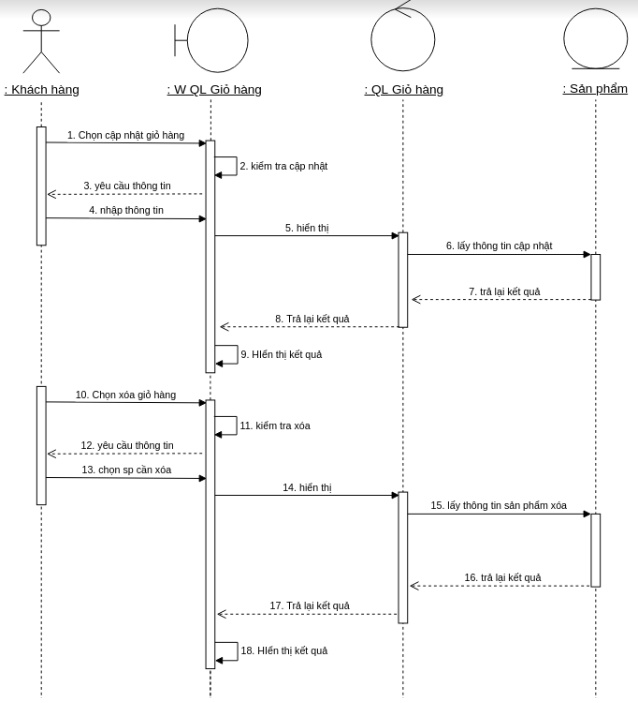
Hình 8 Biểu đồ trình tự chức năng “**Đánh giá**”

### Biểu đồ trình tự cho ca sử dụng “Bình luận”



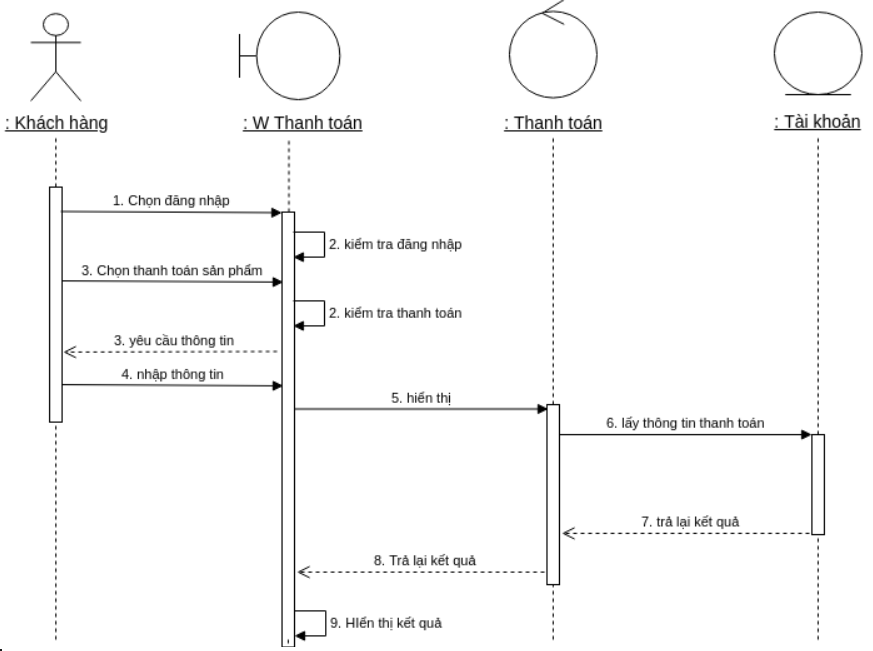
Hình 9 Biểu đồ trình tự chức năng “**Bình luận**”

### Biểu đồ trình tự cho ca sử dụng “Quản lý giỏ hàng”



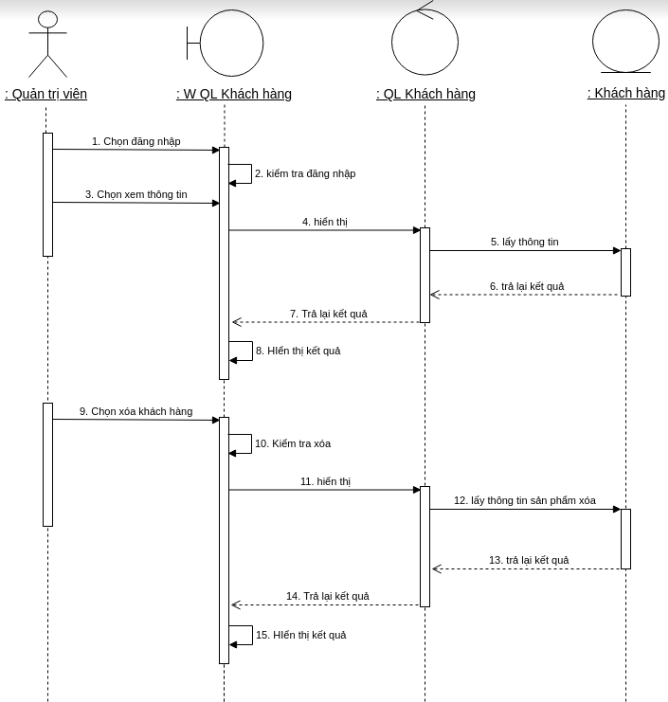
Hình 10 Biểu đồ trình tự chức năng “**Quản lý giỏ hàng**”

### Biểu đồ trình tự cho ca sử dụng “Thanh toán”



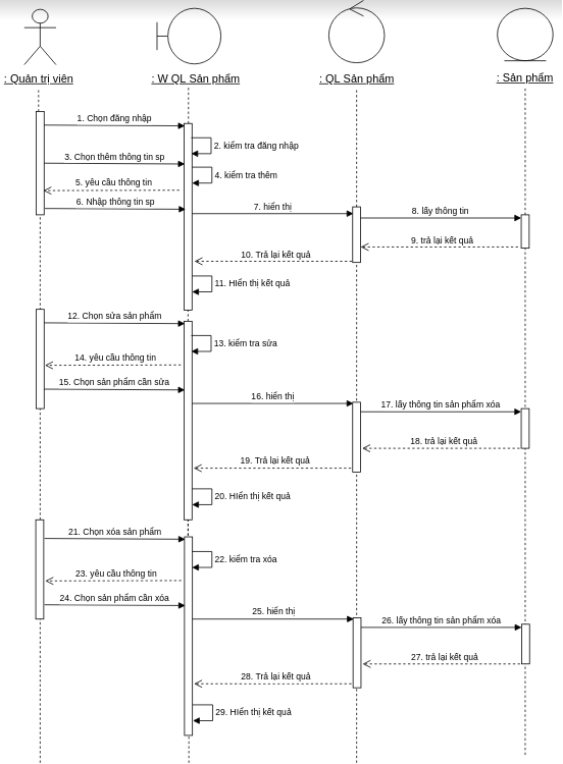
Hình 11 Biểu đồ trình tự chức năng “**Thanh toán**”

### Biểu đồ trình tự cho ca sử dụng “Quản lý khách hàng­”



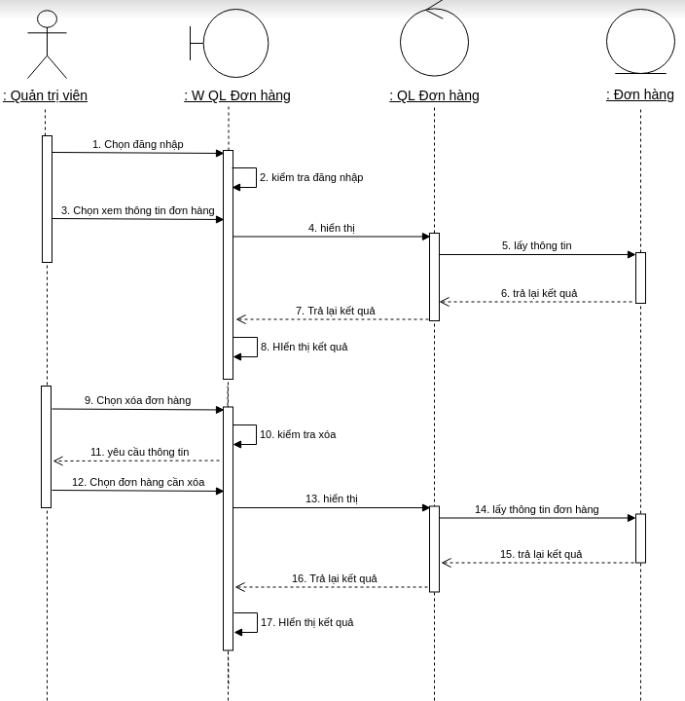
Hình 12 Biểu đồ trình tự chức năng “**Quản lý khách hàng**”

### Biểu đồ trình tự cho ca sử dụng “Quản lý sản phẩm”



Hình 13 Biểu đồ trình tự chức năng “**Quản lý sản phẩm**”

### Biểu đồ trình tự cho ca sử dụng “Quản lý đơn hàng”



Hình 14 Biểu đồ trình tự chức năng “**Quản lý đơn hàng**”

## Thiết kế cơ sở dữ liệu

### Mô hình thực thể liên kết



Hình 16 Biểu đồ mô hình thực thể liên kết

Đây là biểu đồ mô hình thực thể liên kết. Là nơi lưu trữ những dữ liệu của hệ thống, bao gồm như thông tin người dùng(Users), thông tin mặt hàng(Products), thông tin đơn hàng(Orders), nội dung bình luận(Comments),...

### Chi tiết bảng cơ sở dữ liệu

* **Bảng AspNetUsers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Int | Not Null | PK | Mã khách hàng |
| 2 | UserName | String | Not Null |  | Tên đăng nhập |
| 3 | NormalizedUserName | String | Not Null |  |  |
| 4 | Email | String |  |  | Email |
| 5 | NormalizedEmail | String |  |  |  |
| 6 | EmailConfirmed | boolean |  |  |  |
| 7 | PasswordHash | String |  |  | Mật khẩu |
| 8 | SecurityStamp | String |  |  |  |
| 9 | ConcurrencyStamp | String |  |  |  |
| 10 | PhoneNumber | String |  |  | Số điện thoại |
| 11 | PhoneNumberConfirmed | boolean |  |  |  |
| 12 | TwoFactorEnabled | boolean |  |  |  |
| 13 | LockoutEnd | Datetime |  |  |  |
| 14 | LockoutEnabled | boolean |  |  |  |
| 15 | AccessFailedCount | boolean |  |  |  |
| 16 | CreatedDate | Datetime |  |  |  |
| 17 | ModifyDate | Datetime |  |  |  |

Bảng 1 Thông tin bảng User

* **Bảng AspNetRoles**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Int | Not Null | PK | Mã phân quyền |
| 2 | Name | String | Not Null |  | Tên quyền |
| 3 | NormalizedName | String |  |  |  |
| 4 | ConcurrencyStamp | String |  |  |  |
| 5 | CreatedDate | Datetime |  |  |  |
| 2 | ModifyDate | Datetime |  |  |  |

Bảng 2 Thông tin bảng quyền

* **Bảng Orders**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Int | Not Null | PK | Mã đơn hàng |
| 2 | UserId | Int | Not Null | FK | Mã khách hàng |
| 3 | order\_date | Datetime | Not Null |  | Ngày dặt hàng |
| 4 | ReviceverName | String | Not Null |  | Người nhận |
| 5 | Email | String | Not Null |  | Email người nhận |
| 6 | PhoneNumber | String | Not Null |  | Số điện thoại |
| 7 | ReviceverAddress | String | Not Null |  | Địa chỉ nhận |
| 8 | Status | int |  |  | Trạng thái đơn |
|  | CreatedDate |  |  |  |  |
|  | ModifyDate |  |  |  |  |

Bảng 3 Thông tin bảng đơn hàng

* **Bảng OrderDetails**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Int | Not Null | PK | Mã chi tiết đơn hàng |
| 2 | OrderId | Int | Not Null | FK | Mã đơn hàng |
| 3 | ProductId | Int | Not Null | FK | Mã sản phẩm |
| 4 | Quantity | Int | Not Null |  | Số lượng sản phẩm |
| 5 | SalePice | Int | Not Null |  | Giá bán |
| 6 | Status | Int | Not Null |  | Trạng thái sản phẩm trong đơn |
| 7 | CreatedDate |  |  |  |  |
| 8 | ModifyDate |  |  |  |  |

Bảng 4 Thông tin bảng chi tiết đặt hàng

* **Bảng Rates**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Int | Not Null | PK | Mã đánh giá |
| 2 | Star | Int | Not Null |  | Giá trị đánh giá |
| 3 | UserId | Int | Not Null | FK | Mã user |
| 4 | ProductId | Int | Not null | FK | Mã sản phẩm |
| 5 | CreatedDate |  |  |  |  |
| 6 | ModifyDate |  |  |  |  |

Bảng 5 Thông tin bảng chi tiết đánh giá

* **Bảng Comments**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Int | Not Null | PK | Mã đánh giá |
| 2 | Value | String | Not Null |  | Nội dung |
| 3 | UserId | Int | Not Null | FK | Mã user |
| 4 | ProductId | Int | Not null | FK | Mã sản phẩm |
| 5 | Title |  |  |  |  |
| 6 | CreatedDate |  |  |  |  |
| 7 | ModifyDate |  |  |  |  |

Bảng 6 Thông tin bảng chi tiết bình luận

* **Bảng Products**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Guid | Not Null | PK | Mã sản phẩm |
| 2 | Title | String | Not null |  | Tiêu đề |
| 3 | Name | String | Not Null |  | Tên sản phẩm |
| 4 | Trademark | String | Not Null |  | Thương hiệu |
| 5 | Model | String | Not Null |  | Dòng điện thoại |
|  | Status | int |  |  |  |
|  | CategoryId | String | Not Null |  |  |
|  | TrademarkId | String | Not Null |  |  |
|  | Shortescription |  | Not Null |  |  |
|  | Specification | String | Not Null |  |  |
|  | Price | decimal | Not Null |  |  |
|  | OldPrice | decimal |  |  |  |
|  | ImageThumbnail | String | Not Null |  |  |
|  | Quantity | Int | Not Null |  |  |
|  | Gtin | String |  |  |  |

Bảng 7 Thông tin bảng Sản phẩm

* **Bảng Categories**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Int | Not Null | PK | Mã loại |
| 2 | Name | String | Not Null |  | Tên loại |
| 3 | ParentId |  |  |  |  |
| 4 | CreatedDate |  |  |  |  |
| 5 | ModifyDate |  |  |  |  |

Bảng 8 Thông tin bảng loại

* **Bảng Providers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Int | Not Null | PK | Mã NSX |
| 2 | Name | String | Not Null |  | Tên NSX |
| 3 | Address | String | Not Null |  | Địa chỉ |
| 4 | Phone | Int | Not null |  | Số đt |
| 5 | Status | Bool | Not null |  | Trạng thái |
| 6 | CreatedDate |  |  |  |  |
| 7 | ModifyDate |  |  |  |  |

Bảng 9 Thông tin bảng Nhà cung cấp

* **Bảng Colors**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | Int | Not Null | PK |  |
| 2 | Name | String | Not Null |  |  |
| 3 | Description | String | Not Null |  |  |
| 4 | Status | Int | Not null |  |  |
| 6 | CreatedDate | Datetime |  |  |  |
| 7 | ModifyDate | Datetime |  |  |  |

Bảng 10 Thông tin bảng màu sắc

* Bảng AspNetUserRoles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | UserId | String | Not Null | PK |  |
| 2 | RoleId | String | Not Null | PK |  |
| 3 | CreatedDate | Datetime | Not Null |  |  |
| 4 | ModifyDate | Datetime | Not null |  |  |

Bảng 11 Thông tin bảng quyền người dùng

* Bảng UserInfos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | String | Not Null | PK |  |
| 2 | UserId | String | Not Null |  |  |
| 2 | LastName | String |  |  |  |
| 3 | FirstName | String |  |  |  |
| 4 | Address | String |  |  |  |
| 5 | CreatedDate | Datetime | Not Null |  |  |
| 6 | ModifyDate | Datetime | Not null |  |  |

Bảng 12 Thông tin bảng thông tin cá nhân

* Bảng Images

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | String | Not Null | PK |  |
| 2 | Name | String | Not Null |  |  |
| 3 | Url | String |  |  |  |
| 4 | CreatedDate | String |  |  |  |
| 5 | ModifiedDate |  |  |  |  |
| 6 | Status | String |  |  |  |
| 7 | ProductId | Datetime | Not Null |  |  |

Bảng 13 Thông tin bảng hình ảnh

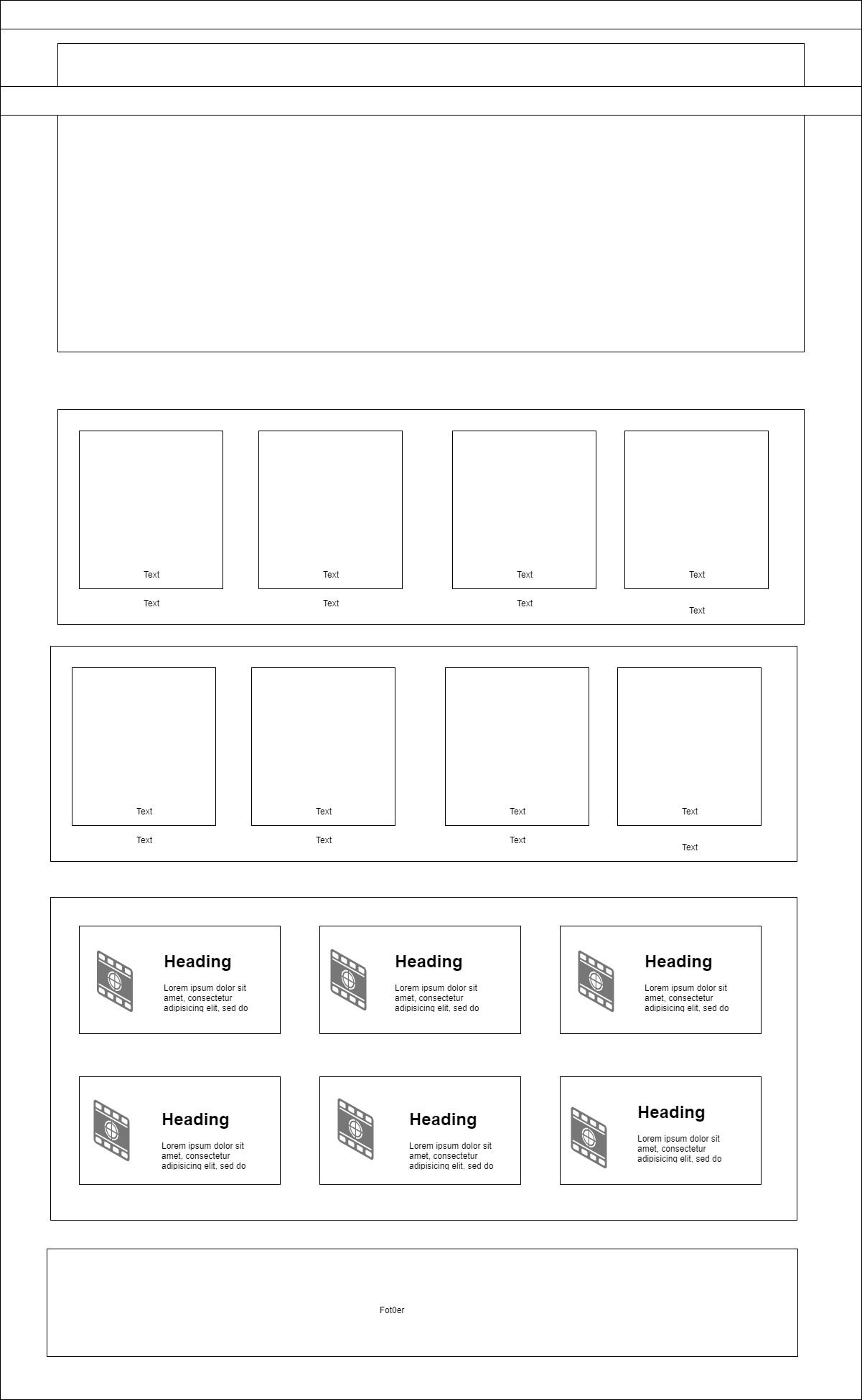
Bảng Trademark

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu | Ràng buộc | PK/FK | Mô tả |
| 1 | Id | String | Not Null | PK |  |
| 2 | Name | String | Not Null |  |  |
| 3 | CreatedDate | String |  |  |  |
| 4 | ModifiedDate |  |  |  |  |
| 5 | Status | String |  |  |  |
| 6 | Description | Datetime | Not Null |  |  |

Bảng 13 Thông tin bảng thương hiệu

## Thiết kế giao diện

### Giao diện trang chủ



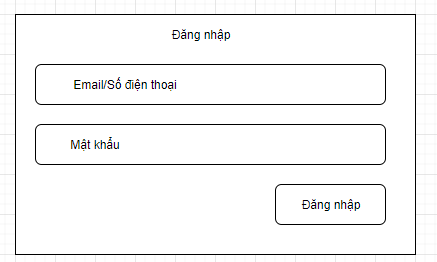
Hình 17 Giao diện trang chủ

Thiết kế giao diện trang chủ được hiển thị với 1 thanh Navbar để hiển thị các đường dẫn cho một số chức năng. Tiếp theo sẽ là 1 khung banner chuyển động để hiển thị hình ảnh các sản phẩm đang giảm giá và các sản phẩm mới nhất.

Bên dưới sẽ hiển thị danh sách các điện thoại được bán chạy nhất, điện thoại mới nhất và các sản phẩm mà trang web đang kinh doanh.

Cuối cùng sẽ có 1 thanh Footer để hiển thị một số thông tin của trang web và thông tin liên lạc

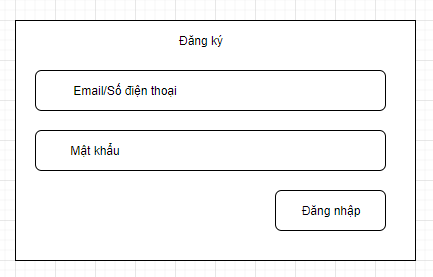
### Giao diện đăng nhập



Hình 18 Giao diện chức năng “**Đăng nhập**”

Giao diện cho chức năng Đăng nhập sẽ gồm có 1 bức ảnh Background và Form để nhập thông tin đăng nhập như Email và Mật khẩu.

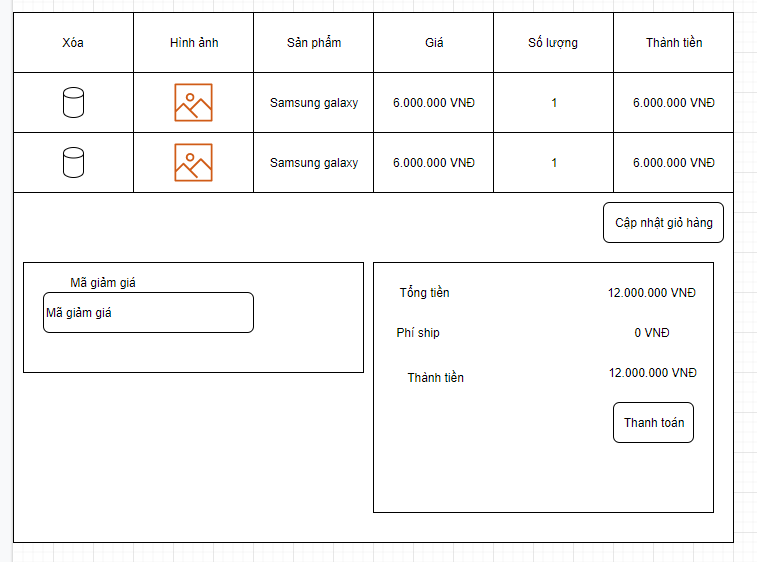
### Giao diện đăng ký



Hình 19 Giao diện chức năng “**Đăng ký**”

Giao diện cho chức năng Đăng ký sẽ gồm có 1 bức ảnh Background và Form để nhập thông tin đăng ký như Email/SĐT, mật khẩu

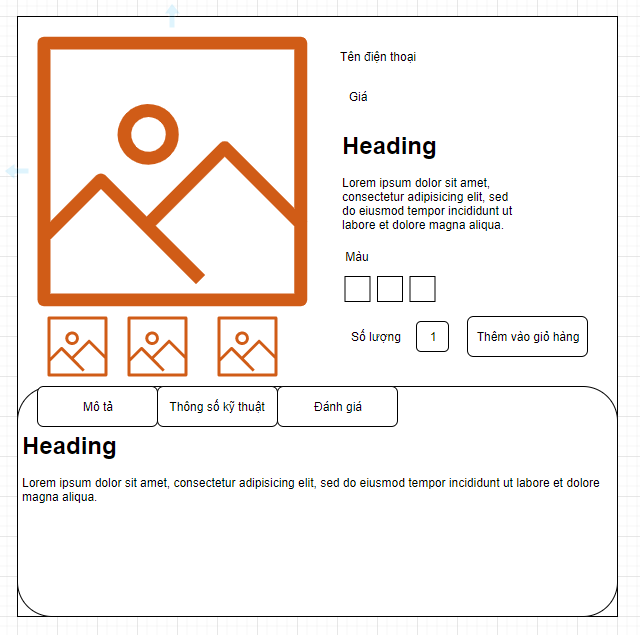
### Giao diện giỏ hàng



Hình 20 Giao diện giỏ hàng

Giao diện giỏ hàng gồm có danh dách những sản phẩm mà người dùng đã thêm vào. Trong bảng này, người dùng có thể thực hiện một số thao tác như xóa sản phẩm khỏi giỏ, cập nhật số lượng sản phẩm, link tới trang chi tiết sản phẩm. Bên cạnh đó, người dùng có thể nhìn thấy 1 ô để nhập phiếu mua hàng để được giảm giá. Và đặc biệt là phần hiển thị Tổng giỏ hàng, sẽ thống kê Tổng số lượng sản phẩm và Tổng tiền trên giỏ hàng.

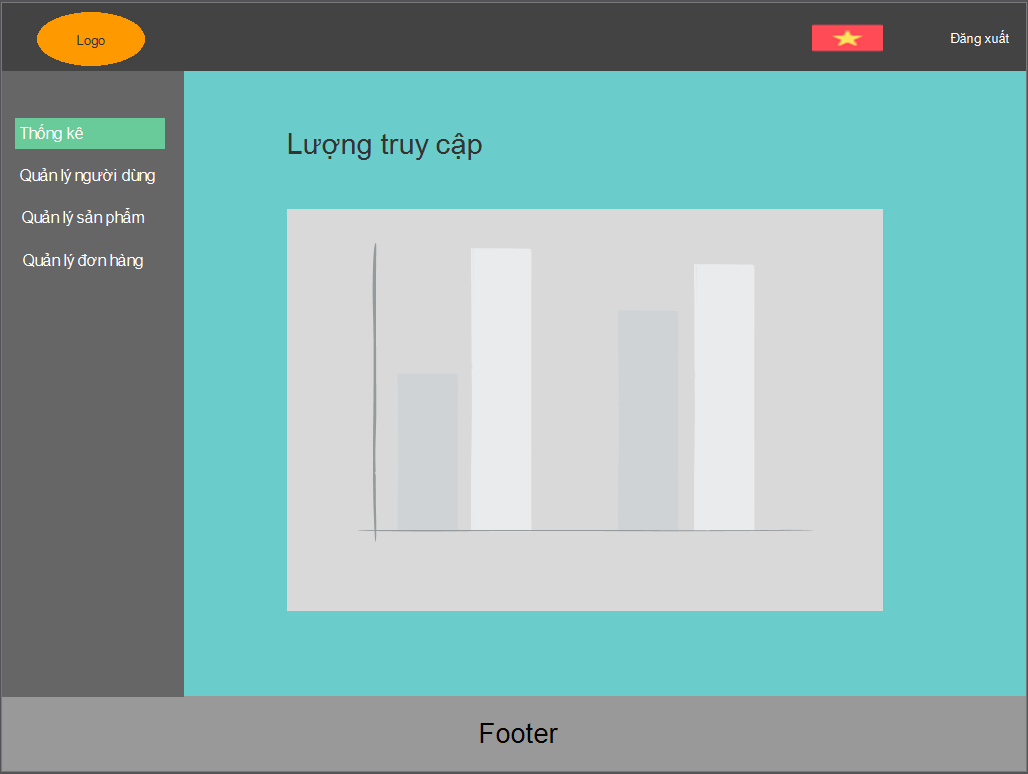
### Giao diện chi tiết sản phẩm



Hình 21 Giao diện sản phẩm

Giao diện sản phẩm sẽ hiển thị thông tin chi tiết sản phẩm của hệ thống.

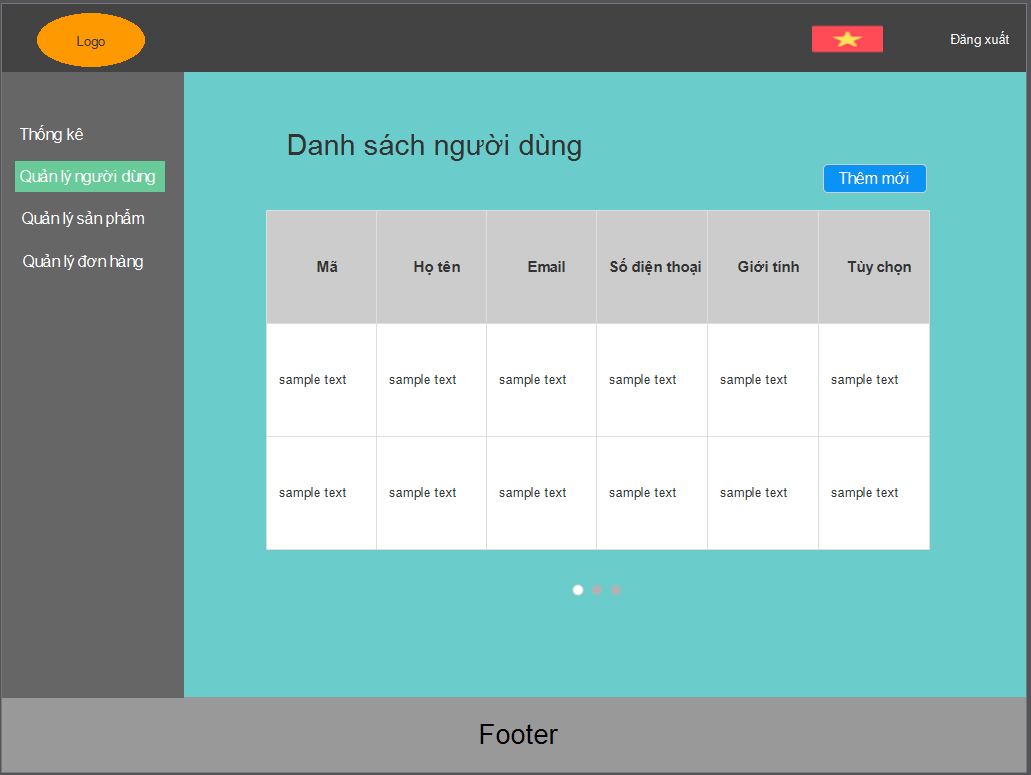
### Giao diện trang Quản trị



Hình 22 Giao diện trang quản trị

Giao diện trang quản trị mặc định là phần Thống kê lượng truy cập, trong trang này người dùng sẽ thấy 1 biểu đồ thể hiện số lượng người truy cập trang web.

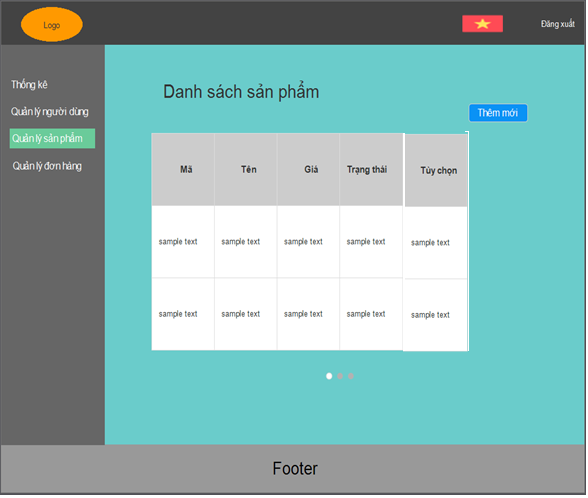
### Giao diện quản lý người dùng



Hình 23 Giao diện quản lý người dùng

Giao diện quản lý người dùng sẽ hiển thị danh sách các tài khoản người dùng gồm các thông tin như, tên, email, số điện thoại,... khi đã đăng ký thành công vào hệ thống

### Giao diện quản lý sản phẩm



Hình 24 Giao diện quản lý sản phẩm

Giao diện trang quản lý sản phẩm hiển thị danh sách các sản phẩm của hệ thống, hiển thị đầy đủ thông tin của sản phẩm như, tên, giá, …

### Giao diện quản lý đặt hàng



Hình 25 Giao diện quản lý đơn hàng

Giao diện quản lý đơn hàng sẽ hiển thị các đơn hàng bao gồm các thông tin như họ tên, tên sản phẩm, số lượng, giá,... mà người dùng đã thanh toán thành công

# CHƯƠNG 3. CÀI ĐẶT

## Một số nền tảng công nghệ

* + 1. **Elasticsearch**
       - 1. ElasticSearch là gì?



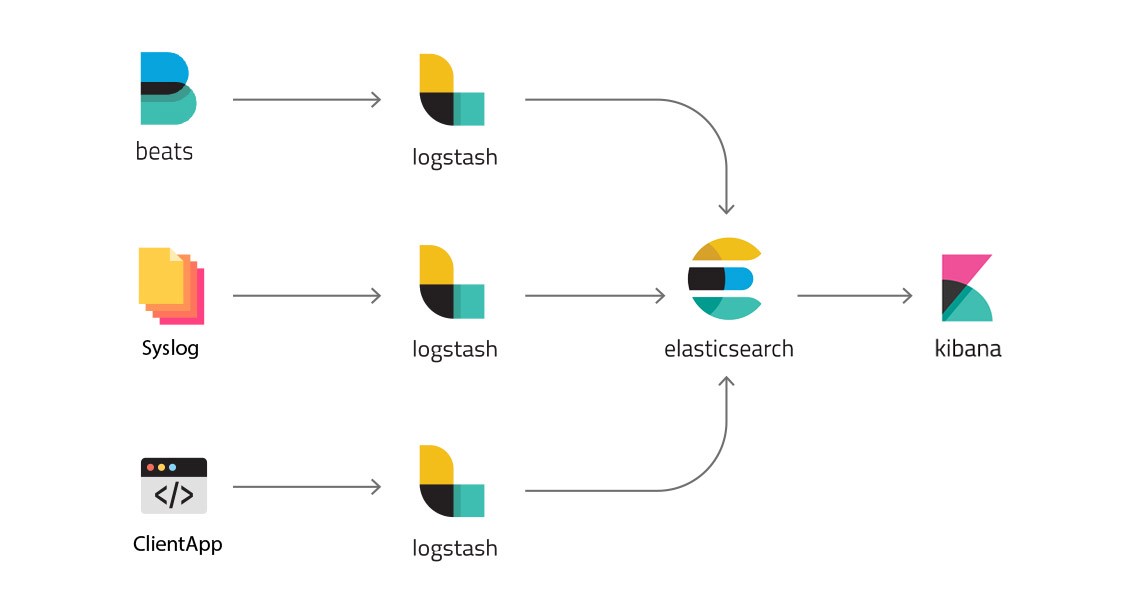
Hình 30 ElasticSearch

ElasticSearcch là một công cụ tìm kiếm mã nguồn mở dựa trên nền tảng Apache Lucene và phát hành phiên bản đầu tiên vào năm 2010 bởi Elasticsearch N.V. ElasticSearch cung cấp các API đơn giản, tính phân tán, tốc độ và khả năng mở rộng là trung tâm của Elastic Stack, một hệ thống mã nguồn mở để tìm kiếm và phân tích dữ liệu đưa ra các biểu đồ trực quan với người dùng. Thường được gọi là ELKStack (ElasticSearch, Logstash, Kibana ElasticSearch). Được phát triển bằng Java và được phát hành giạng nguồn mở giấy phép Apache.

* ElasticSearch là một Search Engine
* ElasticSearch được kế thừa từ Lucene Apache
* ElasticSearch thực chất là hoạt động như một web server, có khả năng tìm kiếm nhanh chóng thông qua giao thức RESTful
* ElasticSearch có khả năng phân tích và thống kê dữ liệu
* ElasticSearch chạy trên server riêng và đồng thời giao tiếp thông qua RESTful do vậy nó không phụ thuộc vào client viết bằng gì hay hiện tại hệ thống của bạn viết bằng gì. Nên việc tích hợp nó vào hệ thống của bạn là dễ dàng, bạn chỉ cần gửi request http lên là nó trả về kết quả.
* ElasticSearch là 1 hệ thống phân tán và có khả năng mở rộng tuyệt vời.
* ElasticSearch là một open source được phát triển bằng Java
  + - * 1. ElasticSearch dùng để làm gì?

Vì tốc độ và khả năng mở rộng của ElasticSearch và khả năng lập chỉ mục với nhiều loại nội dung nên ElasticSearch được sử dụng cho một số trường hợp sau:

* Tìm kiếm ứng dụng
* Tìm kiếm website
* Tìm kiếm doanh nghiệp
* Ghi log và phân tích log
* Giám sát cơ sở hạ tầng và container
* Giám sát hiệu suất ứng dụng
* …
  + - * 1. ElasticSearch hoạt động như thế nào?



Hình 31 Sơ đồ nguyên lý hoạt động của ElasticSearch

* **ClientApp, Syslog, Beats:**Dữ liệu và nhật ký cần phân tích.
* **Logstash:** Thu thập dữ liệu. Thậm chí phân tích cú pháp và biến đổi dữ liệu.
* **ElasticSearch:**Dữ liệu được chuyển đổi từ Logstash và lưu trữ, tìm kiếm và đánh index.
* **Kibana: Kibana sử dụng ElasticSearch để chuyển đổi dữ liệu thành dạng biểu đồ và chia sẽ nó.**
  + - * 1. Index trong ElasticSearch?

ElasticSearch Index là một tập hợp các tài liệu có liên quan đến nhau. Lưu trữ dưới dạng Json (Key-Values).

ElasticSearch sử dụng cấu trúc dữ liệu gọi là Inverted Index, nó được thiết kế để cho phép tìm kiếm toàn văn bản rất nhanh (Full-text search).

* + - * 1. Logtash hoạt động như thế nào?

Tổng hợp và xử lý dữ liệu và chuyển nó đến ElasticSearch.

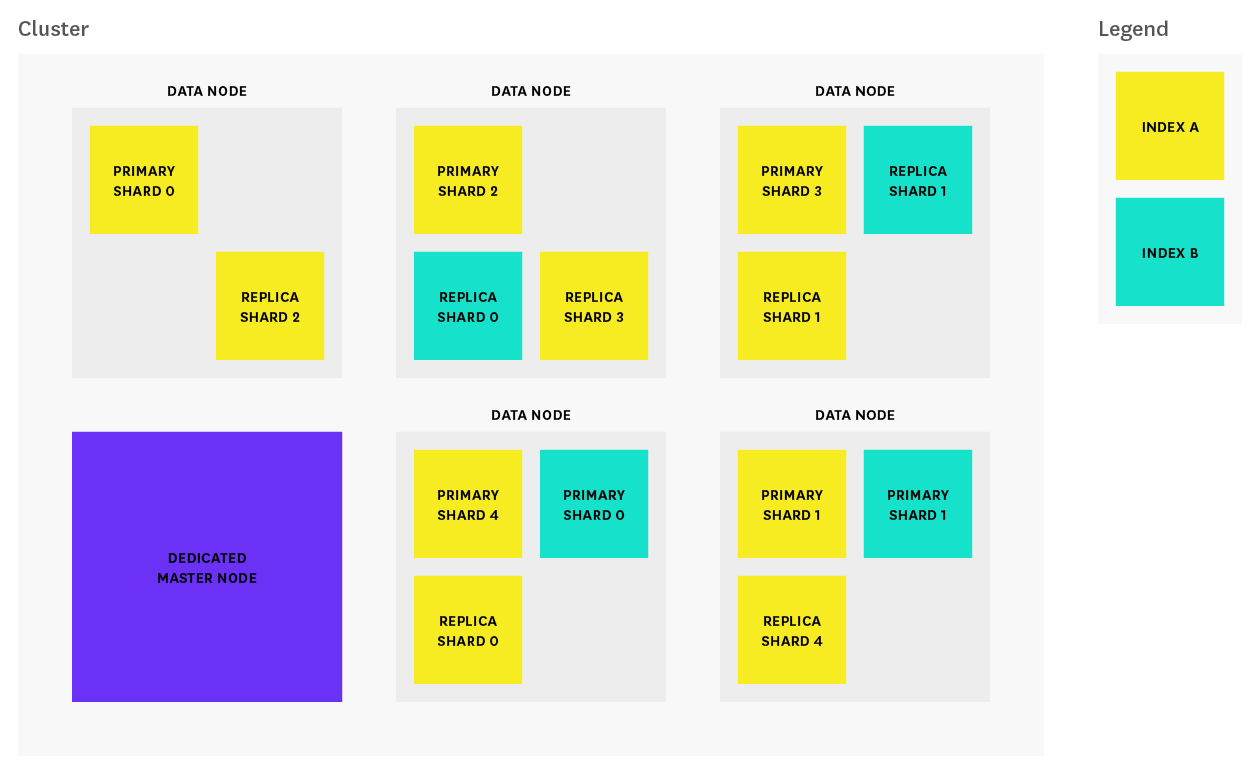
* + - * 1. Kibana hoạt động như thế nào?

Kibana là một công cụ cung cấp giao diện đồ họa cho ElasticSearch bao gồm các biểu đồ thời gian thực, biểu đồ đường, …. Cho phép người dung tạo biểu đồ dựa trên dữ liệu của họ.

* + - * 1. Tại sao lại sử dụng ElasticSearch?
        2. Các khái niệm liên quan đến ElasticSearch?
* **Document:** Document là một JSON object với một số dữ liệu. Đây là basic information unit trong ES. Hiểu 1 cách cơ bản thì đây là đơn vị nhỏ nhất để lưu trữ dữ liệu trong Elasticsearch.
* Index: Trong Elasticsearch, sử dụng một cấu trúc được gọi là inverted index. Nó được thiết kế để cho phép tìm kiếm full-text search. Cách thức của nó khá đơn giản, các văn bản được phân tách ra thành từng từ có nghĩa sau đó sẽ đk map xem thuộc văn bản nào. Khi search tùy thuộc vào loại search sẽ đưa ra kết quả cụ thể.
* **Shard:** Shard là đối tượng của Lucene, là tập con các documents của 1 Index. Một Index có thể được chia thành nhiều shard. Mỗi node bao gồm nhiều Shard. Chính vì thế Shard mà là đối tượng nhỏ nhất, hoạt động ở mức thấp nhất, đóng vai trò lưu trữ dữ liệu. Chúng ta gần như không bao giờ làm việc trực tiếp với các Shard vì Elasticsearch đã support toàn bộ việc giao tiếp cũng như tự động thay đổi các Shard khi cần thiết. Có 2 loại Shard là: primary shard và replica shard.
* **Node và Cluster:**

**+ Node:** Là trung tâm hoạt động của Elasticsearch. Là nơi lưu trữ dữ liễu,tham gia thực hiện đánh index cúa cluster cũng như thực hiện các thao tác tìm kiếm Mỗi node được định danh bằng 1 unique name

**+ Cluster:** Tập hợp các nodes hoạt động cùng với nhau, chia sẽ cùng thuộc tính cluster.name. Chính vì thế Cluster sẽ được xác định bằng 1 'unique name'. Việc định danh các cluster trùng tên sẽ gây nên lỗi cho các node vì vậy khi setup các bạn cần hết sức chú ý điểm này Mỗi cluster có một node chính (master), được lựa chọn một cách tự động và có thể thay thế nếu sự cố xảy ra. Một cluster có thể gồm 1 hoặc nhiều nodes. Các nodes có thể hoạt động trên cùng 1 server. Tuy nhiên trong thực tế, một cluster sẽ gồm nhiều nodes hoạt động trên các server khác nhau để đảm bảo nếu 1 server gặp sự cố thì server khác (node khác) có thể hoạt động đầy đủ chức năng so với khi có 2 servers. Các node có thể tìm thấy nhau để hoạt động trên cùng 1 cluster qua giao thức unicast.



* + - * 1. Ưu nhược điểm của ElasticSearch
* Ưu điểm
* Tìm kiếm dữ liệu rất nhanh chóng, mạnh mẽ dựa trên Apache Lucene (near-realtime searching)
* Có khả năng phân tích dữ liệu (Analysis data)
* Khả năng mở rộng theo chiều ngang tuyệt vời
* Hỗ trợ tìm kiếm mờ (fuzzy), tức là từ khóa tìm kiếm có thể bị sai lỗi chính tả hay không đúng cú pháp thì vẫn có khả năng elasticsearch trả về kết quả tốt.
* Hỗ trợ Structured Query DSL (Domain-Specific Language), cung cấp việc đặc tả những câu truy vấn phức tạp một cách cụ thể và rõ ràng bằng JSON.Các bạn có thể tìm hiểu thêm tại đây
* Hỗ trợ nhiều Elasticsearc client như Java, PhP, Javascript, Ruby, .NET, Python
* Nhược điểm
* Elasticsearch được thiết kế cho mục đích search, do vậy với những nhiệm vụ khác ngoài search như CRUD thì elastic kém thế hơn so với những database khác như Mongodb, Mysql …. Do vậy người ta ít khi dùng elasticsearch làm database chính, mà thường kết hợp nó với 1 database khác.
* Trong elasticsearch không có khái niệm database transaction, tức là nó sẽ không đảm bảo được toàn vẹn dữ liệu trong các hoạt độngInsert, Update, Delete.Tức khi chúng ta thực hiện thay đổi nhiều bản ghi nếu xảy ra lỗi thì sẽ làm cho logic của mình bị sai hay dẫn tới mất mát dữ liệu. Đây cũng là 1 phần khiến elasticsearch không nên là database chính.
* Không thích hợp với những hệ thống thường xuyên cập nhật dữ liệu. Sẽ rất tốn kém cho việc đánh index dữ liệu.
  + - * 1. Cài đặt Elasticsearch
* Download và cài đặt Elasticsearch PGP Key bằng lệnh sau:

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –

* Để cài đặt Elasticsearch trên Ubuntu ta có 2 cách là cài từ Repository hoặc cài từ file .deb
* **Cách 1: Cách Elasticsearch bằng APT Repository**

Cài gói apt-transport-https

sudo apt-get install apt-transport-https

Lưu định nghĩa repository vào /etc/apt/sources.list.d/elastic-6.x.list:

echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list

Cài đặt Elasticsearch bằng lệnh:

sudo apt-get update && sudo apt-get install elasticsearch

* **Cách 2: cài đặt elasticsearch bằng gói .deb**

 Các lệnh dưới đây để tải và cài đặt elasticsearch bằng file .deb

* wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.4.0.deb
* wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.4.0.deb.sha512
* shasum -a 512 -c elasticsearch-6.4.0.deb.sha512
* sudo dpkg -i elasticsearch-6.4.0.deb
* Chạy Elasticsearch

Elasticsearch sau khi được cài đặt sẽ không tự động chạy.

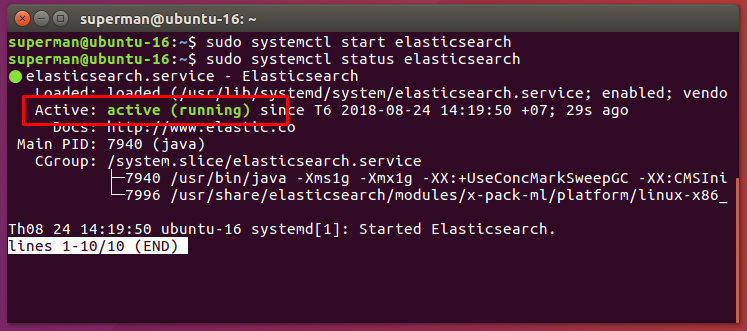
Để khởi động, start, stop elasticsearch ta làm như sau:

Kích hoạt serivce elasticsearch và tự động start khi khởi động máy:

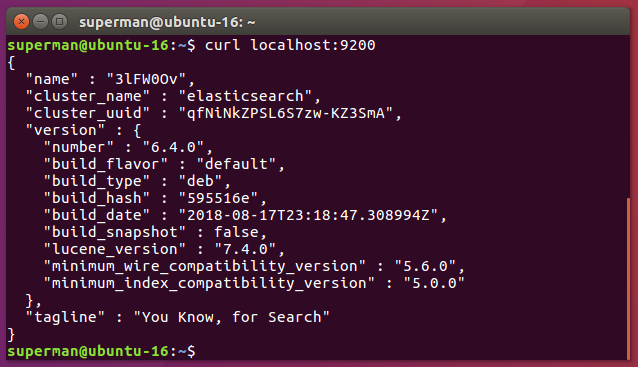
* sudo /bin/systemctl daemon-reload
* sudo /bin/systemctl enable elasticsearch.service

Để start, stop, xem thông tin service elasticsearch ta dùng các lệnh sau:

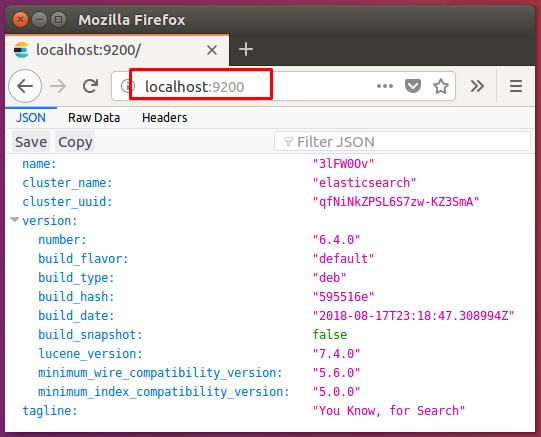
* sudo systemctl start elasticsearch
* sudo systemctl stop elasticsearch
* sudo systemctl status elasticsearch
* Demo:



Xem trạng thái của Elasticsearh



Xem thông tin của Elasticsearch



Xem thông tin của Elasticsearch

* Gỡ Elasticsearch:

sudo apt-get --purge autoremove elasticsearch

* + - * 1. Lưu trữ dữ liệu và tìm kiếm?

**Mô hình lưu trữ:**

* Elasticsearch sử dụng Lucene của Apache - một thử viên full text search viết bằng Java. Về bản chất, nó sử dụng một cấu trúc dữ liệu gọi là inverted index (chỉ mục đảo ngược) để có thể thực hiện tìm kiếm với hiệu suất cao.
* Tài liệu chính đơn vị cơ sở để quản lỹ dữ liệu trong Elasticsearch và inverted index được tạo ra bằng việc tokenize (thuật ngữ trong Elasticsearch) các khái niệm trong tài liệu. Bằng việc sử dụng kỹ thuật inverted index, một bảng chỉ mục các khái niệm và danh sách tài liệu liên quan đến khái niệm đó sẽ được tạo ra.
* Nó khá tương đồng với chỉ mục của một cuốn sách, nơi hiển thị một danh sách các khái niệm được sử dụng cùng với số trang mà nó xuất hiện. Trong Elasticsearhc, khi nói rằng một tài liệu được index, thì chúng ta hiểu rằng, inverted index của tài liệu đó đã được tạo ra.
* Ví dụ, khi chúng ta cần tìm từ khoá "Việt Nam", Elasticsearch sẽ thực hiện tìm kiếm trên inverted index (rất nhanh do các từ được sắp xếp), tìm được từ khoá "Việt Nam", sau đó trả về ID của các tài liệu tương ứng. Để có thể tìm kiếm nâng cao hơn (ví dụ, tìm với từ khoá "Viet Nam"), thì quá trình phân tích và index tài liệu là rất quan trọng.
* Ngoài ra, quá trình tìm kiếm thực tế còn phức tạp hơn nữa, khi mà Elasticsearch còn phải đánh giá mức độ tương qua giữa những từ trong index với từ khoá cần tìm.

**Tìm kiếm:**

Quá trình tìm kiếm tài liệu sẽ diễn ra theo 2 giai đoạn:

**Giai đoạn query:**

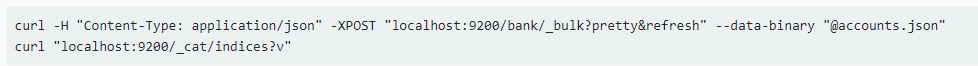
* Đầu tiên, truy vấn tìm kiếm sẽ đến node điều phối, ở đây, truy vấn này sẽ được chuyển tiếp đến mọi shard (cả primary và replica) trong index. Mỗi shard sẽ thực hiện thao tác tìm kiếm tương ứng với truy vấn một cách độc lập và trả về ID của các tài liệu có độ tương quan cao nhất (mặc định sẽ là 10 tài liệu có điểm tương quan cao nhất được trả về).
* Các ID này được trả về cho node điều phối, ở đây, nó sẽ được gộp với kết quả của các shard khác và sắp xếp lại để tìm ra những tài liệu có độ tương quan cao nhất.

**Gia đoạn fetch:**

* Sau khi node điều phối gộp và sắp xếp lại kết quả nhận được từ các shard, nó sẽ thực hiện thao tác lấy thông tin của tài liệu đó. Các shard lại thực hiện công việc của mình và trả về tài liệu (toàn bộ tài liệu chứ không chỉ ID như giai đoạn trước) cho node điều phối.
* Sau khi quá trình lấy dữ liệu đã xong, kết quả sẽ được trả về cho client.
* Lưu ý rằng, để thực hiện tìm kiếm hiệu quả, mọi node trong cluster cần biết được trạng thái của cluster đó. Trạng thái của cluster sẽ bao gồm các thông tin như mỗi node có chứa index và shard nào. Nhờ đó, mọi node đều có thể trở thành node điều phối (và Elasticsearch sẽ thay đổi node điều phối trong trường hợp node được chọn không hoạt động).
* Tìm kiếm cơ bản trong Elasticsearch:
* File dữ liệu mẫu của Elasticsearch:

[https://raw.githubusercontent.com/elastic/elasticsearch/master/docs/src/test/resources/accounts.json#](https://raw.githubusercontent.com/elastic/elasticsearch/master/docs/src/test/resources/accounts.json)

* Đẩy dữ liệu vào Elasticsearch:



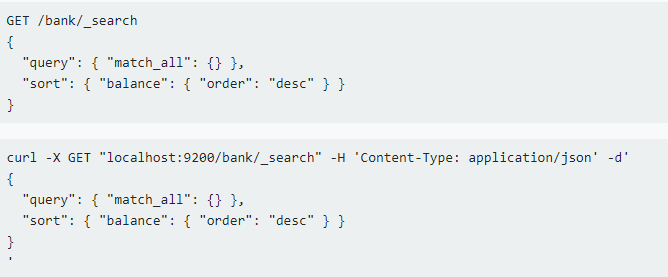
* Kiểm tra một document:



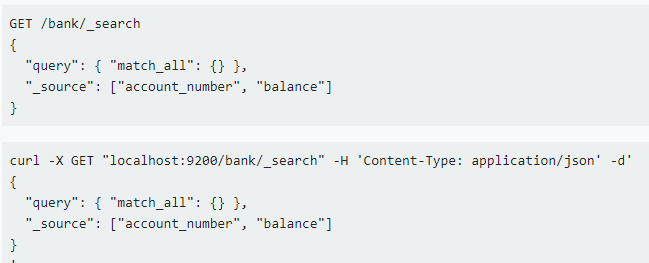
Tìm tất cả document trong index /bank/



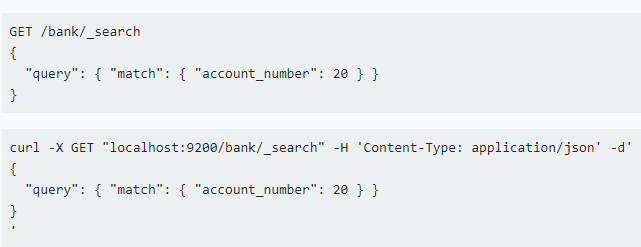
* Sắp xếp kết quả trả về:



* Tùy chọn trường dữ liệu trả về:



* Tìm kiếm dữ liệu phù hợp:



* ….

### **ASP.Net Core**

1. Giới thiệu

Khi nhắc đến .NET, chúng ta thường nghĩ đến mã nguồn đóng, Only Window, Build with Visual Studio. Nhưng sau khi cho ra mắt ASP.NET Corre 1, Microsoft đã thay đổi hoàn toàn suy nghĩ của chúng ta với các điểm sau:

* Open source Microsoft đã open source .NET và public trên github để thu hút các lập trình viên cũng như tận dùng sức mạnh của cộng đồng.
* Build for Windows, Mac, or Linux, có lẽ khoảng vài năm trước ít ai nghĩ chuyện này có thể xảy ra nhưng bây giờ, Microsoft đã rất thân thiện với linux và .NET run with linux chính là một minh chứng rõ nét.
* Visual Studio Code không còn gắn chặt với Visual studio, ASP.NET Core bây giờ có thể viết với Visual studio code (free) trên cả Window, Linux và Mac.

1. **ASP.net Core là gì?**

ASP.NET Core là một nền tảng mã nguồn mở, cross-platform framework cho việc xây dựng và kết nối các ứng dụng web hiện đại chẳng hạn như ứng dụng web, Internet of Thing, Mobile Backend - Web API

ASP.NET Core cung cấp một kiến trúc để tối ưu hóa việc xây dựng các ứng dụng đám mây (microsoft azure) hoặc các ứng dụng độc lập. Bạn có thế phát triển ứng dụng ASP.NET Core đa nên tảng (Window, Mac, Linux), với mã nguồn mở tại Github.

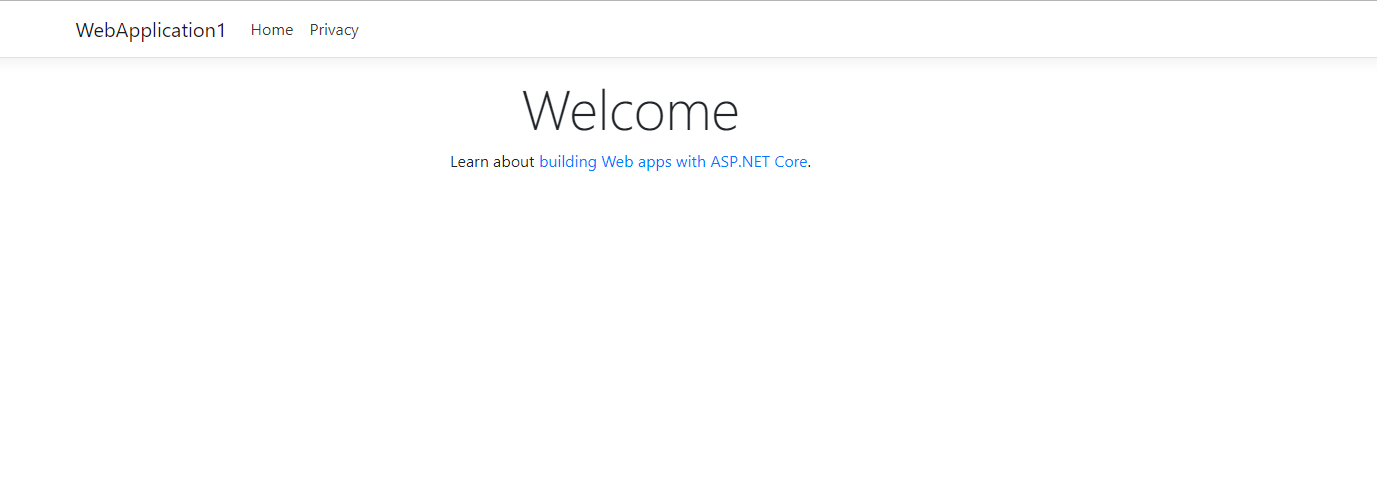
1. Build ứng dụng ASP.net core với Visual Studio

Đầu tiên chúng ta cần phải tải Visual Studio 2017 và .NET Core 2.2

Các bước tạo một project ASP.net core:

1. Tạo web app Mở Visual studion chọn new Project hoặc file -> new -> Project
2. Chọn .Net Core sau đó chọn  [ASP.NET](http://asp.net/) Core Web Application (.NET Core) và nhập tên cho project -> click OK
3. Tiếp theo bạn có thể chọn tạo project theo các cách như trong hình

Visual studio sẽ tạo ra một project ASP.net core và sử dụng them default. Ấn shift + F5 để build chương trình kết quả sẽ được hiển thị:



Hình 26 Giao diện website mặc định khi tạo project với ASP.net core

1. **ASP.net core API.**
   * + - API là gì?

* API là gì?

API là viết tắt của Application Programming Interface (giao diện lập trình ứng dụng) phương thức kết nối với các thư viện và ứng dụng khác. Windows có nhiều API, và Twitter cũng có web API, tuy nhiên chúng thực hiện các chức năng khác nhau, với mục tiêu khác nhau. Nó chính là một phần mềm giao tiếp được sử dụng bởi các ứng dụng khác nhau. Nó cũng giống như bàn phím là thiết bị dùng để giao tiếp giữa người sử dụng và máy tính, API là một phần mềm giao tiếp giữa chương trình và hệ điều hành. api Mỗi bộ API dành cho các hệ điều hành khác nhau là hoàn toàn khác nhau và không có sự tương thích với nhau. API dành cho các hệ điều hành Windows và Linux là hoàn toàn khác nhau. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Nói nôm na API là một loại công cụ để tạo ra phần mềm, cũng giống như các viên gạch xây nên một toà nhà. Chúng ta chia việc xây nhà ra các phần khác nhau, API cũng như viên gạch là một bộ phận của ngôi nhà, nhưng việc tạo ra nó là một công đoạn riêng hoàn toàn, chúng ra có thể tự làm hoặc đơn giản là đi mua của các nhà cung cấp. Web API là một trong những công nghệ mới của Microsoft dùng để xây dựng dịch vụ thành phần phân tán. Web API là mô hình dùng để hỗ trợ MVC bao gồm: routing, controller, action result, filter, filter, loc container, model binder, unit test, injection. Bên cạnh đó nó còn hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu.

* Những điểm nổi bật của API.

Đây là một trong những framework mới sẽ giúp ít cho bạn trong việc xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Mã nguồn mở nên bạn có thể được sử dụng bởi bất kì một client nào hỗ trợ XML, JSON. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma. Bạn có thể sử dụng các host nằm trong phần ứng dụng hoặc trên IIS. Một kiểu kiến trúc vô cùng phù hợp dành cho các thiết bị trang bị băng thông giới hạn như smartphone, tablet. Thường nó có định dạng dữ liệu là JSON, XML hoặc một kiểu dữ liệu bất kỳ.

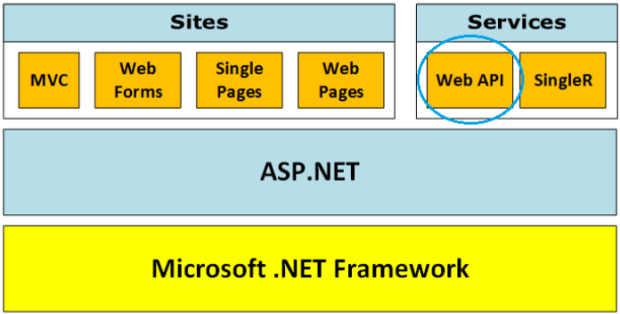
* Ưu điểm:

Cấu hình đơn giản khi được so sánh với WCF Khả năng trình diễn cao Hỗ trợ chức năng RESTful một cách đầy đủ Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test Mã nguồn mở.

* API có phải là một ngôn ngữ lập trình không?

Câu trả lời là không. Các hàm API cũng như các hàm bình thường mà chúng ta hay viết và trên từng ngôn ngữ khác nhau cũng có các bộ API tương ứng khác nhau. Ví dụ: Google API có bộ cho .NET, PHP ,.. Và mỗi bộ API hầu như đều có hướng dẫn sử dụng rất cụ thể và chi tiết. Cách tạo dự án MVC web API Đầu tiên bạn vào File/ New/ Project, sau đó chọn [ASP.NET](http://asp.net/) Web Application, rồi OK, chọn tiếp template Web API Tiếp theo để tạo cơ sở dữ liệu cho dự án bạn chọn chuột phải vào dự án Add New Item, SQL Server Database, sau đó lưu vào thư mục App\_Data. Tạo những model, controller, view sau đó vào các thư mục Models, Controllers và Views.

* + - * ASP.net core API
* ASP.net web API là gì?
* Là framework giúp chúng ta tạo ra các Web API **–**API trên nền web (HTTP).
* Web API là các dịch vụ Web (Web service) được xây dựng dựa trên HTTP sử dụng mô hình lập trình convention (như ASP.NET MVC).



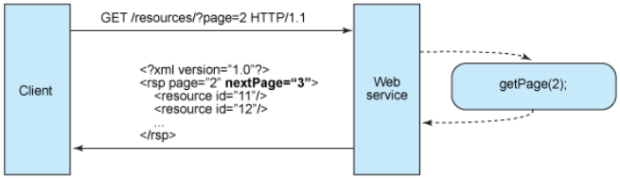
Hình 27 Mô hình lập trình Convention

* Đặc điểm Web API
* Giúp cho việc xây dựng các HTTP service rất đơn giản, nhanh chóng
* Mã nguồn mở (Open Source) và có thể được sử dụng bởi bất kì client nào hỗ trợ XML, JSON.
* Hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content formats.
* Có thể host trong ứng dụng hoặc trên IIS.
* Kiến trúc lý tưởng cho các thiết bị có băng thông giới hạn như các thiết bị di động.
* Định dạng dữ liệu có thể là JSON, XML hoặc một kiểu dữ liệu bất kỳ.
* Làm mới và hiện đại hóa các mẫu dự án mặc định
* Mẫu dự án trên điện thoại di động
* Nhiều tính năng mới để hỗ trợ các ứng dụng di động
* Tùy chỉnh sinh mã(code).
* Tăng cường hỗ trợ cho các phương pháp bất đồng bộ
* Đọc danh sách đầy đủ tính năng trong các ghi chú phát hành
* Ưu điểm của Web API
* Cấu hình đơn giản hơn nhiều so với WCF.
* Hiệu suất(performance) cao.
* Hỗ trợ RESTfull đầy đủ.
* Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test, …
* Mã nguồn mở (Open source).
* Giới thiệu cơ bản về RESTful Service
* Là một dịch vụ web đơn giản sử dụng giao thức HTTP và tính chất của REST
* Nó tuân thủ theo 4 nguyên tắc thiết kế cơ bản sau:
* Sử dụng các giao thức HTTP một cách rõ rang
* Phi trạng thái
* Hiển thị cấu trúc thư mục như một URLs
* Chuyển đổi linh hoạt JavaScript Object Notation (JSON) và XML hoặc cả 2
* Nguyên tắc cơ bản để tạo ra RESTful Service

4 nguyên tắc thiết kế cơ bản sảu:

* **Nguyên tắc 1:** Sử dụng các giao thức HTTP một cách rõ rang
* Thiết lập một ánh xạ 1-1 giữa các hành động: tạo, đọc, cập nhật và xoá (CRUD) các quá trình vận hành và các phương thức HTTP:
* POST (HttpPost) – Tạo một tài nguyên trên máy chủ
* GET (HttpGet) – Truy xuất một tài nguyên
* PUT (HttpPut) – Thay đổi trạng thái một tài nguyên hoặc để cập nhật nó
* DELETE (HttpDelete) – Huỷ bỏ hoặc xoá một tài nguyên
* **Nguyên tắc 2:** Phi trạng thái

Mô hình phi trạng thái:



Hình 28 Mô Hình phi trạng thái

Mô hình trạng thái:



Hình 29 Mô hình trạng thái

**Nguyên tắc 3:** Hiển thị cấu trúc thư mục như một URLs

Cấu trúc địa chỉ của RESTful service:

* Giấu các đuôi tài liệu mở rộng của bản gốc trong máy chủ (.jsp, .php, .asp).
* Để mọi thứ là chữ thường (thực ra là không phân biệt, nhưng cũng nên tuân thủ để khỏi phải nhớ HOA-thường lung tung).
* Thay thế các khoảng trống bằng gạch chân hoặc gạch nối (một trong hai loại).
* Tránh các chuỗi yêu cầu.
* Thay vì sử dụng mã (404 Not Found) khi yêu cầu địa chỉ cho một phần đường dẫn thì luôn luôn cung cấp một trang mặc định hoặc tài nguyên như một phản hồi.
* **Nguyên tắc 4:** Chuyển đổi JavaScript Object Notation (JSON) và XML hoặc cả hai.
* Là một bản tóm tắt các thuộc tính của những thứ trong mô hình dữ liệu hệ thống.
* Định dạng dữ liệu mà ứng dụng và trao đổi dịch vụ trong mức đáp ứng yêu cầu/ phản hồi hoặc trong phần thân của HTTP.
* Các chủ thể trong mô hình dữ liệu có liên quan với nhau.
* Cấu trúc dịch vụ sao cho nó tận dụng được phần đầu chấp nhận HTTP có sẵn bên trong – một loại MIME
* Các bước xây dựng web Service: Tương tự như phần **1.2.1 c**

### JQuery

**jQuery** là một thư viện JavaScript nhỏ gọn, chạy nhanh. jQuery giải quyết các vấn đề tương tác với HTML, bắt sự kiện, hiệu ứng động ... trở lên rất đơn giản.

Các tính năng chính jQuery cung cấp gồm:

* Tương tác với HTML/DOM
* Tương tác với CSS
* Bắt và xử lý sự kiện HTML
* Các hiệu ứng và chuyển động trong HTML
* AJAX (Asynchronous JavaScript and XML)
* JSON parsing
* Các tiện ích xây dựng sẵn

### **Ajax**

**Ajax** không phải là một công nghệ đơn lẻ mà là tập hợp của một vài công nghệ, chúng được kết hợp lại với nhau và tạo nên một sự đột phá. Cụ thể, danh sách ấy gồm 5 thành phần:

* XHTML và CSS
* DOM
* XML và XSLT
* XMLHttpRequest
* Javascript

### **Bootstrap**

Bootstrap là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn.

Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tao ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế reponsive của bạn dễ dàng hơn và nhanh chóng hơn.

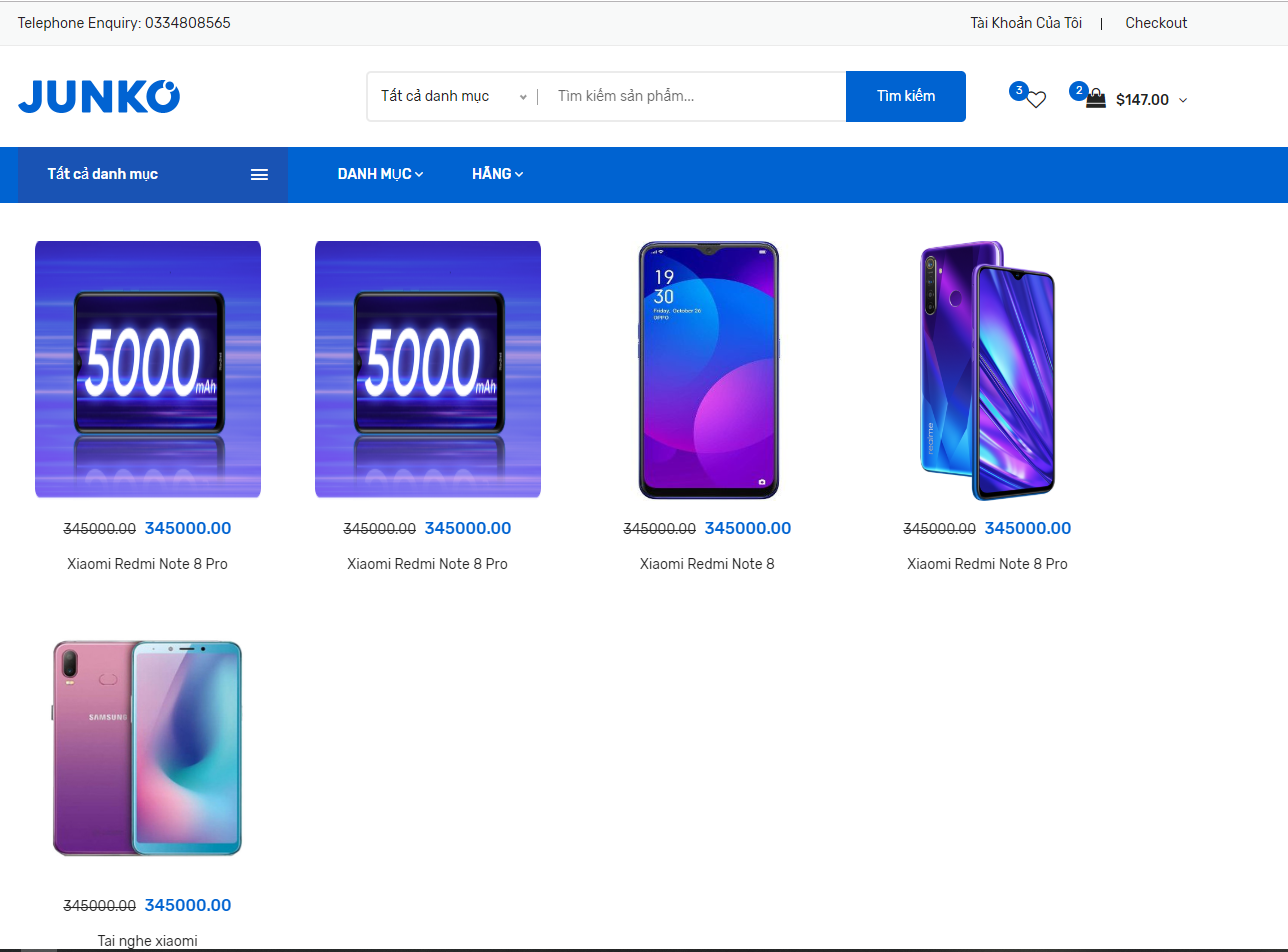
Những điểm thuận lợi khi bản sử dụng bootstrap:

Rất dễ để sử dụng: Nó đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cẩn có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.

Tính năng Responsive: Bootstrap’s xây dựng sẵn reponsive css trên các thiết bị phones, tablets, và desktops.

## Một số kết quả đạt được

### Giao diện trang chủ

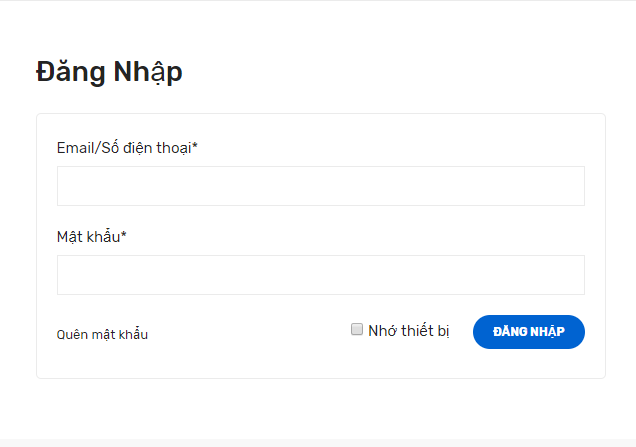


Hình 32 Giao diện trang chủ

Thiết kế giao diện trang chủ được hiển thị với 1 thanh Navbar để hiển thị các đường dẫn cho một số chức năng.

Cuối cùng sẽ có 1 thanh Footer để hiển thị một số thông tin của trang web và thông tin liên lạc

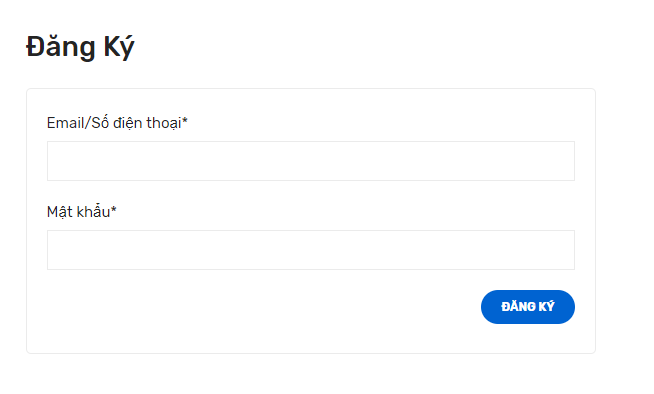
### Giao diện đăng nhập



Hình 33 Giao diện chức năng “**Đăng nhập**”

Giao diện cho chức năng Đăng nhập sẽ gồm có 1 bức ảnh Background và Form để nhập thông tin đăng nhập như Email và Mật khẩu.

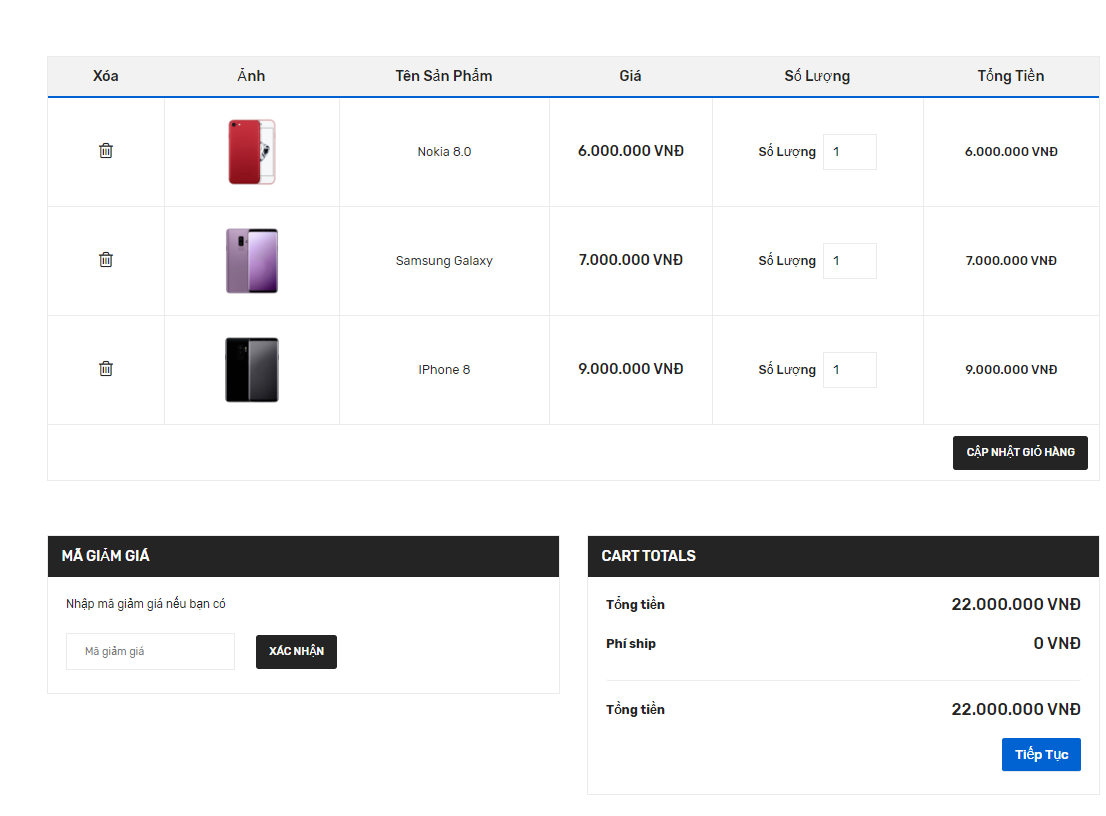
### Giao diện đăng ký



Hình 34 Giao diện chức năng “**Đăng ký**”

Giao diện cho chức năng Đăng ký sẽ gồm có 1 bức ảnh Background và Form để nhập thông tin đăng ký như Họ tên, Email, Ngày sinh, Giới tính, Mật khẩu...

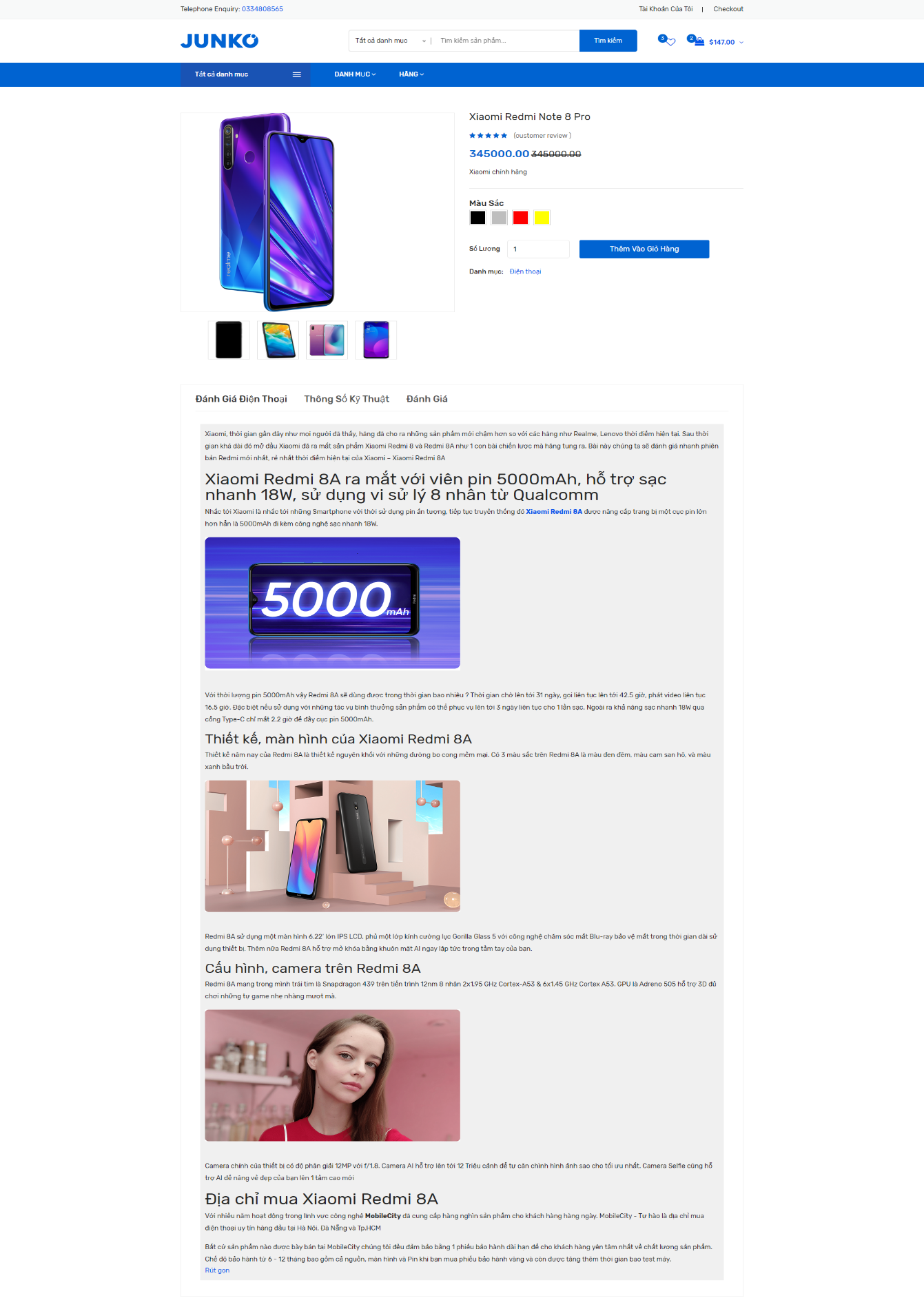
### Giao diện giỏ hàng



Hình 35 Giao diện giỏ hàng

Giao diện giỏ hàng gồm có danh dách những sản phẩm mà người dùng đã thêm vào. Trong bảng này, người dùng có thể thực hiện một số thao tác như xóa sản phẩm khỏi giỏ, cập nhật số lượng sản phẩm, link tới trang chi tiết sản phẩm. Bên cạnh đó, người dùng có thể nhìn thấy 1 ô để nhập phiếu mua hàng để được giảm giá. Và đặc biệt là phần hiển thị Tổng giỏ hàng, sẽ thống kê Tổng số lượng sản phẩm và Tổng tiền trên giỏ hàng.

### Giao diện chi tiết sản phẩm



Hình 36 Giao diện chi tiết sản phẩm

Giao diện sản phẩm sẽ hiển thông tin chi tiết sản phẩm của hệ thống.

### Giao diện quản lý đơn hàng người dùng

Hình 37 Giao diện quản lý đơn hàng người dùng

Đây là giao diện quản lý đơn hàng của người dùng. Người dùng có thể thấy được mã, trạng thái, ngày đặt của các đơn hàng mà mình đã đặt.

### Giao diện chi tiết đơn hàng người dùng

Hình 38 Giao diện chi tiết đơn hàng người dùng

Đây là giao diện chi tiết của từng đơn hàng. Người dùng có thể thấy được mã, hình ảnh, tên và trạng thái của sản phẩm trong mỗi một đơn hàng.

### Giao diện trang quản lý

Hình 39 Giao diện trang quản lý

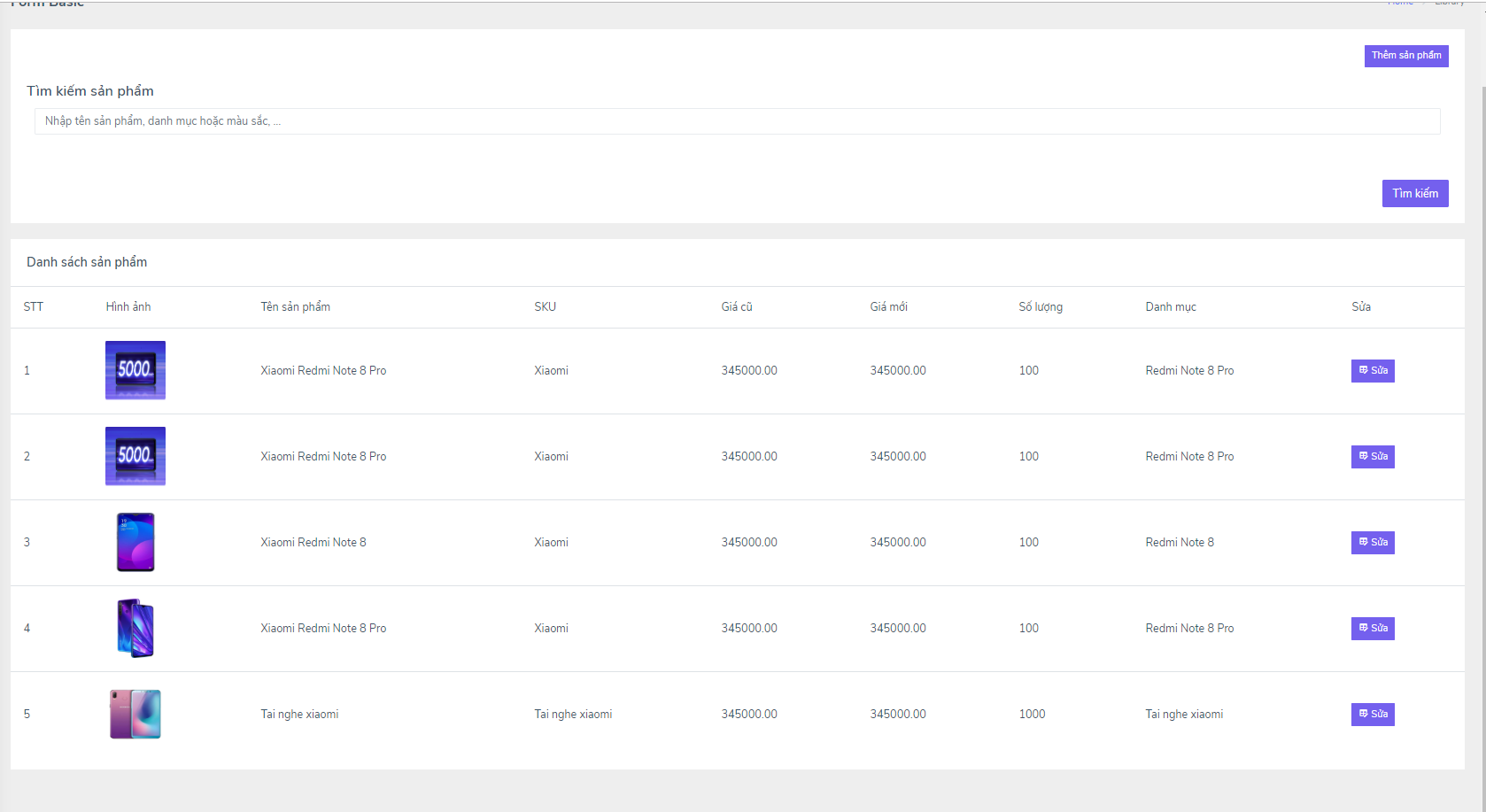
Giao diện trang quản lý mặc định là phần Thống kê lượng truy cập, trong trang này người dùng sẽ thấy 1 biểu đồ thể hiện số lượng người truy cập trang web.

### Giao diện quản lý người dùng

Hình 40 Giao diện quản lý người dùng

Giao diện quản lý người dùng sẽ hiển thị danh sách các tài khoản người dùng gồm các thông tin như, tên, email, số điện thoại,... khi đã đăng ký thành công vào hệ thống

### Giao diện quản lý sản phẩm



Hình 41 Giao diện quản lý sản phẩm

Giao diện trang quản lý sản phẩm hiển thị danh sách các sản phẩm của hệ thống, hiển thị đầy đủ thông tin của sản phẩm như, tên, giá, …

### Giao diện quản lý đơn hàng

Hình 42 Giao diện quản lý đơn hàng

Giao diện quản lý đơn hàng sẽ hiển thị các đơn hàng bao gồm các thông tin như họ tên, tên sản phẩm, số lượng, giá,... mà người dùng đã thanh toán thành công.

# KẾT LUẬN

Xây dựng được **Website bán điện thoại**. Website giới thiệu thông tin về các dòng điện thoại smartphone và cho phép khách hàng có thể đặt hàng ngay trên website. Website cho phép khách hàng có thể quản lý được các đơn hàng mà mình đã đặt. Ngoài ra website cung cấp cho admin một trang giao diện dễ sử dụng và dễ quản lý được các mục.

Website chạy được trên tất cả trên các trình duyệt web như IE, Google Chrome, Mozilla Firefox.

Qua việc tự tìm hiểu và xây dựng website, cùng với sự hướng dẫn chỉ bảo của thầy giáo hướng dẫn. Em đã biết thêm được nhiều kiến thức mới hơn như công nghệ MVC, ASP.Net Core kết hợp ElasticSearch, cách triển khai một dự án website. Hiểu được quy trình nghiệp vụ khi làm một website bán hàng, đưa thông tin trên mạng.

Hệ thống website điện thoại chưa cho phép khách hàng thanh toán trực tuyến được, vẫn còn một số thủ tục thanh toán thủ công.

Một số chức năng vẫn còn chưa hoàn thiện và chưa đúng với mong muốn của đã đặt ra. Cách thiết kế giao diện, màu sắc vẫn chưa được chuyên nghiệp. Một số chỗ vẫn còn chưa đúng và dư thừa …

Vì công nghệ MVC còn mới nên một số chức năng của hệ thống vẫn còn lỗi và một số khác vẫn chưa được hiện thực

Trong thời gian nghiên cứu và thực hiện đề tài, em đã vạch ra được hướng phát triển tiếp theo của đề tài như sau:

* + - * + Xây dựng thêm các Control làm phong phú trên việc tùy biến giao diện của website.
        + Bổ sung các Module hỗ trợ cho các website giới thiệu sản phẩm và thương mại điện tử.

# TÀI LIỆU THAM KHẢO

[1]. **Xây dựng Web Service dùng API RESTful Service**

<https://duythanhcse.wordpress.com/2015/11/10/bai-68-xay-dung-web-service-dung-api-restful-servicephan-1/>

[2]. Elasticsearch là gì?

<https://viblo.asia/p/elasticsearch-la-gi-1Je5E8RmlnL>

<https://www.elastic.co/what-is/elasticsearch>

[3]. Elasticsearch: Data Organization

<https://manhhomienbienthuy.github.io/2019/May/20/elasticsearch-data-organization.html>

[4]. Truy cấn tìm kiếm cơ bản trong Elasticsearch

<https://xuanthulab.net/truy-van-tim-kiem-co-ban-trong-elasticsearch.html>

[5] **ASP.NET Web API**

<https://duythanhcse.wordpress.com/2015/11/10/bai-68-xay-dung-web-service-dung-api-restful-servicephan-1/>