**ĐẠI HỌC THÁI NGUYÊN**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**HOÀNG THANH THIỆN**

**XÂY DỰNG WEBSITE BÁN ĐỒ THỂ THAO**

**CHO CỬA HÀNG ACE SPORT TẠI TP TUYÊN QUANG**

# ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**THÁI NGUYÊN, NĂM 2020**

**ĐẠI HỌC THÁI NGUYÊN**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

****

## ĐỒ ÁN

## TỐT NGHIỆP ĐẠI HỌC

**NGÀNH CÔNG NGHỆ THÔNG TIN**

***Đề tài:***

**XÂY DỰNG WEBSITE BÁN ĐỒ THỂ THAO**

**CHO CỬA HÀNG ACE SPORT TẠI TP TUYÊN QUANG**

**Sinh viên thực hiện : Hoàng Thanh Thiện**

**Lớp  : CNTT K14E**

**Giáo viên hướng dẫn : ThS. Vũ Việt Dũng**

**THÁI NGUYÊN, NĂM 2020**

**THÁI NGUYÊN, NĂM 2015**

# LỜI CẢM ƠN

Em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin trường Đại học Công Nghệ Thông Tin Và Truyền Thông đã tạo điều kiện tốt cho em hoàn thành đồ án này.

Em xin gửi lời cảm ơn chân thành tới thầy **Vũ Việt Dũng** đã nhiệt tình hướng dẫn cho em hoàn thành đồ án tốt nghiệp này.

Em xin gửi lời cảm ơn sâu sắc tới tất cả các thầy cô, những người đã giảng dạy, trang bị cho em những kiến thức quý báu trong suốt những năm học vừa qua tại trường Đại học Công nghệ thông tin và Truyền thông– Đại học Thái Nguyên.

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

Mặc dù em đã cố gắng hoàn thiện thật tốt đồ án nhưng do kiến thức có hạn do đó không thể tránh khỏi những sai sót, em rất mong nhận được sự cảm thông, ý kiến đóng góp của các quý Thầy Cô và các bạn!

Em xin chân thành cảm ơn!

# 

# LỜI CAM ĐOAN

Em xin cam đoan về nội dung đồ án tốt nghiệp với tên đề tài **“**X**ây dựng website bán đồ thể thao cho cửa hàng Ace Sport tại Tp Tuyên quang”** là không sao chép nội dung cơ bản từ các đồ án khác, hay các sản phẩm tương tự không phải do em làm ra. Sản phẩm của đồ án là chính bản thân em nghiên cứu và xây dựng.

Nếu có điều gì không trung thực em xin chịu mọi hình thức kỉ luật của trường Đại học công nghệ thông tin và truyền thông Thái Nguyên.

Sinh viên thực hiện đề tài

**Hoàng Thanh Thiện**

# ­­­­­­­­

# MỤC LỤC

# DANH MỤC HÌNH ẢNH

# LỜI NÓI ĐẦU

Ngày nay, công nghệ web trên nền .NET Framework của Microsoft ngày càng phát triển mạnh mẽ, cùng với các công nghệ khác như: PHP, JSP,....Trong đó công nghệ ASP.NET MVC là một công nghệ mang nhiều tính năng ưu việt. Bên cạnh đó, việc bán hàng qua mạng thông qua những website trực tuyến đã không còn quá xa lạ đối với người Việt Nam ta và cho đến nay ngày càng phát huy thế mạnh của nó. Những gian hàng online giúp tiết kiệm được rất nhiều thời gian cho những người bận rộn. Vì vậy em đã chọn đề tài “ X**ây dựng website bán đồ thể thao cho cửa hàng Ace Sport tại Tp Tuyên quang** ” dựa trên nền tảng framework ASP.NET MVC và một số tính năng khác để hỗ trợ khách hàng trong việc đặt hàng qua mạng Internet. Các vấn đề mà em đã nghiên cứu được về framework ASP.NET MVC, công nghệ AJAX, cấu trúc JSON, Bootstrap. Các hướng tiếp cận phát triển của em là tìm kiếm thông tin qua Internet, xử lý thông tin dưới sự giúp đỡ của Giáo viên hướng dẫn. Trong quá trình nghiên cứu em đã gặp nhiều khó khăn và có một số phương pháp giải quyết vấn đề như: lên các forum tham khảo và hỏi ý kiến mọi người, xin ý kiến đóng góp của giáo viên hướng dẫn, trao đổi ý kiến với bạn cùng lớp. Kết quả cuối cùng, em đã xây dựng thành công website bán đồ thể thao đáp ứng đầy đủ các nhu cầu của cửa hàng. Website sử dụng giao diện Bootstrap, hỗ trợ người dùng sử dụng nhiều thiết bị với các kích thước màn hình khác nhau như : desktop, laptop, tablet, smartphone.

­­

# CHƯƠNG 1: TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT

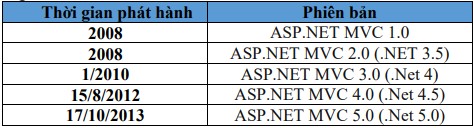
# 1.1 Công nghệ ASP.NET MVC

## 1.1.1 Lịch sử ra đời của MVC

Tất cả bắt đầu vào những năm 70 của thế kỷ 20, tại phòng thí nghiệm Xerox PARC ở Palo Alto. Sự ra đời của giao diện đồ họa (GUI) và lập trình hướng đối tượng (Object Oriented Programming) cho phép lập trình viên làm việc với những thành phần đồ họa như những đối tượng đồ họa có thuộc tính và phương thức riêng của nó. Không dừng lại ở đó, những nhà nghiên cứu ở Xerox PARC còn đi xa hơn khi cho ra đời cái gọi là kiến trúc MVC (viết tắt của Model – View – Controller).

MVC được phát minh tại Xerox Parc vào những năm 70, bởi TrygveReenskaug. MVC lần đầu tiên xuất hiện công khai là trong Smalltalk-80. Các giấy tờ quan trọng đầu tiên được công bố trên MVC là “A Cookbook for Using the ModelView-Controller User Interface Paradigm in Smalltalk – 80”, bởi Glenn Krasner và Stephen Pope, xuất bản trong tháng 8 / tháng 9 năm 1988. - Thế hệ tiếp theo của MVC xuất hiện cùng với hệ điều hành NeXT và các phần mềm của nó.

Kiến trúc này ngày càng được phát triển và hoàn thiện nhằm giải quyết các vấn đề phát sinh cũng như các giải pháp cho quá trình phát triển phần mềm. Vì vậy sau đó, lần lượt các MVC framework ra đời dựa trên mô hình MVC như: CodeIgniter, Zend, ASP.NET MVC …



## 1.1.2 Sự khác nhau giữa MVC và webform

|  |  |  |
| --- | --- | --- |
|  | **MVC** | **Webform** |
| Thành phần | ASP.net MVC chia ra làm 3 phần:  – Models  – View  – Controller.  Mọi tương tác của người dùng với Views sẽ được xử lý thông qua việc thực hiện các action hành động trong Controllers, không còn postback, lifecycle và events. | ASP.net WebForm sử dụng ViewState để quản lý Các trang ASP.net đều có lifecycle, postback và dùng các web controls, các event để thực hiện các hành động cho UI (User Interface) . |
| Việc kiểm tra (test), gỡ lỗi (debug) | Đối với MVC thì việc đó có thể sử dụng các unit test có thể thẩm định rất dễ dàng các Controllers thực hiện như thế nào. | Với ASP.net WebForm đều phải chạy tất cả các tiến trình của ASP.net, và sự thay đổi ID của bất kỳ Controls nào cũng |

## 1.1.3 Kiến trúc sử dụng ASP.NET MVC

ASP.NET MVC được thiết kế một cách gọn nhẹ, giảm thiểu sự phức tạp của việc xây dựng ứng dụng website bằng cách chia một ứng dụng thành 3 tầng (layer): Model, View và Controller. Sự chia nhỏ này giúp lập trình viên dễ dàng kiểm soát các thành phần trong khi phát triển, cũng như lợi ích lâu dài trong việc kiểm tra, bảo trì và nâng cấp.

1. **Model**

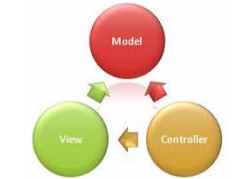
Model chứa và thể hiện các đặc tính và logic ứng dụng. Theo một cách hiểu khác, Model đại diện cho dữ liệu và logic cốt lõi. Nó chính là những lớp (class) chứa thông tin về các đối tượng mà ta cần phải thao tác, làm việc trên nó. Ví dụ: "Sách" chứa các thông tin như "Tên", "Ngày phát hành", "Giá", "Nhà sản xuất", ...

1. **View**

View làm nhiệm vụ thể hiện một Model hay nhiều Model một cách trực quan, nó nhận thông tin (một Model hoặc nhiều Model) sau đó biểu diễn lên trang website.

1. **Controller**

Controller nằm giữa tầng View và Model, làm nhiệm vụ tìm kiếm, xử lý một hoặc nhiều Model, sau đó gửi Model tới View để View hiển thị



Hình 1. Mô hình Asp.net MVC

## 1.1.4 Đặc điểm ASP.NET MVC Framework

Tiếp tục hỗ trợ các tính năng trong ASP.NET :

- Hỗ trợ sử dụng các các tập tin:.ASPX, .ASCX, .Master như là thành phần View

- Hỗ trợ đầy đủ các tính năng bảo mật của ASP.net: Form/ Windows authenticate, URL authorization, membership/roles, output và data caching, section/ profile state, configuration system, provider architecture.

Tách rõ ràng các mối liên quan, mở ra khả năng test TDD (Test Driven Developer).

- Có thể test unit trong ứng dụng mà không cần phải chạy Controllers cùng với tiến trình của ASP.NET và có thể dùng bất kỳ một unit testing framework nào như NUnit, MBUnit, MS Test,…

Có khả năng mở rộng, mọi thứ trong MVC được thiết kế để dễ thay thế, dễ dàng tùy biến. Ánh xạ URL mạnh mẽ, cho phép xây dựng ứng dụng với những URL sạch.

Không sử dụng mô hình post-back từ giao diện gửi đến server. Thay vào đó, chủ động đưa những post-back từ View đến thẳng lớp Controller.

Hỗ trợ nhiều công cụ tạo View (Support for Multiple View Engines):

- Cho phép chọn công cụ tạo view. Hộp thoại New Project cho phép xác định view engine mặc định cho một project.

- Các loại view engine o Web Forms (ASPX) o Razor o Hay một view engine nguồn mở như Spark, NHaml, NDjango.

Hỗ trợ định tuyến :

- ASP.NET MVC Framework có một bộ máy ánh xạ URL thật sự mạnh mẽ. - Bộ máy này cung cấp phương pháp rất linh hoạt trong việc ánh xạ URLs sang các Controller Classes.

- Bạn có thể dễ dàng định ra các quy luật, cài đặt đường đi, ASP.NET dựa vào các quy luật đường đi đó để xác định Controller và action cần phải thực thi.

- ASP.NET còn có khả năng phân tích URL, chuyển các thông số trong URL thành các tham số trong lời gọi hàm của Controller.

**Model Binding** :

- Model Binding là tính năng thế mạnh của ASP.NET MVC (và bây giờ nó cũng được áp dụng cho cả Web Forms trong phiên bản ASP.NET 4.5).

- Hỗ trợ bạn viết phương thức nhận một đối tượng tùy biến như là một tham số 5

- Với sự hỗ trợ của Model Binding, bây giờ bạn chỉ cần tập trung vào việc cài đặt các nghiệp vụ logic, không cần phải bận tâm về việc suy nghĩ làm cách nào để ánh xạ dữ liệu từ người dùng sang các đối ượng .NET.

**Filters :**

- Là tính năng mạnh trong ASP.NET MVC. Hỗ trợ cho việc kiểm tra tính hợp lệ trước khi một action method được gọi hoặc sau khi một action method thi hành.

**Razor View:** Từ ASP.net MVC 3 đi kèm với một công cụ View mới có tên là Razor với những lợi ích sau:

- Cú pháp Razor là sạch sẽ và xúc tích, đòi hỏi một số lượng tối thiểu các tổ hợp phím.

- Việc tìm hiểu Razor tương đối dễ dàng vì nó dựa trên ngôn ngữ C# và Visual Basic.

- Visual Studio bao gồm IntelliSense và mã cú pháp Razor được màu hóa.

- Với Razor views có thể kiểm tra từng đơn vị mà không đòi hỏi bạn phải chạy các ứng dụng hoặc phải chạy website.

## 1.1.5 Các phiên bản ASP MVC

1. **MVC 3**

- So với phiên bản ASP.NET MVC2 thì MVC3 được hỗ trợ thêm HTML5 và CSS3. - Cải thiện về Model Validation

– tính năng kiểm tra tính hợp lệ của dữ liệu do người dùng nhập vào.

- Hỗ trợ 2 View Engine: RAZOR, ASPX, kể cả Open source trong khi ASP.NET MVC2 chỉ có ASPX.

- Controller được cải tiến hơn như thuộc tính ViewBag và kiểu ActionResult.

- Cải thiện Dependency Injection với IDpendencyResolver (có 2 phần: DependencyResolver và interface IDpendencyResolver) trong ASP.NET MVC3; đây là lớp thực thi mô hình Service Locator, cho phép framework gọi DIContainer khi cần làm việc với 1 lớp thực thi từ 1 kiểu cụ thể.

- Cách tiếp cận với JavaScript được hạn chế. - Hỗ trợ caching trong Partial page.

1. **MVC 4**

- ASP.NET Web API ra đời, nhằm đơn giản hoá việc lập trình với HTML hiện đại và đây là một cải tiến mới thay cho WCF Web API.

- Mặc định của dự án được cải thiện hơn về hình thức bố trí, giúp dễ nhìn hơn.

- Mẫu Empty Project là project trống, phù hợp cho những developer muốn nâng cao khả năng lập trình với ASP.NET MVC4. - Giới thiệu jQuery Mobile và mẫu Mobile Project cho dự án.

- Hỗ trợ Asynchrnous Controller.

- Kiểm soát Bundling và Minification thông qua web.config.

- Hỗ trợ cho việc đăng nhập OAuth và OpenID bằng cách sử dụng thư viện DotNetOpenAuth. Cho phép Logins từ Facebook và những tài khoản khác.

- Phiên bản mới Windows Azure SDK 1.6 được phát hành.

1. **MVC 5**

Với MVC5 thì cải tiến hơn so với ASP.NET MVC4, Bootstrap được thay thế mẫu MVC mặc định.

- Chứng thực người dùng Authentication Filter được tuỳ chỉnh hoặc chứng thực từ hãng thứ 3 cung cấp.

- Với Filter overrides, chúng ta có thể Filter override trên Method hoặc Controller.

- Thuộc tính Routing được tích hợp vào MVC5.

# 1.2 Bootstrap

* Bootstrap là một framework CSS được Twitter phát triển. Nó là một tập hợp các bộ chọn, thuộc tính và giá trị có sẵn để giúp web designer tránh việc lặp đi lặp lại trong quá trình tạo ra các class CSS và những đoạn mã HTML giống nhau trong dự án web của mình. Ngoài CSS ra, thì bootstrap còn hỗ trợ các function tiện ích được viết dựa trên JQuery(Carousel, Tooltip, Popovers ,...).
* Những lý do để sử dụng Bootstrap:

- Được viết bởi những người có óc thẩm mỹ và tài năng trên khắp thế giới. Sự tương thích của trình duyệt với thiết bị đã được kiểm tra nhiều lần nên có thể tin 7 tưởng kết quả mình làm ra và nhiều khi không cần kiểm tra lại. Vì vậy, giúp cho dự án của bạn tiết kiệm được thời gian và tiền bạc.

- Chỉ cần biết sơ qua HTML, CSS, Javascript, Jquery là bạn có thể sử dụng Bootstrap để tạo nên một trang web sang trọng và đầy đủ. Nhưng lại không cần code quá nhiều CSS.

- Với giao diện mặc định là màu xám bạc sang trọng, hỗ trợ các component thông dụng mà các website hiện nay cần có. Vì nó là opensource nên bạn có thể vào mã nguồn của nó để thay đổi theo ý thích của bản thân.

- Do có sử dụng Grid System nên Bootstrap mặc định hỗ trợ Responsive. Bootstrap được viết theo xu hướng Mobile First tức là ưu tiên giao diện trên Mobile trước. Nên việc sử dụng Bootstrap cho website của bạn sẽ phù hợp với tất cả kých thước màn hình. Nhờ đó mà chúng ta không cần xây dựng thêm một trang web riêng biệt cho mobile.

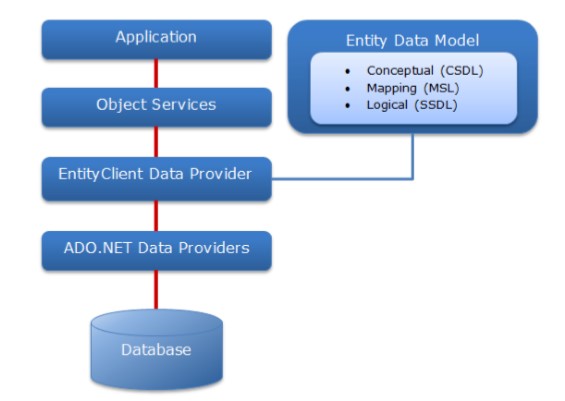
- Đội ngũ phát triển Bootstrap đã bổ sung thêm tính năng Customizer. Giúp cho designer có thể lựa chọn những thuộc tính, component phù hợp với project của họ. Chức năng này giúp ta không cần phải tải toàn bộ mã nguồn về máy



# 1.3 Giới thiệu về Entity Framework

## 1.3.1 Tổng quan

Entity Framework (EF) là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. EF cho phép các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Lợi ích lớn nhất của EF là giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu. EF được Microsoft hỗ trợ phát triển lâu dài và bền vững, vì vậy EF là 1 framework mạnh nhất hiện nay để phát triển ứng dụng Web với sự hỗ trợ đông đảo của các nhà phát triển Web. Kiến trúc của Entity Framework được minh họa như sau:



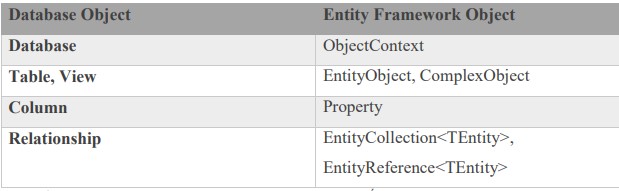
Hình 2. Mô hình kiến trúc Entity Framework

## 1.3.2 Tầng Application

* Application (ứng dụng) là tầng chứa giao diện trang Web (HTML, CSS, Javascript, hình ảnh, …) và các đoạn mã nguồn (C#, VB) để tương tác dữ liệu với các tầng khác trong mô hình thông qua Object Services.

## 1.3.3 Tầng Object Services

* Object Services (tạm dịch là các dịch vụ đối tượng) là tầng chứa quá trình tương tác giữa ứng dụng và database, hay nói cách khác nó là nơi chủ yếu để truy cập dữ liệu từ database và trả ngược kết quả về giao diện. Object Services cung cấp các tiện ích để truy vết các thay đổi và quản lý nhận dạng, đồng thời là các quan hệ và thay đổi ở database.
* Đây là các class tự động sinh ra tương ứng với mô hình dữ liệu. Các class này bao gồm:
* ObjectContext đại diện cho một database. ObjectContext có chức năng quản lý các kết nối, định nghĩa mô hình dữ liệu với metadata và thao tác với database. Lớp này cũng có thể thêm vào các phương thức đại diện cho các stored procedure trong database.
* ObjectSet là một một tập hợp các entity. Mỗi đối tượng này tương ứng với một table. Có thể lấy được các đối tượng này thông qua các property tương ứng của ObjectContext.
* EntityObject, ComplexObject là các lớp tương ứng cho một dòng dữ liệu của table trong database. Khác biệt chính giữa hai loại này là ComplexObject không chứa primary key.
* EntityCollection và EntityReference: là các đối tượng thể hiện mối quan hệ (relationship) giữa hai entity class. Mỗi đối tượng này có thể được truy xuất thông qua các property của entity class.
* Sự ánh xạ tương đương được thể hiện bảng sau:



Hình 3. Bảng ánh xạ tương đương giữa các đối tượng trong database và EF

## 1.3.4 Tầng EntityClient Data Provider

EntityClient là một data provider mới của ADO.NET dùng để truy xuất đến database. Được xây dựng bên trên các ADO.NET data provider cơ bản, EntityClient không truy xuất trực tiếp dữ liệu mà thông qua các data provider khác dựa vào các thông tin dữ liệu từ Entity Data Model.

EntityClient cũng bao gồm các lớp giống như các ADO.NET data provider khác và tên lớp được đặt với tiền tố Entity. Ví dụ bạn có thể tạo kết nối bằng EntityConnection, tạo các câu truy vấn bằng EntityCommand và đọc kết quả bằng EntityDataReader.

Một điểm khác biệt với các data provider khác là EntityClient sử dụng Entity SQL để truy vấn dữ liệu. Các lệnh Entity SQL sẽ được chuyển thành một cấu trúc lệnh dạng cây (command tree) và chuyển xuống cho các data provider khác.

## 1.3.5 Tầng ADO.NET Data Providers

Đây là tầng thấp nhấp để dịch các truy vấn L2E (LINQ to Entity) thông qua cây lệnh thành các câu lệnh SQL và thực thi các câu lệnh trong hệ thống DBMS (database management system – hệ quản lý dữ liệu) nào đó. Tầng này kết với database sử dụng ADO.NET.

EntityClient là một data provider mới của ADO.NET dùng để truy xuất đến database. Được xây dựng bên trên các ADO.NET data provider cơ bản, EntityClient không truy xuất trực tiếp dữ liệu mà thông qua các data provider khác dựa vào các thông tin dữ liệu từ Entity Data Model.

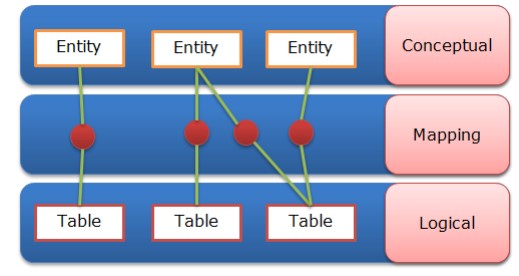
EntityClient cũng bao gồm các lớp giống như các ADO.NET data provider khác và tên lớp được đặt với tiền tố Entity. Ví dụ bạn có thể tạo kết nối bằng EntityConnection, tạo các câu truy vấn bằng EntityCommand và đọc kết quả bằng EntityDataReader.

Một điểm khác biệt với các data provider khác là EntityClient sử dụng Entity SQL để truy vấn dữ liệu. Các lệnh Entity SQL sẽ được chuyển thành một cấu trúc lệnh dạng cây (command tree) và chuyển xuống cho các data provider khác.

## 1.3.6 Tầng EDM (Entity Data Model)

Entity Data Model (EDM) là mô hình dữ liệu được mô tả thông qua các ngôn ngữ theo chuẩn XML. EDM được chia làm 3 lớp là: Conceptual, Mapping và Logical. Mỗi lớp này được định nghĩa bởi ngôn ngữ riêng theo định dạng XML:

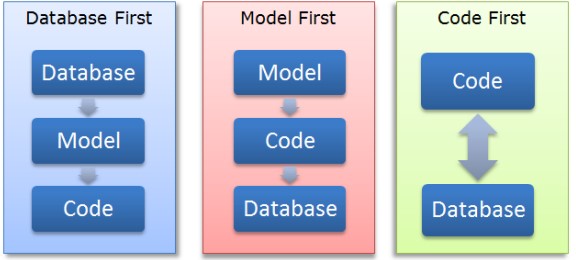
* Conceptual – Conceptual Schema Definition Language (CSDL): là ngôn ngữ định nghĩa các entity, relationship, hàm trong tập tin với phần mở rộng .csdl. Có thể tạo được các entity class (object layer).
* Mapping – Mapping specification language (MSL): định nghĩa các ánh xạ giữa lớp conceptual và logical, nội dung này được lưu trong tập tin .msl.
* Logical – Store Schema Definition Language (SSDL): định nghĩa mô hình lưu trữ của dữ liệu, lưu trữ trong tập tin .ssdl.



Hình 4. Lưu đồ Entity Mapping Model

## 1.3.7 Cách sử dụng trong Entity Framework

* Là thành phần trung gian giữa ứng dụng và database, Entity Data Model (EDM) giúp tạo liên kết và mô tả giữa nguồn dữ liệu vật lý và các đối tượng bussiness (hay object layer) trong ứng dụng. Dựa vào sự lựa chọn thành phần nào sẽ được tạo ra trước, một ứng dụng Entity Framework (EF) có thể sử dụng 1 trong 3 cách tiếp cận: Database First, Model First và Code First.
* Các sơ đồ dưới đây minh hoạt mối quan hệ và thứ tự tạo ra giữa các thành phần dữ liệu trong ứng dụng của Entity Framework. Phần Model tương ứng với tập tin .edmx, Code tương ứng với tập tin .cs (hoặc .vb nếu bạn dùng VB.NET).



Hình 5. Lưu đồ Entity Mapping Model

# 1.4 Tổng quan về Sql Server

Là một hệ quản trị cơ sở dữ liệu do Microsoft phát triển. SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ mạng máy tính hoạt động theo mô hình khách chủ cho phép đồng thời cùng lúc có nhiều người dùng truy xuất đến dữ liệu, quản lý việc truy nhập hợp lệ và các quyền hạn của từng người dùng trên mạng.

# 1.5 JavaScript, JQuery và Ajax

## 1.5.1 JavaScript

- JavaScript (có phần mở rộng .js) là một ngôn ngữ lập trình dạng kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ được sử dụng rộng rãi trên các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng đối tượng có sẵn trong các ứng dụng. Giống Java, JavaScript có cú pháp tương tự C, nhưng gần giống với Self hơn Java. Hiện nay, có rất nhiều framework, libraries được viết bằng ngôn ngữ JavaScript: o jQuery: Một thư viện mạnh mẽ và thông dụng. o AngularJS: Một thư viện xây dựng ứng dụng Single Page. o NodeJS: Một thư viện được phát triển phía Server dùng để xây dựng ứng dụng realtime. o Firebase: Một nền tảng di động dùng lưu trữ dữ liệu theo thời gian thực.

## 1.5.2 Jquery

- jQuery là một thư viện kiểu mới của JavaScript, được tạo bởi John Resig vào năm 2006 với một phương châm tuyệt vời: Write less, do more. - jQuery làm đơn giản hóa việc truyền tải HTML, xử lý sự kiện, tạo hiệu ứng động và tương tác Ajax. Với jQuery, khái niệm Rapid Web Development đã không còn quá xa lạ. - jQuery là một bộ công cụ tiện ích JavaScript làm đơn giản hóa các tác vụ đa dạng với việc viết ít code hơn. Dưới đây liệt kê một số tính năng tối quan trọng được hỗ trợ bởi jQuery: o Thao tác DOM − jQuery giúp dễ dàng lựa chọn các phần tử DOM để traverse một cách dễ dàng như sử dụng CSS, và chỉnh sửa nội dung của chúng bởi sử dụng phương tiện Selector mã nguồn mở, mà được gọi là Sizzle. o Xử lý sự kiện − jQuery giúp tương tác với người dùng tốt hơn bằng việc xử lý các sự kiện đa dạng mà không làm cho HTML code rối tung lên với các Event Handler. o Hỗ trợ AJAX − jQuery giúp bạn rất nhiều để phát triển một site giàu tính năng và phản hồi tốt bởi sử dụng công nghệ AJAX. o Hiệu ứng − jQuery đi kèm với rất nhiều các hiệu ứng đa dạng và đẹp mắt mà bạn có thể sử dụng trong các Website của mình. o Gọn nhẹ − jQuery là thư viện gọn nhẹ - nó chỉ có kích cỡ khoảng 19KB (gzipped). o Được hỗ trợ hầu hết bởi các trình duyệt hiện đại − jQuery được hỗ trợ hầu hết bởi các trình duyệt hiện đại, và làm việc tốt trên IE 6.0+, FF 2.0+, Safari 3.0+, Chrome và Opera 9.0+ o Cập nhật và hỗ trợ các công nghệ mới nhất − jQuery hỗ trợ CSS3 Selector và cú pháp XPath cơ bản.

## 1.5.3 Ajax

AJAX, là viết tắt của Asynchronous JavaScript and XML, (tạm dịch là JavaScript và XML không đồng bộ), là một kỹ thuật mới để tạo các ứng dụng web giàu tính tương tác, nhanh hơn và mượt mà hơn với sự giúp đỡ của XML, HTML, CSS và JavaScript. AJAX không phải là ngôn ngữ lập trình mới. Tên AJAX ở đây không có nghĩa là bạn phải học XML trước khi cần học AJAX. Điều này là không cần thiết. AJAX cho phép các trang web được cập nhật một cách không đồng bộ bằng cách trao đổi các lượng dữ liệu nhỏ với Server. Tức là AJAX giúp cập nhật các phần nhỏ trong trang mà không cần tải lại toàn bộ trang. Nếu đến đây bạn chưa hiểu thì các dòng tiếp theo đây sẽ giúp bạn hiểu cách AJAX làm việc hơn.

# 1.6 Phương thức mã hóa dữ liệu MD5

Viết tắt của Message-Digest algorithm 5 (Giải thuật Tiêu hóa tin 5) là một hàm băm mật mã học được diễn tả bằng một số hệ thập lục phân 32 ký tự, theo chuẩn RFC 1321, các chương trình MD5 thường được gọi là MD5Sum. Nó được dùng để tạo ra một chuỗi 128 bit duy nhất từ một chuỗi dữ liệu nhập bất kỳ, và vì thế thường được dùng để kiểm tra tính toàn vẹn dữ liệu của một tập tin.

# CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG WEBSITE

# 2.1 Khảo sát hệ thống

## 2.1.1 Môi trường

Trong thời đại công nghiệp hóa hiện đại hóa đất nước, đặc biệt là sự phát triển mạnh mẽ của ngành công nghệ thông tin, và nhu cầu sử dụng website để quảng bá sản phẩm, website để xây dựng hệ thống thông tin của ngành giáo dục, viết blog... càng cao.

Và thực tế cho thấy, hướng phát triển xây dựng website cho các công ty lớn, các doanh nghiệp lớn để quảng cáo, quản lý ngày càng nhiều và do đó họ yêu cầu riêng một số lượng lớn các kỹ sư thiết kế xây dựng cho họ là sự cần thiết.

Công ty lớn, doanh nghiệp lớn điều có các bộ phận riêng là nhà thiết kế xây dựng bảo trì website cho chính họ. Vậy còn các công ty vừa và nhỏ? Doanh nghiệp nhỏ? Thậm chí là một xưởng sản xuất nhỏ, hay là 1 blog của một cá nhân không đủ tiền để chi trả riêng cho một bộ phận thiết kế xây dựng bảo trì cho website quản bá của mình thì sao? Không có bộ mặt, không có sự quảng bá mạnh mẽ trong truyền thông thì các công ty đó, doanh nghiệp đó khi nào mới đủ tầm để vươn xa, khi nào mới phát triển mạnh hơn nữa.

Vậy hướng phát triển của hệ thống này đi đến mục đích của những khó khăn đó. Giúp các công ty vừa và nhỏ, cá nhân hay một nhóm tổ chức có riêng cho mình một hệ thống website quảng cáo và quản lý sản phẩm của mình.

Vậy tại sao phải có hệ thống Website qua những mục ở trên và những gì đã khảo sát thực tế, tự nhận thấy được rằng nhu cầu của các doanh nghiệp vừa và nhỏ các xưởng ... đều có nhu cầu xây dựng Website của họ rất lớn và tất nhiên khoản phí để họ chi trả đến những nhà thiết kế Website cũng không quá đắt đỏ. Nhưng một bài toán được đưa ra như sau: Bạn là nhà thiết kế, và hàng ngày bạn đi ra ngoài đường tìm kiếm và đưa ra những vẫn đề để bạn sẽ là nhà thiết kế Website quảng bá cho họ, và tất nhiên bạn sẽ mất thời gian để quảng cáo những gì bạn làm được cho khách hàng của bạn. Và kết quả bạn được họ yêu cầu và chấp nhận sự giúp đỡ, nhưng có những nơi không nhận đến sự giúp đỡ của bạn vậy chắc chắn bạn sẽ mất rất nhiều thời gian và không chủ động của công việc của mình. Vậy để chủ động hơn và giảm chi phí cho việc tìm kiếm khách hàng, hệ thống Website sẽ làm viêc đó thay bạn.

## 2.1.2 Hiện trạng hệ thống

Ở hệ thống cũ, để có một công việc bạn cần tìm kiếm những khách hàng của mình một cách bị động và xảy ra nhiều bất cập trong việc tiếp nhận khách hàng và làm phát sinh những chi phí không đáng có trong quá trình làm việc cụ thể ví dụ như:

+ Mất thời gian vào việc tìm kiếm khách hàng.

+ Mất chi phí cho việc đi lại.

+ Mọi công việc đều được thực hiện thủ công như hợp đồng, giấy tờ, cam kết, giao dịch.

+ Thời gian sắp xếp dữ liệu khó khăn, sơ xuất giấy tờ có thể xảy ra làm mất khách hàng đã nhận.

+ Không hiệu quả khi khách hàng bạn tìm đến lại chưa có nhu cầu.

## 2.1.3 Nhược điểm của phương thức hoạt động cũ

+ Không kiểm soát được lượng khách hàng đến với cửa hàng, không nắm được thông tin khách hàng.

+ Chi phí quảng cáo và Marketing tốn kém.

+ Chi phí mặt bằng cao.

+ Cập nhât thông tin giá cả chậm.

+ Bán hàng thụ động.

+ Các góp ý không được lưu lại.

+ Không thuận tiện cho việc thống kê, báo cáo.

## 2.1.4 Bài toán đặt ra

Theo những khó khăn và bất cập trong hệ thống cũ, bài toán cần xây dựng một hệ thống mới để công việc của nhà thiết kế phải chủ động trong việc tìm kiếm khách hàng, và mọi công việc của nhà thiết kế chỉ là chấp nhận mọi yêu cầu đến từ khách hàng. Mọi dữ liệu phải là có sẵn theo trình tự để khách hàng yêu cầu

Ở hệ thống mới này, mọi dữ liệu của khách hàng sẽ được đặt chung vào một hệ cơ sở dữ liệu, ví dụ như:

* Hồ sơ thông tin về khách hàng.
* Yêu cầu mẫu, tính thiết kế của khách hàng.
* Giá thành mà khách hàng lựa chọn.
* Cách thức giao dịch của khách hàng.

Và công việc của người thiết kế là: Chấp nhận hồ sơ, yêu cầu đúng thời gian đến từ khách hàng.

## 2.1.5 Yêu cầu hệ thống mới

-Để khắc phục hiện trạng xảy ra ở hệ thống cũ cần phải xây dựng một hệ thống mới toàn diện, hiệu quả và có tính thực tế cao. Đồng thời xây dựng một hệ thống có tính đồng bộ về dữ liệu. Dữ liệu dễ dàng được quản lý.

Hệ thống hoạt động tốt, dữ liệu từ khách hàng phải nhanh được xử lý dễ sử dụng cho khách hàng.

## 2.1.6 Mục tiêu xây dựng hệ thống mới

Áp dụng các ứng dụng trên công nghệ thông tin vào quá trình hoạt động của cửa hàng, từ đó xây dựng một hệ thống quản lý đảm bảo yêu cầu sau:

* Tự động trong quản lý thông tin đầu vào của các mặt hàng như: thêm, sửa, xoá, cập nhật.
* Tự động trong việc tìm kiếm hàng hoá theo tiêu chí đề trước.
* Hỗ trợ lập báo cáo, thống kê mặt hàng, khách hàng, nhà cung cấp,..
* Quản lý thông tin của nhà cung cấp, khách hàng thân thiện, nhân viên…
* Hệ thống phải thân thiện, dễ dàng truy xuất, vận hành sử dụng, tính thẩm mỹ cao.

Đạt và phù hợp mục đích của người dùng, phù hợp với trình độ và khả năng của người sử dụng (chỉ cần đào tạo cơ bản nếu cần ), sử dụng ngôn ngữ trên hệ điều hành windown .. và môi trường lưu trữ Sql Server sử dụng công nghệ .NET

Phải có tính phân cấp vai trò của các nhân viên để người dùng khác có thể dễ dàng nắm được công việc của mình trên của toàn hệ thống. Đồng thời hệ thống phải ổn định, đảm bảo an toàn thông tin, có khả năng cung cấp thông tin đáp ứng nhu cầu của người dùng khi họ cần. Dễ dàng kiểm tra, cải tiến, nâng cấp khi có những lỗi hệ thống bất kỳ xẩy ra. Khi lượng thông tin nhập vào lớn như cần cập nhập thường xuyên, cần có cơ chế lưu trữ hợp lý.

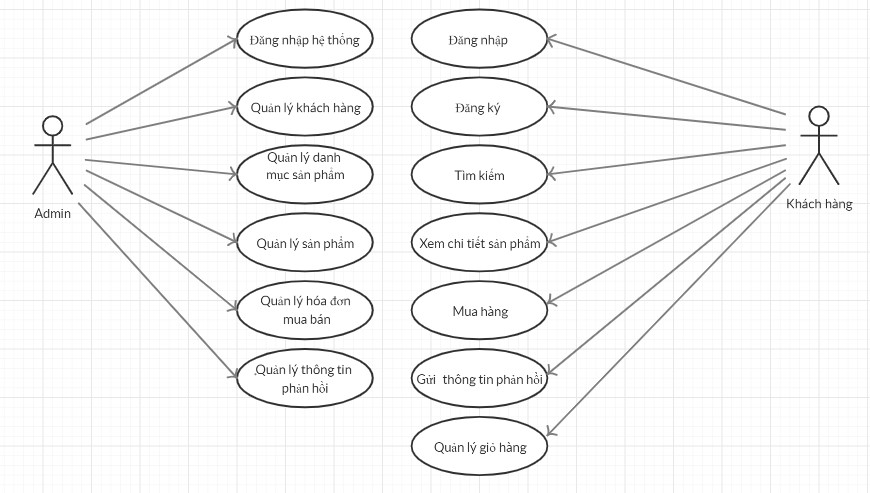
# 2.2 Phân tích hệ thống

## ­­­­2.2.1 Biểu đồ Use Case

* 1. Nhận diện tác nhân và Use Case

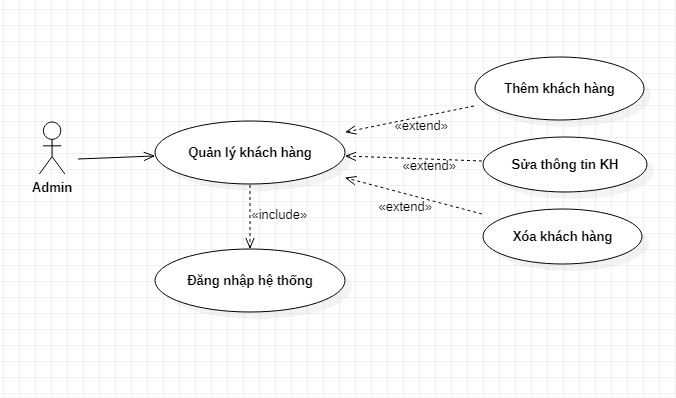
|  |  |
| --- | --- |
| **Khách hàng**  ***(****Người thực hiện truy vấn thông tin, giao dịch, mua hàng****)*** | * Đăng ký thành viên * Đăng nhập thành viên * Tìm kiếm * Xem chi tiết sản phẩm * Mua hàng * Quản lý giỏ hàng * Gửi – nhận thông tin phản hồi |
| **Người quản trị**  ***(****Quản lý mọi hoạt động và chức năng của*  *hệ thống****)*** | * Đăng nhập hệ thống * Quản lý thành viên * Quản lý danh mục sản phẩm * Quản lý sản phẩm * Quản lý hóa đơn mua hàng * Quản lý thông tin phản hồi * Xác nhận thanh toán |

* 1. Biểu đồ Use Case tổng quát của hệ thống



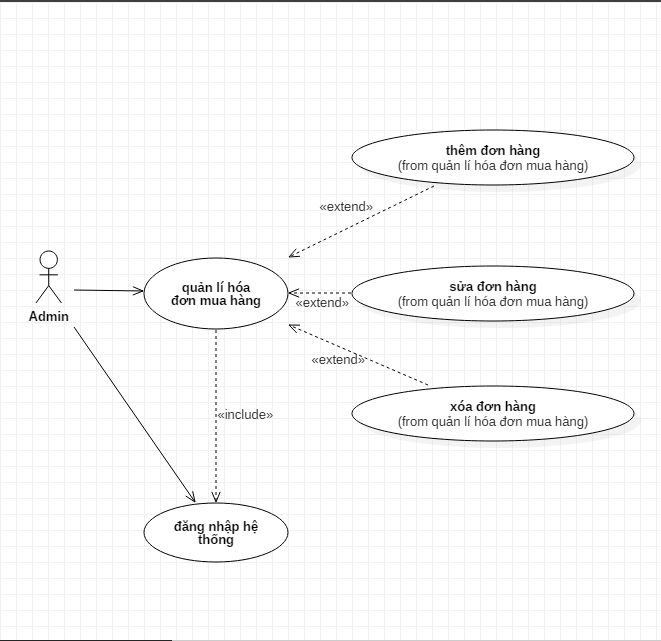
Hình 6. Biểu đồ user case của hệ thống

* 1. Biểu đồ phân rã use case quản lý khách hàng



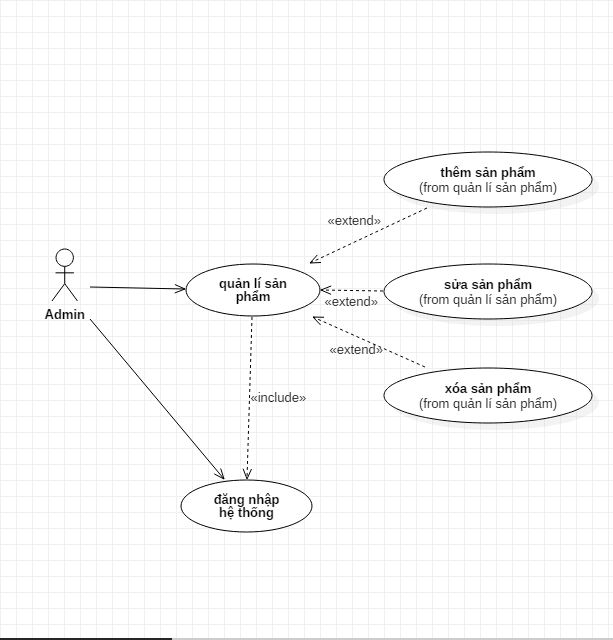
Hình 7. Biểu đồ phân rã use case quản lý khách hàng

* 1. Biểu đồ phân rã use case quản lý đơn hàng cho Admin



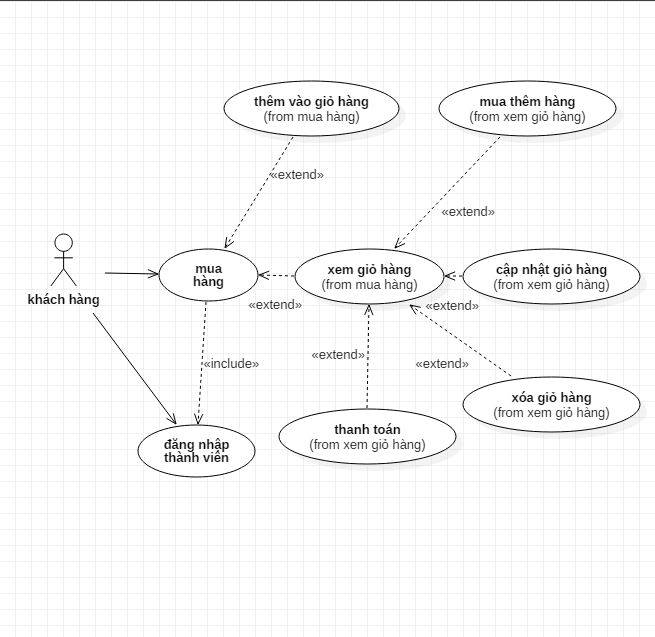
Hình 8. Biểu đồ phân rã use case quản lý đơn hàng cho Admin

* 1. Biểu đồ phân rã use case quản lý sản phẩm cho Admin



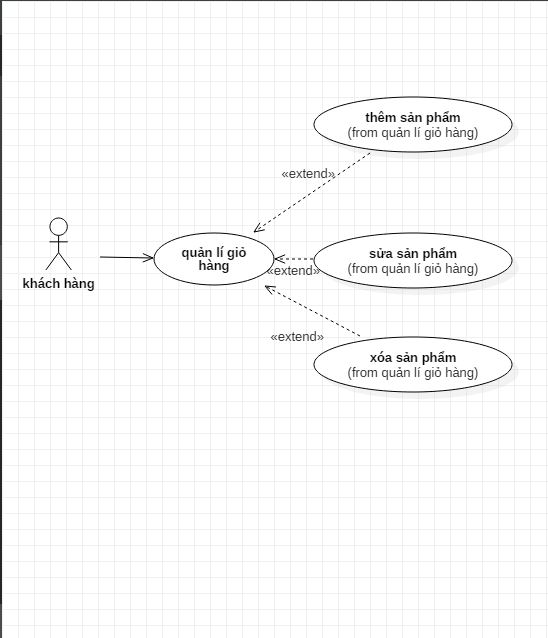
Hình 9. Biểu đồ phân rã use case quản lý sản phẩm cho Admin

* 1. Biểu đồ phân rã use case mua hàng cho khách hàng



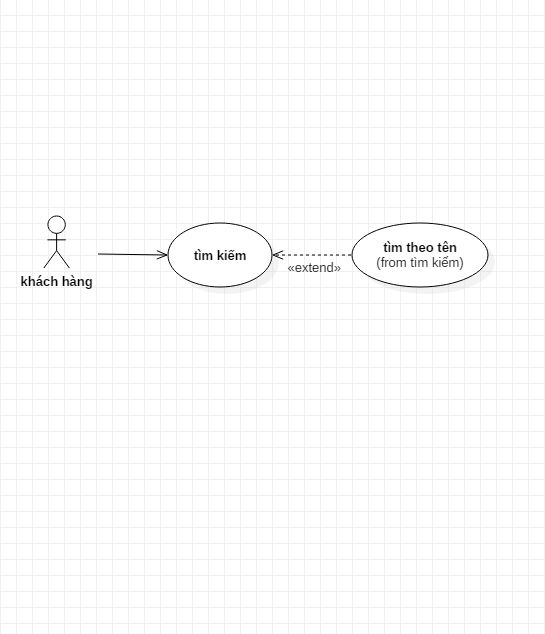
Hình 10. Biểu đồ phân rã use case mua hàng cho khách hàng

* 1. Biểu đồ phân rã use case quản lý giỏ hàng cho khách hàng



Hình 11. Biểu đồ phân rã use case quản lý giỏ hàng cho khách hàng

* 1. Biểu đồ phân rã use case tìm kiếm cho khách hàng



Hình 12. Biểu đồ phân rã use case tìm kiếm cho khách hàng

## 2.2.2 Đặc tả Use case

1. Use case tìm kiếm

* *Tên Use case*: Tìm kiếm
* *Mô tả:* Cho phép khách hàng và người quản lý tìm kiếm sản phẩm
* *Tiền điều kiện:* Vào hệ thống website
* *Luồng sự kiện chính*:
* Khách hàng chọn chức năng tìm kiếm sản phẩm.
* Khách hàng có thể tìm kiếm sản phẩm theo tên.
* Nhấn nút tìm kiếm
* Hệ thống trả về kết quả tìm kiếm nếu có trong CSDL
* Liệt kê danh sách kết quả trả về
* Khách hàng xem thông tin sản phẩm hệ thống trả về theo yêu cầu
* UC kết thúc

1. Use case xem chi tiết sản phẩm
   * *Tên Use case* : xem chi tiết sản phẩm
   * *Mô tả*: Cho phép khách hàng xem chi tiết thông tin của sản phẩm có trong hệ thống
   * *Tiền điều kiện:*Vào hệ thống website
   * *Luồng sự kiện chính:*

* Khách hàng chọn sản phẩm trên hệ thống cần xem
* Kích chọn xem chi tiết sản phẩm
* Hệ thống chuyển trang về chi tiết sản phẩm
* Hiện thị thông tin đầy đủ về sản phẩm (tên, giá ,…)
* UC kết thúc

1. Use case mua hàng
   * *Tên Use case* : Mua hàng
   * *Mô tả*: Cho phép khách hàng mua hàng
   * *Tiền điều kiện*: Đăng nhập thành viên thành công
   * *Luồng sự kiện chính:*

* Khách hàng chọn sản phẩm cần mua
* Kích thêm vào giỏ hàng
* Hệ thống thêm sản phẩm khách hàng mua vào giỏ hàng
* Hệ thống hiện thị thông tin giỏ hàng
* Khách hàng nhấn nút thanh toán
* Khách hàng nhập thông tin cá nhân, sau đó nhấn nút gửi thông tin
* Hệ thống lưu thông tin vào hóa đơn mua hàng trong cơ sở dữ liệu
* Khách hàng chọn thanh toán qua chuyển khoản ngân hang
* Khách hàng chọn trả tiền mặt khi nhận hàng
* Khách hàng thực hiện thanh toán qua ngân hàng
* Cửa hàng xác nhận thanh toán của khách hàng:

+ Nếu khách hàng chưa thực hiện thanh toán qua ngân hàng thì sẽ xóa hóa đơn mua hàng đó trong cơ sở dữ liệu

* UC kết thúc.
* *Luồng sự kiện phụ*: Nếu khách hàng mua thêm hàng thì hệ thống chuyển về trang chủ, khách hàng chọn cập nhật giỏ hàng thì số lượng các sản phẩm trong giỏ hàng sẽ được cập nhật lại, chọn xóa giỏ hàng thì các sản phẩm trong giỏ hàng sẽ bị xóa hết. Nếu khách hàng nhập thông tin cá nhân không hợp lệ thì hệ thống yêu cầu nhập lại.

1. Use case gửi thông tin phản hồi
   * *Tên Use case* : Gửi thông tin phản hồi
   * *Mô tả*: Cho phép khách hàng gửi thông tin đánh giá sản phẩm, cách phục vụ của cửa hàng…đến cửa hàng
   * *Tiền điều kiện:* Vào hệ thống website
   * *Luồng sự kiện chính:*

* Khách hàng chọn bất kì sản phẩm vào xem chi tiết
* Khách hàng đăng nhập tài khoản cá nhân
* Khách hàng chọn mức độ đánh giá dựa vào ngôi sao ,điền đánh giá, nhận xét rồi ấn nút gửi.
* UC kết thúc

1. Use case đăng nhập hệ thống

* *Tên Use case* : đăng nhập hệ thống
* *Mô tả*: UC đăng nhập thực hiện đăng nhập cho các tài khoản đã được đăng ký vào hệ thống. Người quản lý đều thực hiện đăng nhập bằng UC này.
* *Tiền điều kiện*: UC đăng ký đã được thực hiện
* *Luồng sự kiện chính*:
* Vào trang đăng nhập .
* Hệ thống hiển thị trang đăng nhập.
* Admin nhập user name và password đã đăng kí ở hệ thống.
* Hệ thống sẽ kiểm tra với cơ sở dữ liệu.
* Admin đăng nhập thành công.
* UC kết thúc
* *Luồng sự kiện phụ*: Nếu admin nhập user name và password không đúng thì hệ thống thông báo sai user name hoặc password và yêu cầu admin đăng nhập lại.

1. Use case quản lý thành viên

* *Tên UC* : Quản lý thành viên
* *Mô tả* : Use case này mô tả cách thức người quản trị quản lý thành viên: Thêm, xóa, sửa thành viên.
* *Luồng sự kiện chính*:
* Từ trang Admin quản trị viên nhấn nút thành viên.
* Hệ thống hiển thị ra danh sách các thành viên. Quản trị viên chọn các chức năng:

+ Thêm mới: Quản trị viên nhập đầy đủ các thông tin về thành viên ( tên, mật khẩu, email, số điện thoại,…) cần thêm mới.

+ Chức năng xóa: Quản trị viên chọn thành viên cần xóa và ấn nút xóa.

+ Chức năng sửa: Quản trị viên nhập đầy đủ các thông tin cần sửa và ấn nút sửa đổi.

* UC kết thúc

1. Use case quản lý sản phẩm

* *Tên UC* : Quản lý sản phẩm
* *Mô tả* : Use case này mô tả cách thức người quản trị quản lý sản phẩm như: Thêm, xóa, sửa, thiết lập tình trạng cho sản phẩm.
* *Luồng sự kiện chính*:
* Từ trang Admin quản trị viên chọn sản phẩm, sau đó kích chọn tất cả sản phẩm để hệ thống hiển thị danh sách tất cả các sản phẩm.
* Hệ thống hiển thị ra các sản phẩm. Quản trị viên chọn các chức năng:
* Thêm mới: Quản trị viên nhập đầy đủ các thông tin về sản phẩm cần thêm mới.
* Chức năng xóa: Quản trị viên chọn sản phẩm cần xóa và ấn nút xóa.
* Chức năng sửa: Quản trị viên nhập đầy đủ các thông tin cần sửa và ấn nút sửa đổi.
* UC kết thúc
* *Tiền điều kiện*: Quản trị viên phải đăng nhập thành công vào hệ thống

1. Use case quản lý hóa đơn mua hàng

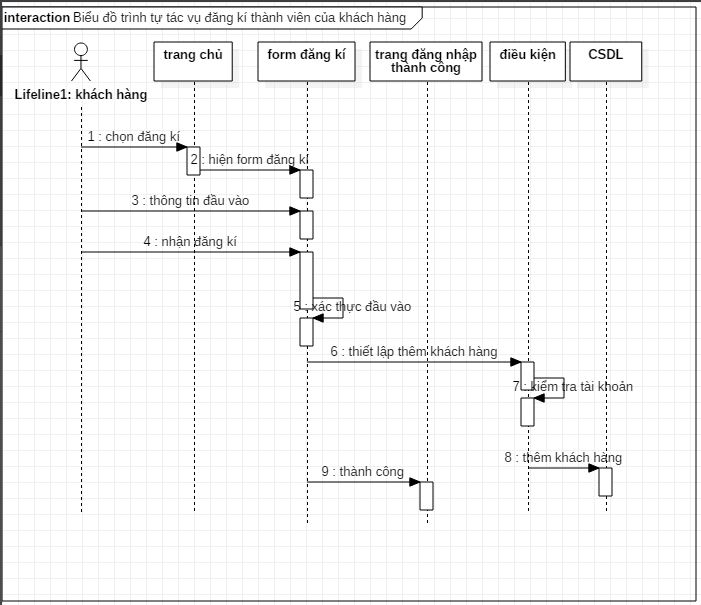
* *Tên UC* : Quản lý hóa đơn mua hàng
* *Mô tả* : Use case này mô tả cách thức người quản trị quản lý hóa đơn mua hàng như: theo dõi và cập nhật hóa đơn mua hàng.
* *Luồng sự kiện chính*:
* Từ trang admin, quản trị viên nhấn vào nút woocommerce và chọn đơn hàng
* Hệ thống hiển thị trang quản lý hóa đơn mua hàng và liệt kê danh sách các hóa đơn mua hàng của khách hàng.
* Người quản trị có thể xem thông tin hóa đơn mua hàng mà họ muốn hay xóa khỏi cơ sở dữ liệu.
* UC kết thúc
* *Tiền điều kiện*: Quản trị viên phải đăng nhập thành công vào hệ thống

1. Use case quản lý thông tin phản hồi

* *Tên UC* : Quản lý thông tin phản hồi
* *Mô tả* : UC này mô tả cách thức người quản trị quản lý các thông tin phản hồi của khách hàng
* *Luồng sự kiện chính*:
  + - * Từ trang admin, quản trị viên có thể thấy thông tin phản hồi ở bảng tin.
      * Hệ thống hiển thị trang quản lý thông tin phản hồi, liệt kê danh sách các thông tin phản hồi của khách hàng, quản trị viên sẽ trả lời phản hồi thông qua địa chỉ email của khách hàng.
      * UC kết thúc
* *Tiền điều kiện:* Quản trị viên phải đăng nhập thành công vào hệ thống.

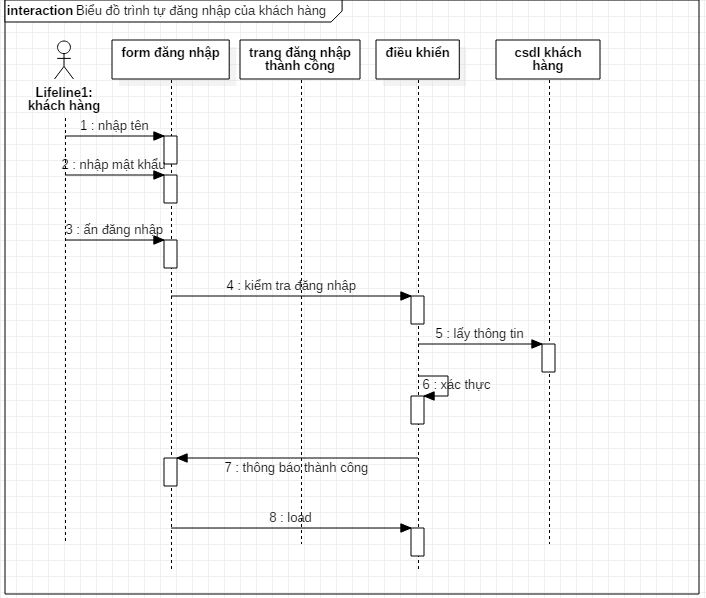
## 2.2.3. Biểu đồ trình tự

* **Phần giao dịch với khách hàng**
  + Khách hàng đăng ký làm



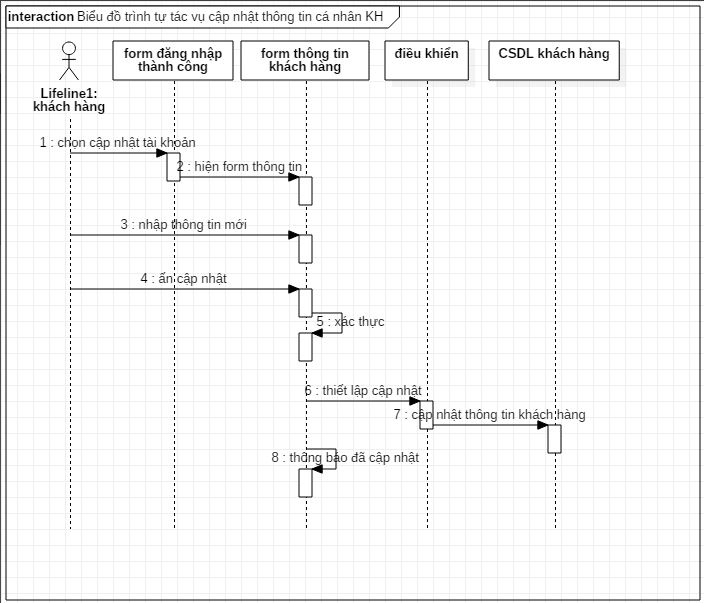
Hình 13. Biểu đồ Trình tự cho tác vụ đăng ký

* + Khách hàng đăng nhập hệ thống



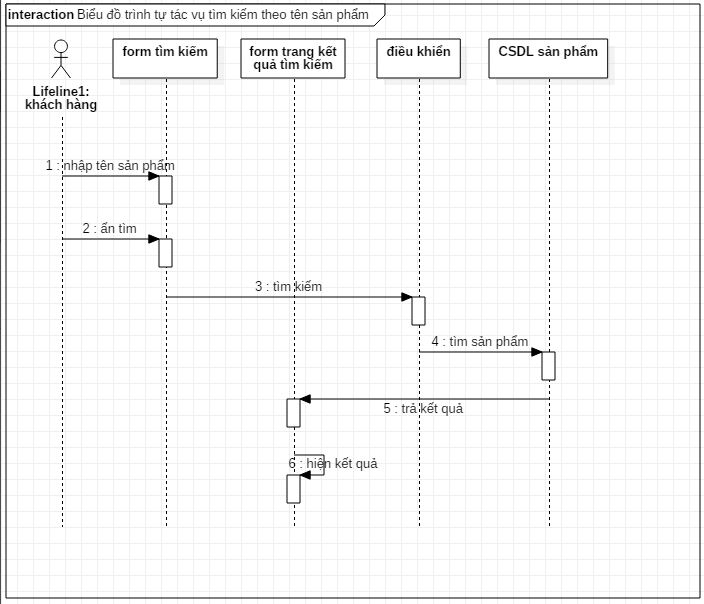
Hình 13. Biểu đồ trình tự cho tác vụ đăng nhập hệ thống

* + Khách hàng cập nhật thông tin cá nhân



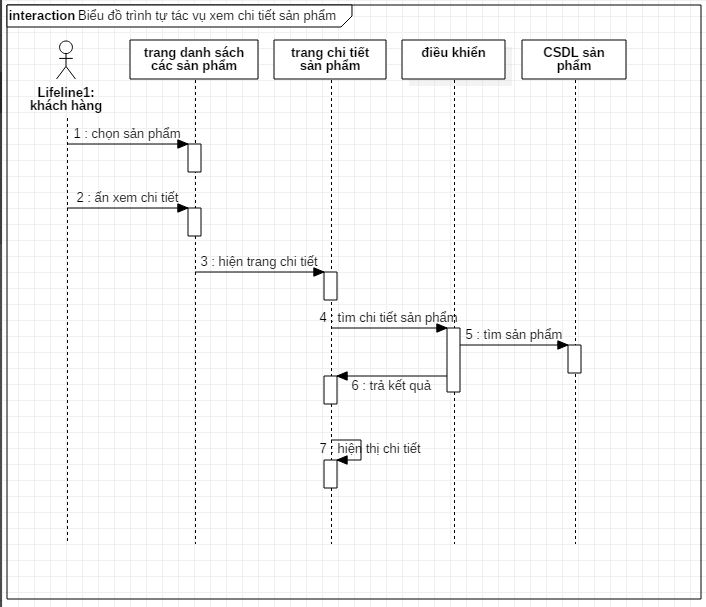
Hình 14. Biểu đồ trình tự cho tác vụ cập nhật thông tin

* + Khách hàng tìm kiếm theo tên sản phẩm



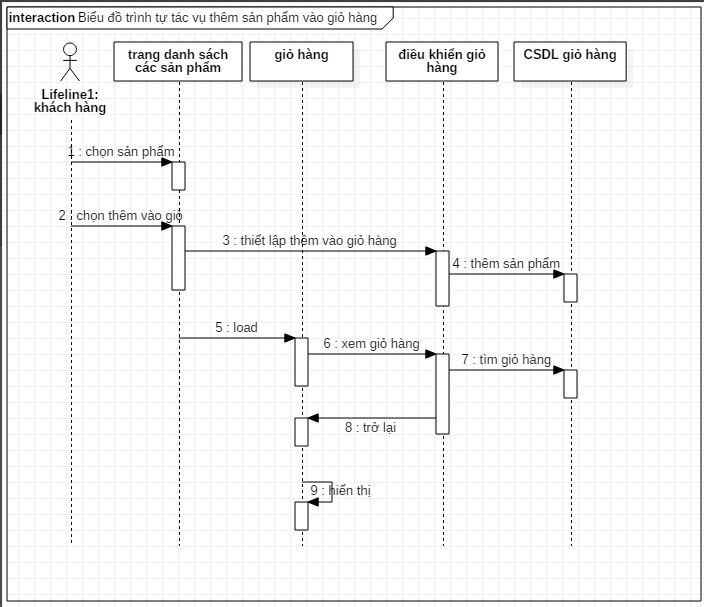
Hình 15. Biểu đồ trình tự cho tác vụ tìm kiếm

* + Khách hàng xem chi tiết sản phẩm



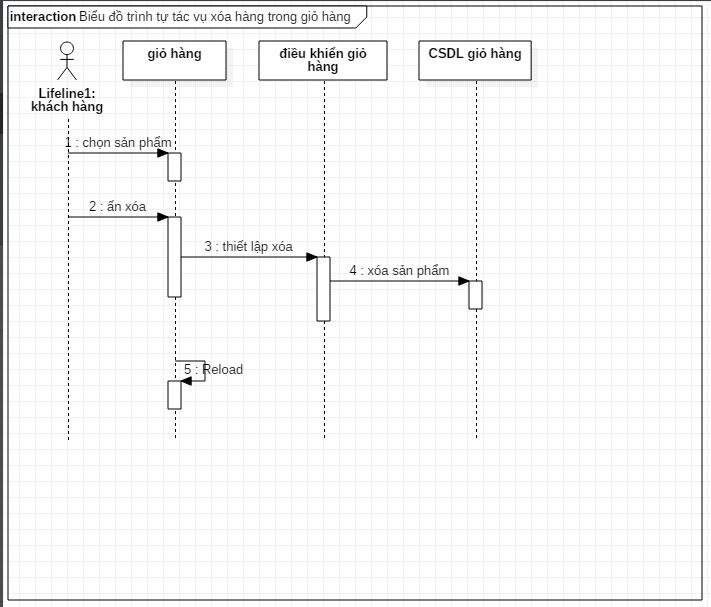
Hình 16. Biểu đồ trình tự cho tác vụ xem chi tiết sản phẩm

* + Khách hàng thêm sản phẩm vào giỏ hàng



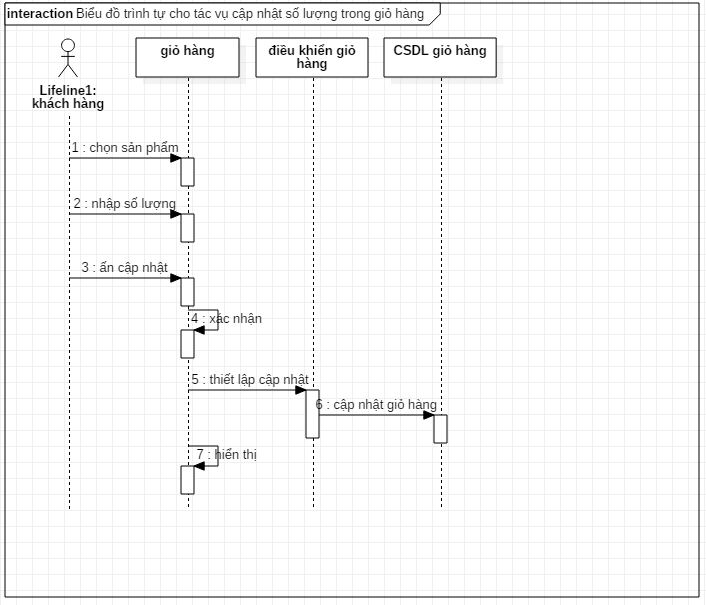
Hình 17. Biểu đồ trình tự cho tác vụ thêm sản phẩm vào giỏ hàng

* + Khách hàng xóa sản phẩm trong giỏ hàng



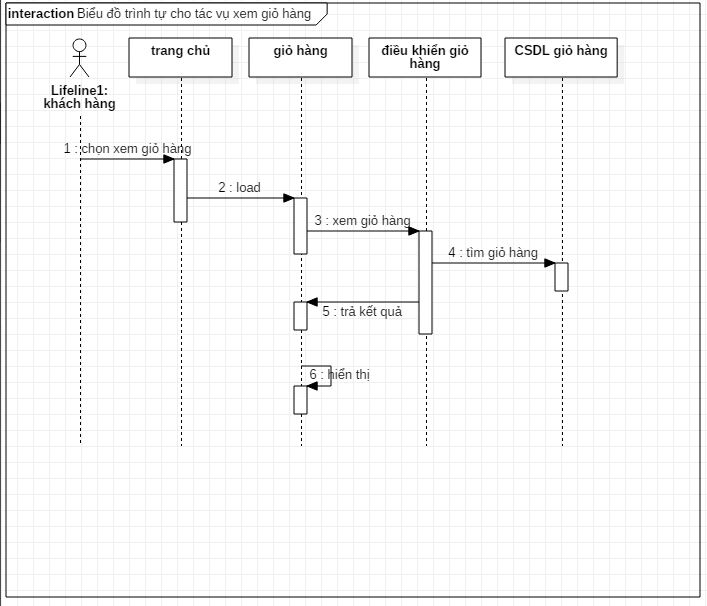
Hình 18. Biểu đồ trình tự cho tác vụ xóa sản phẩm trong giỏ hàng

* + Khách hàng cập nhật số lượng trong giỏ hàng



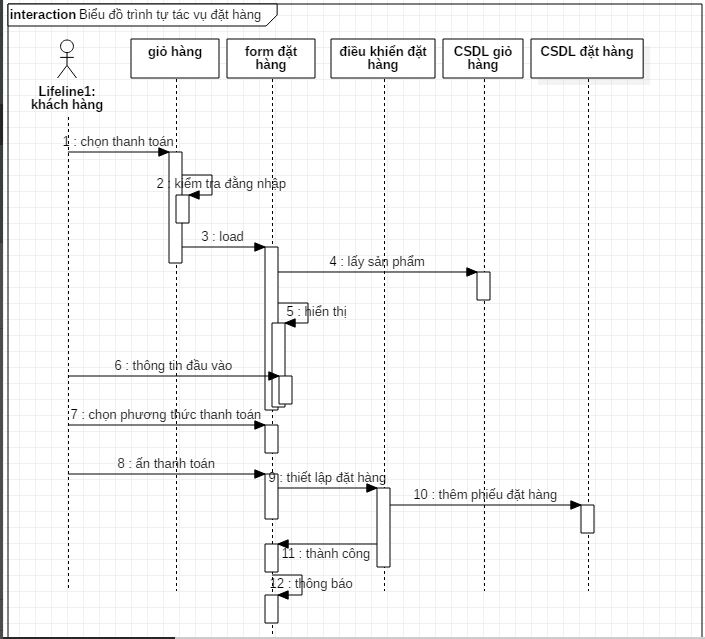
Hình 19. Biểu đồ trình tự cho tác vụ cập nhật số lượng sản phẩm trong giỏ hàng

* + Khách hàng xem giỏ hàng



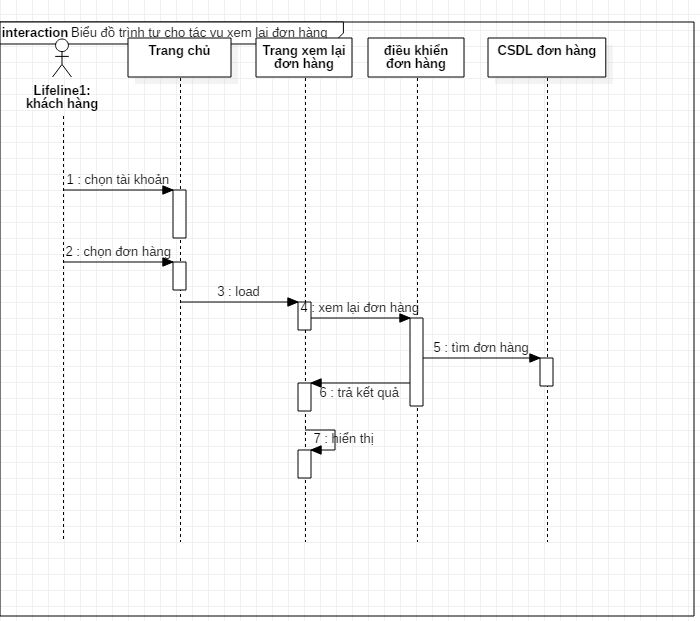
Hình 20. Biểu đồ trình tự cho tác vụ xem sản phẩm trong giỏ hàng

* + Khách hàng thanh toán



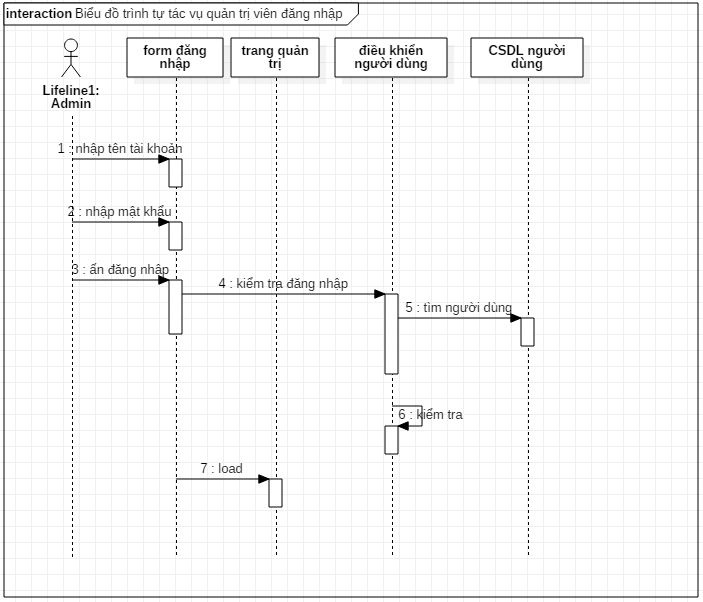
Hình 22. Biểu đồ trình tự cho tác vụ thanh toán

* + Khách hàng xem lại đơn hàng



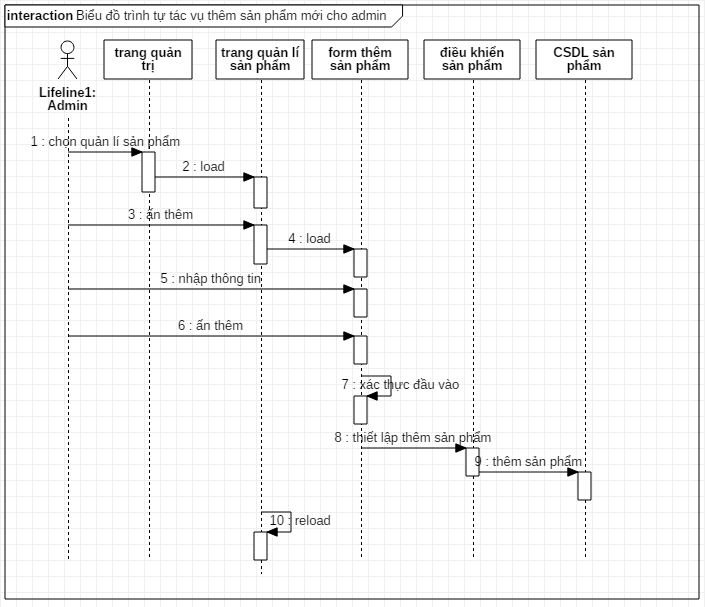
Hình 23. Biểu đồ trình tự cho tác vụ xem lại đơn hàng

* **Phần quản trị hệ thống**
  + Quản trị viên đăng nhập hệ thống



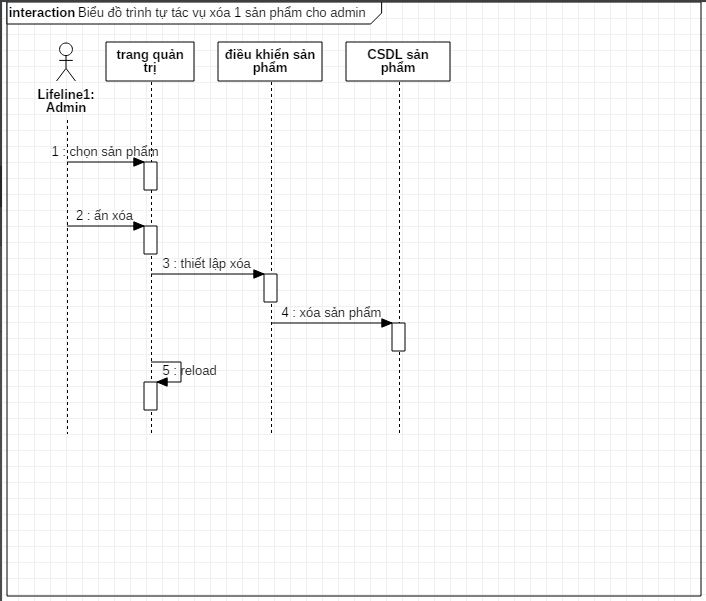
Hình 24. Biểu đồ trình tự cho tác vụ quản trị viên đăng nhập hệ thống

* + Quản trị viên thêm sản phẩm mới



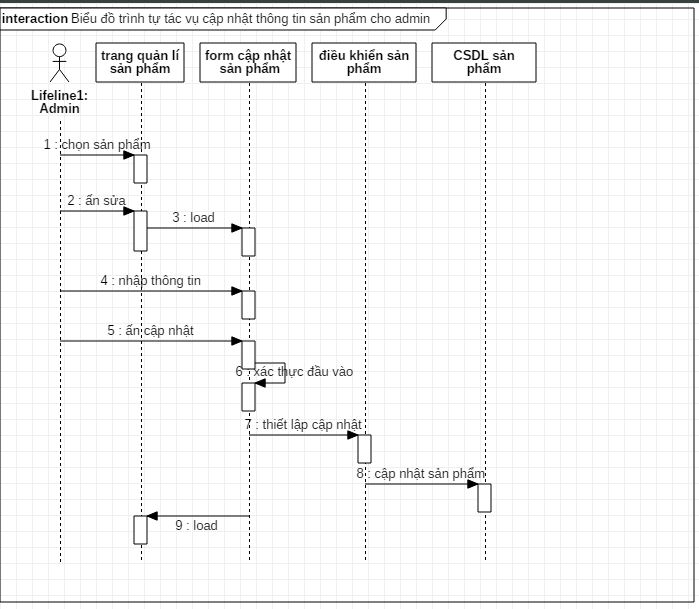
Hình 25. Biểu đồ trình tự cho tác vụ quản trị viên thêm sản phẩm mới

* + Quản trị viên xóa sản phẩm



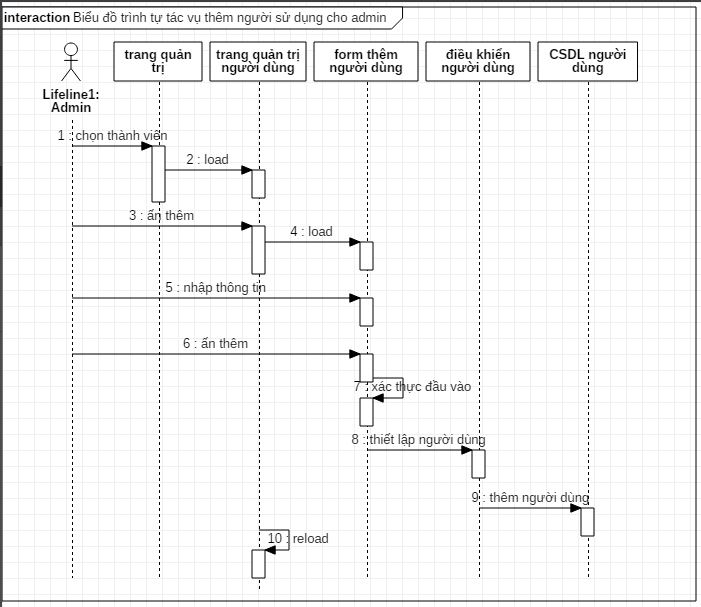
Hình 26. Biểu đồ trình tự cho tác vụ quản trị viên xóa sản phẩm

* + Quản trị viên cập nhật thông tin sản phẩm



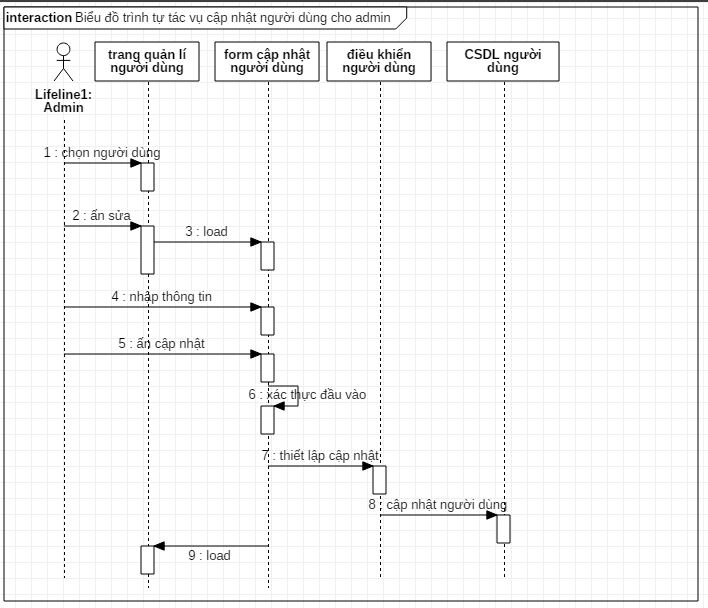
Hình 27. Biểu đồ trình tự cho tác vụ quản trị viên cập nhật thông tin sản phẩm

* + Quản trị viên thêm người dùng



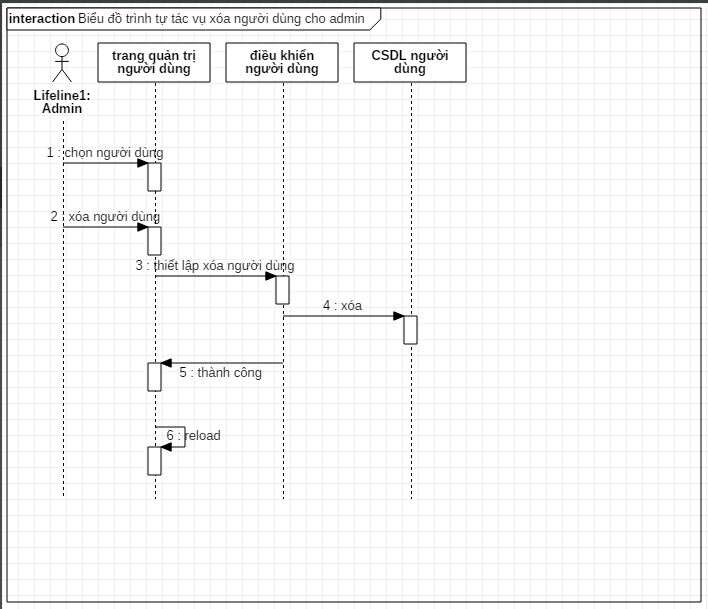
Hình 28. Biểu đồ trình tự cho tác vụ quản trị viên thêm người dùng

* + Quản trị viên sửa thông tin người dùng



Hình 29. Biểu đồ trình tự cho tác vụ quản trị viên cập nhật thông tin người dùng

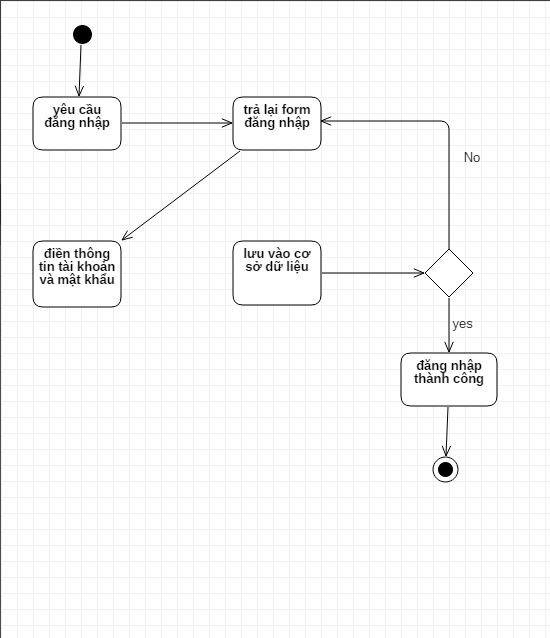
* + Quản trị viên xóa người dùng



Hình 30. Biểu đồ trình tự cho tác vụ quản trị viên xóa người dùng

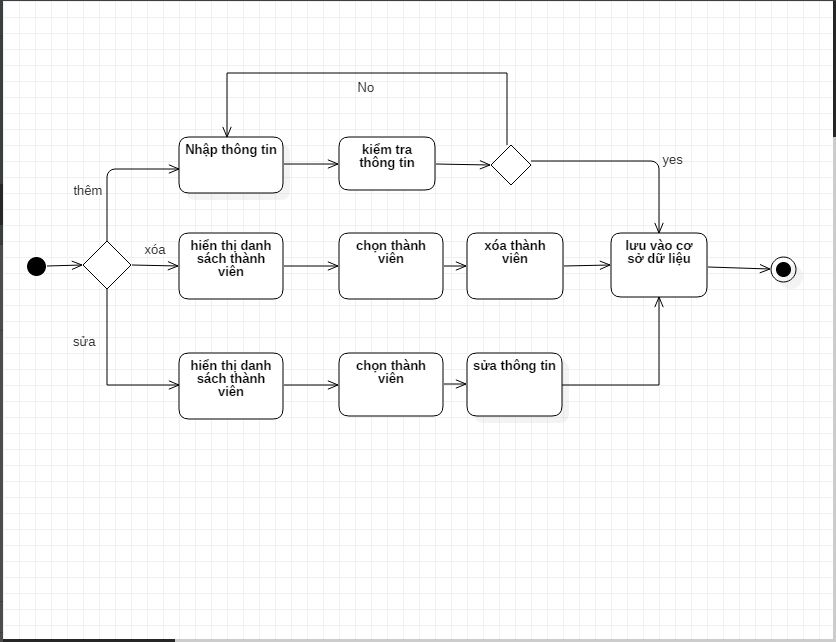
## 2.2.4. Biểu đồ hoạt động

* Biểu đồ hoạt động chức năng đăng nhập:



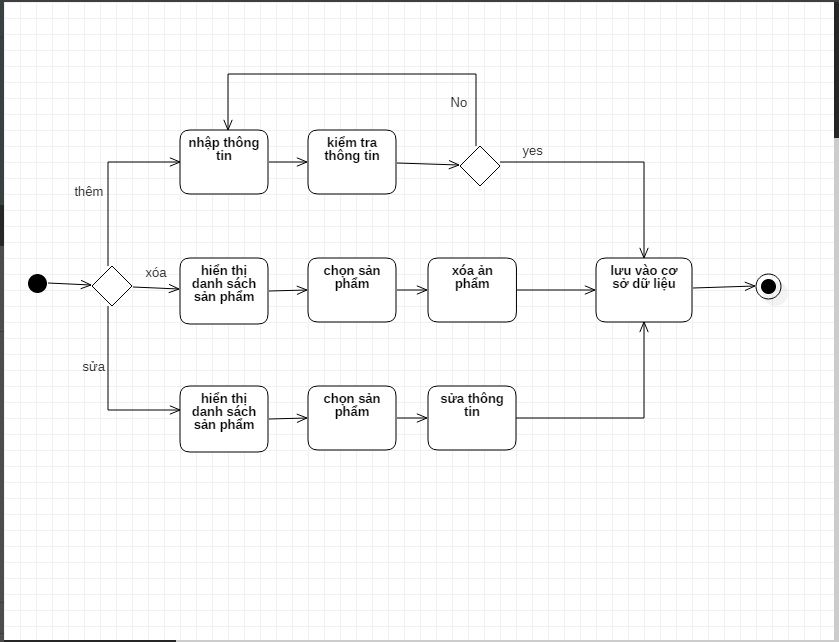
Hình 31. Biểu đồ hoạt động cho chức năng đăng nhập

* Biểu đồ hoạt động chức năng quản lý khách hàng:



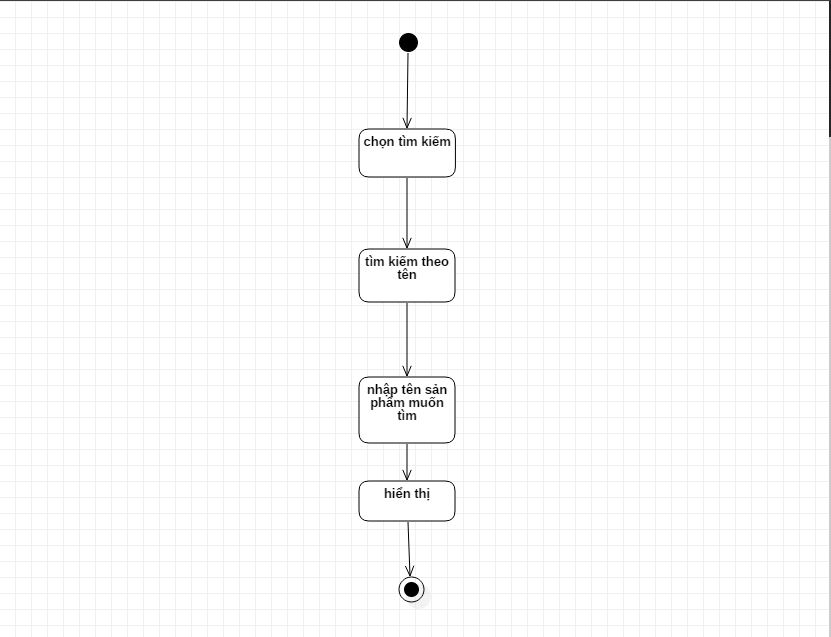
Hình 32. Biểu đồ hoạt động cho chức năng quản lý khách hàng

* Biểu đồ hoạt động chức năng quản lý sản phẩm:



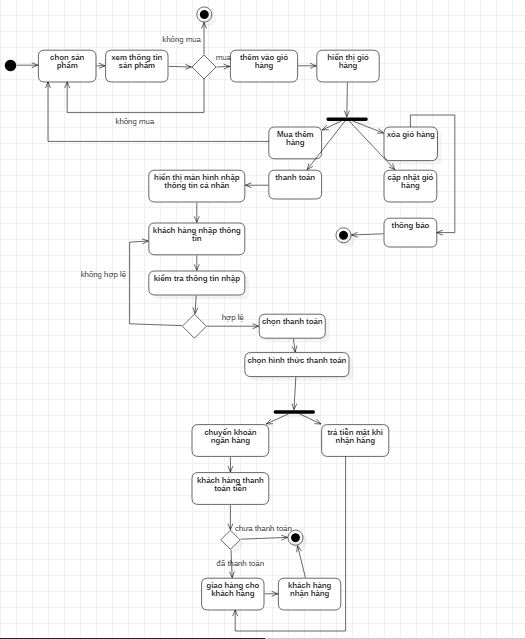
Hình 33. Biểu đồ hoạt động cho chức năng quản lý sản phẩm

* Biểu đồ hoạt động chức năng tìm kiếm:



Hình 34. Biểu đồ hoạt động cho chức năng tìm kiếm

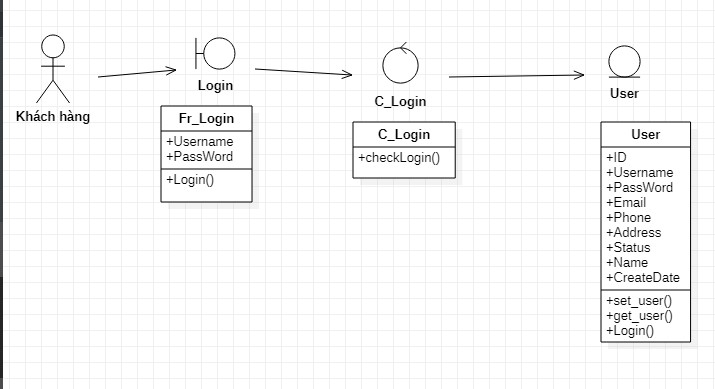
* Biểu đồ hoạt động cho chức năng mua hàng:



Hình 35. Biểu đồ hoạt động cho chức năng mua hàng

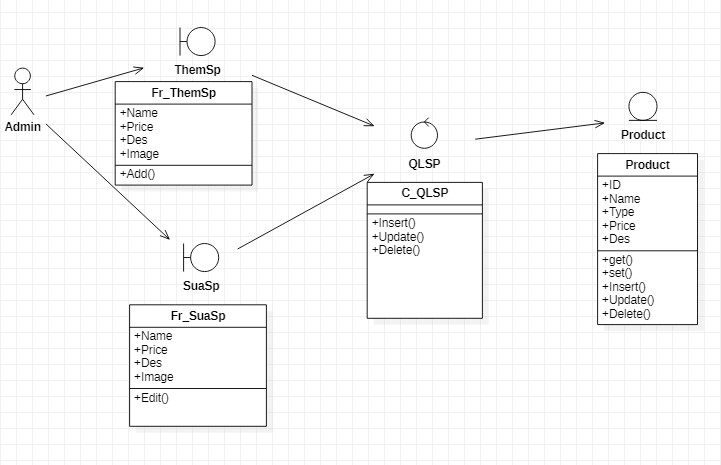
## 2.2.5 Biểu đồ lớp

* Biểu đồ lớp của chức năng đăng nhập:



Hình 36. Biểu đồ lớp cho chức năng đăng nhập

* Biểu đồ lớp của chức năng quản lý sản phẩm:

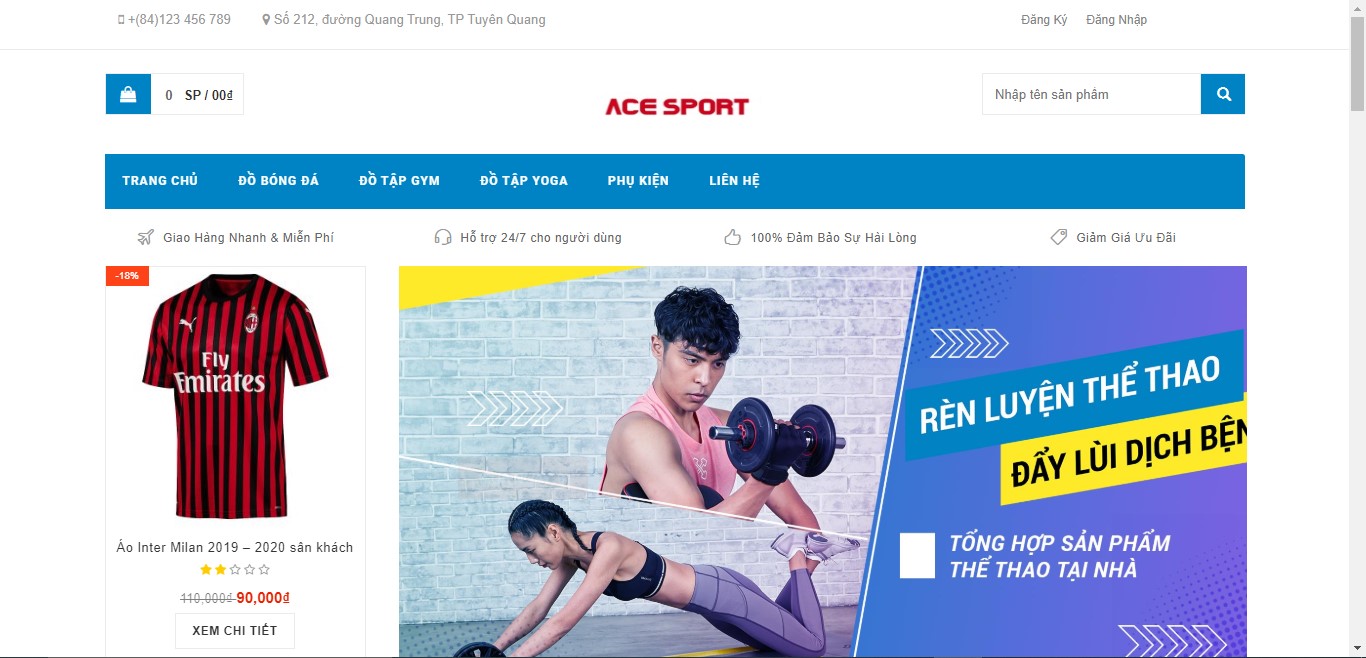


Hình 37. Biểu đồ lớp cho chức năng quản lý sản phẩm

# CHƯƠNG 3: XÂY DỰNG WEBSITE BÁN ĐỒ THỂ THAO

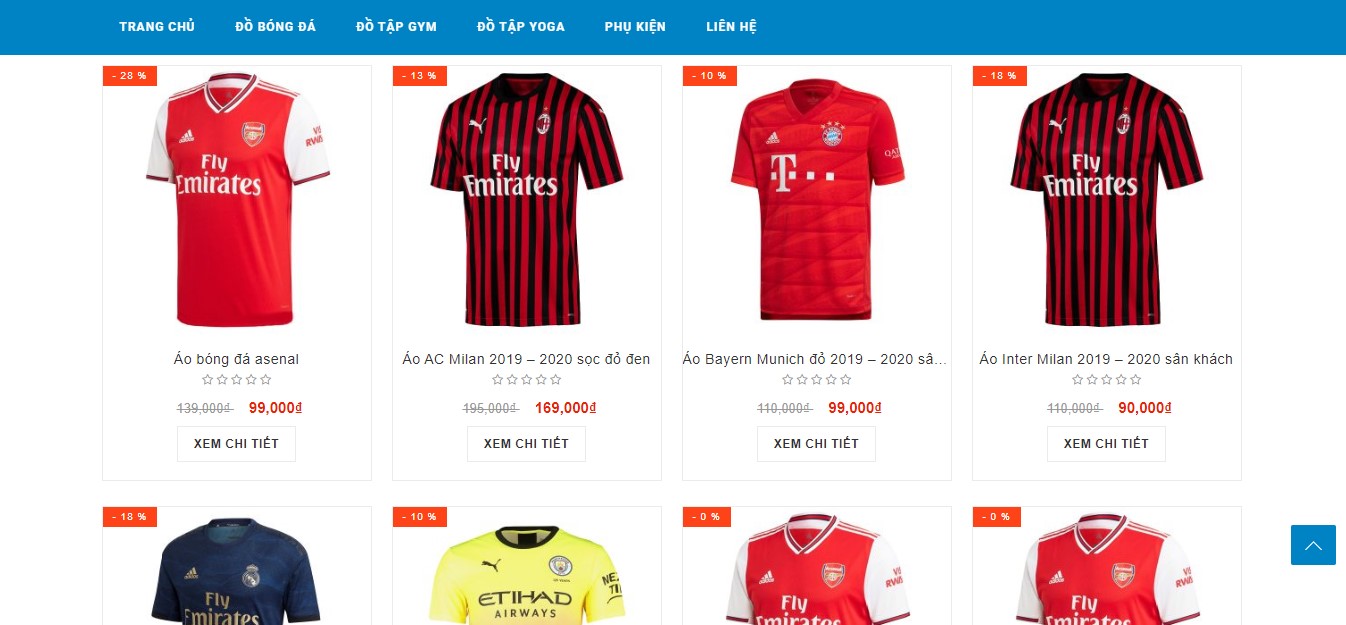
# 3.1 Trang dành cho khách hàng

## 3.1.1 Giao diện Trang chủ



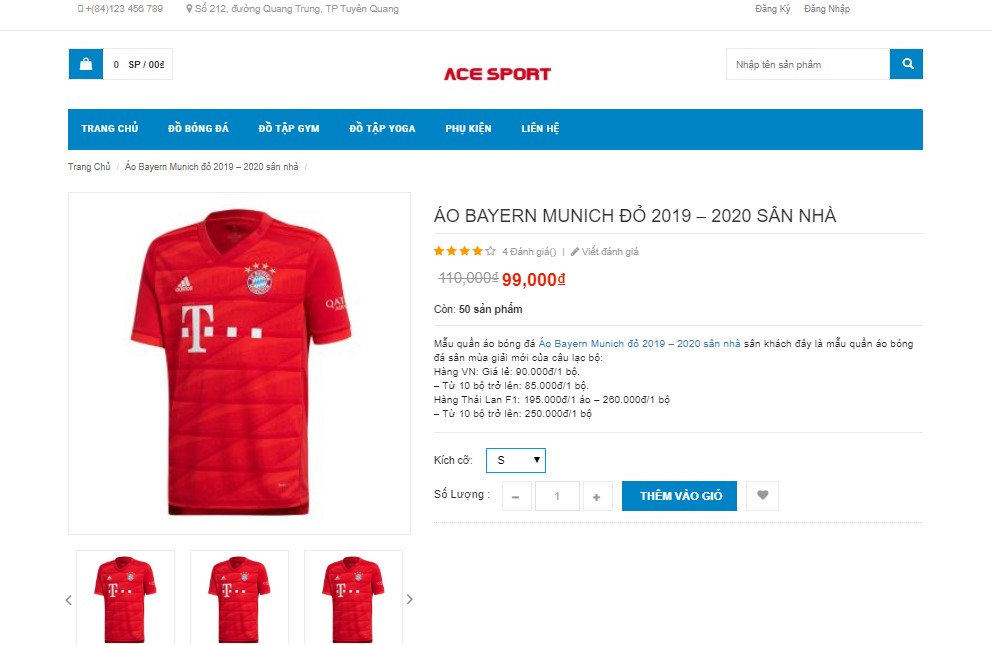
Hình 38. Giao diện Trang chủ

## 3.1.2 Giao diện Trang danh mục đồ bóng đá



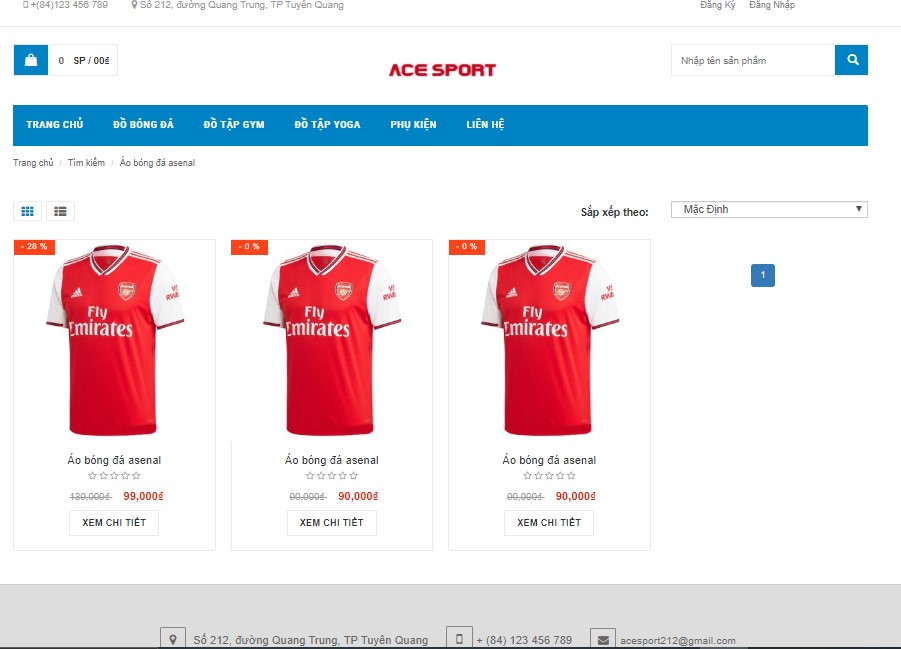
Hình 39. Giao diện Trang danh mục đồ bóng đá

## 3.1.3 Giao diện Trang chi tiết sản phẩm



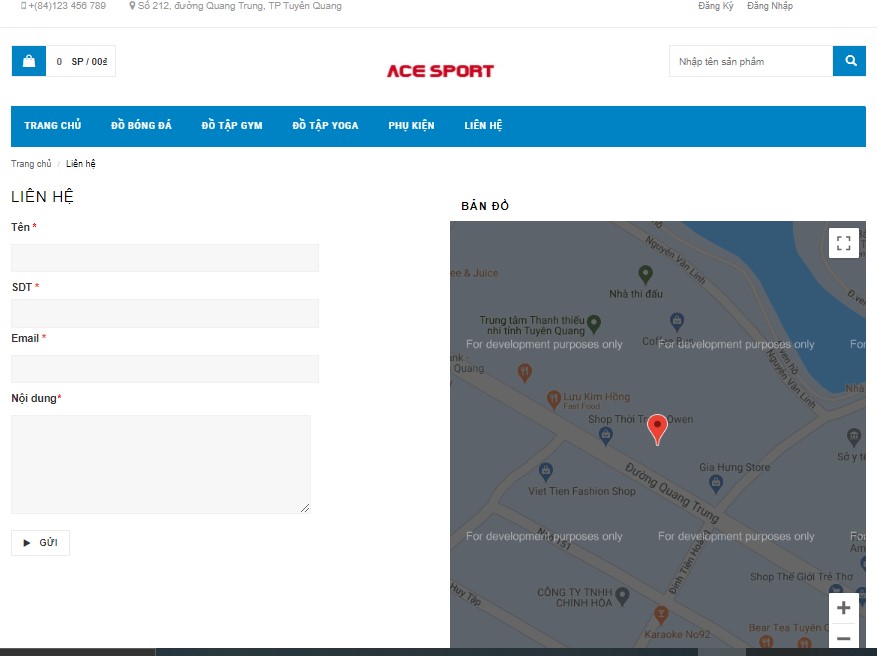
Hình 40. Giao diện Trang chi tiết sản phẩm

## 3.1.4 Giao diện Trang tìm kiếm



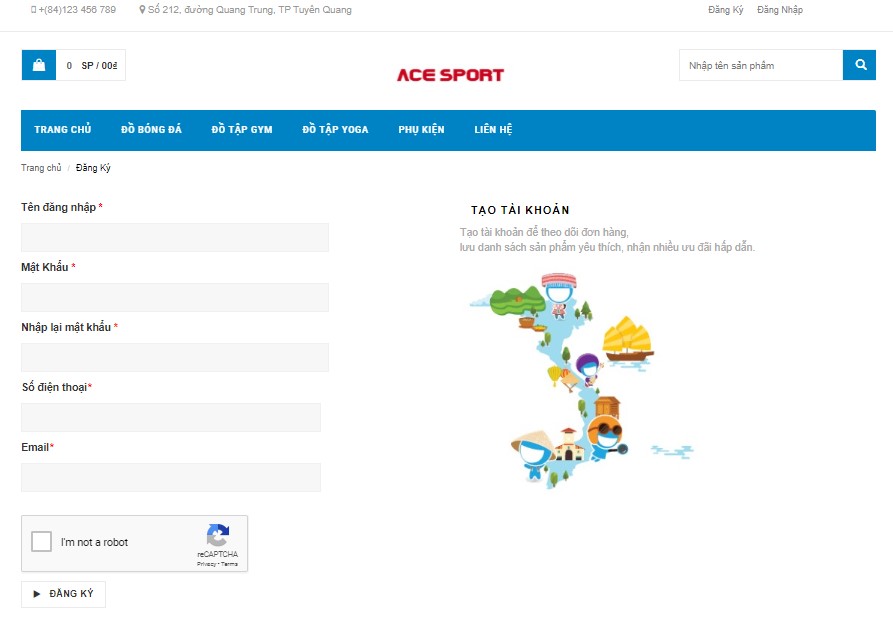
Hình 41. Giao diện Trang tìm kiếm

## 3.1.5 Giao diện Trang liên hệ



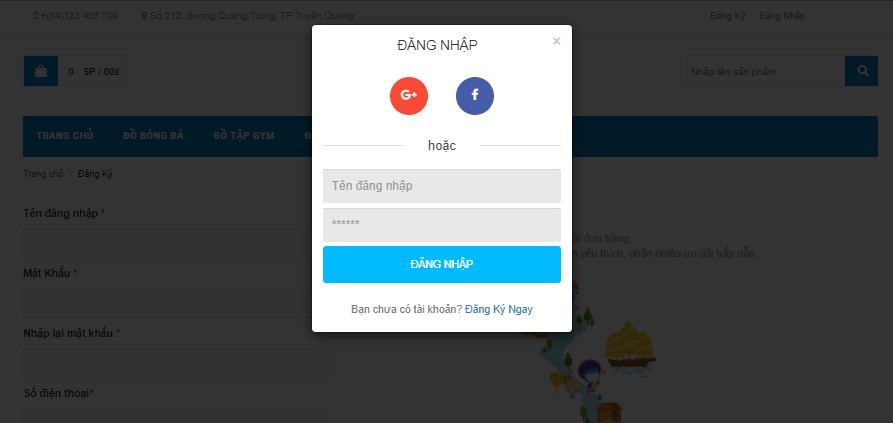
Hình 42. Giao diện Trang liên hệ

## 3.1.6 Giao diện Trang đăng ký



Hình 43. Giao diện Trang đăng ký

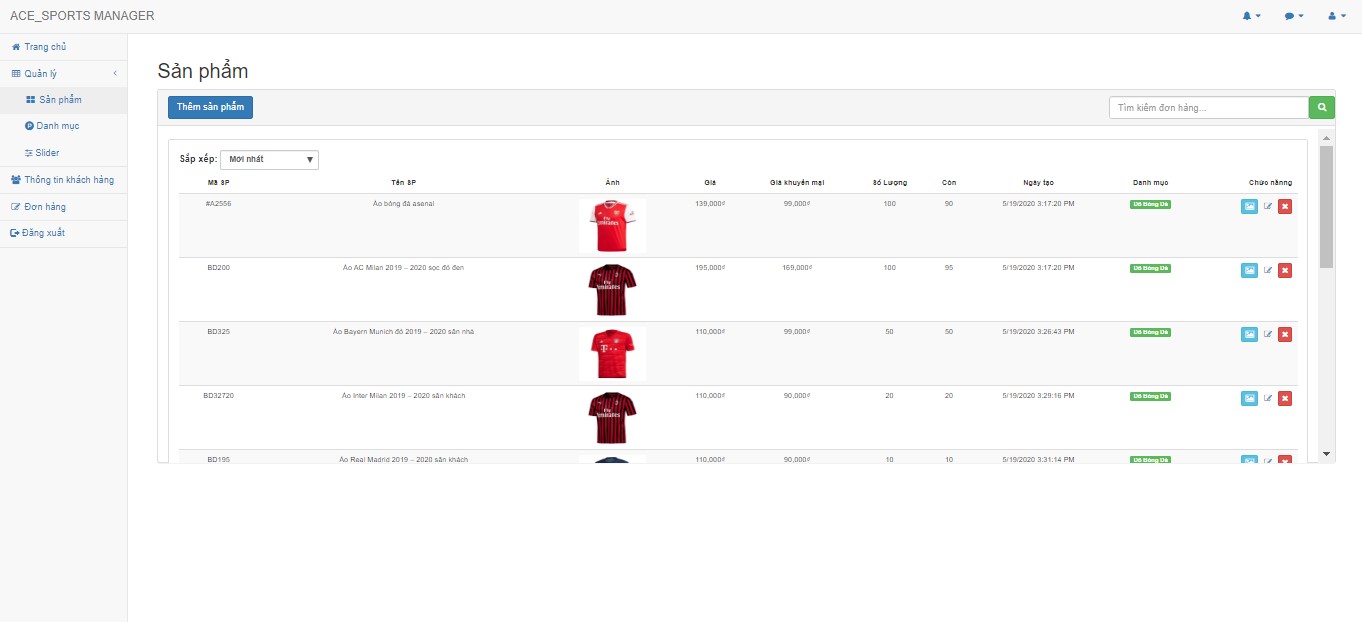
## 3.1.7 Giao diện Trang đăng nhập



Hình 44. Giao diện Trang đăng nhập

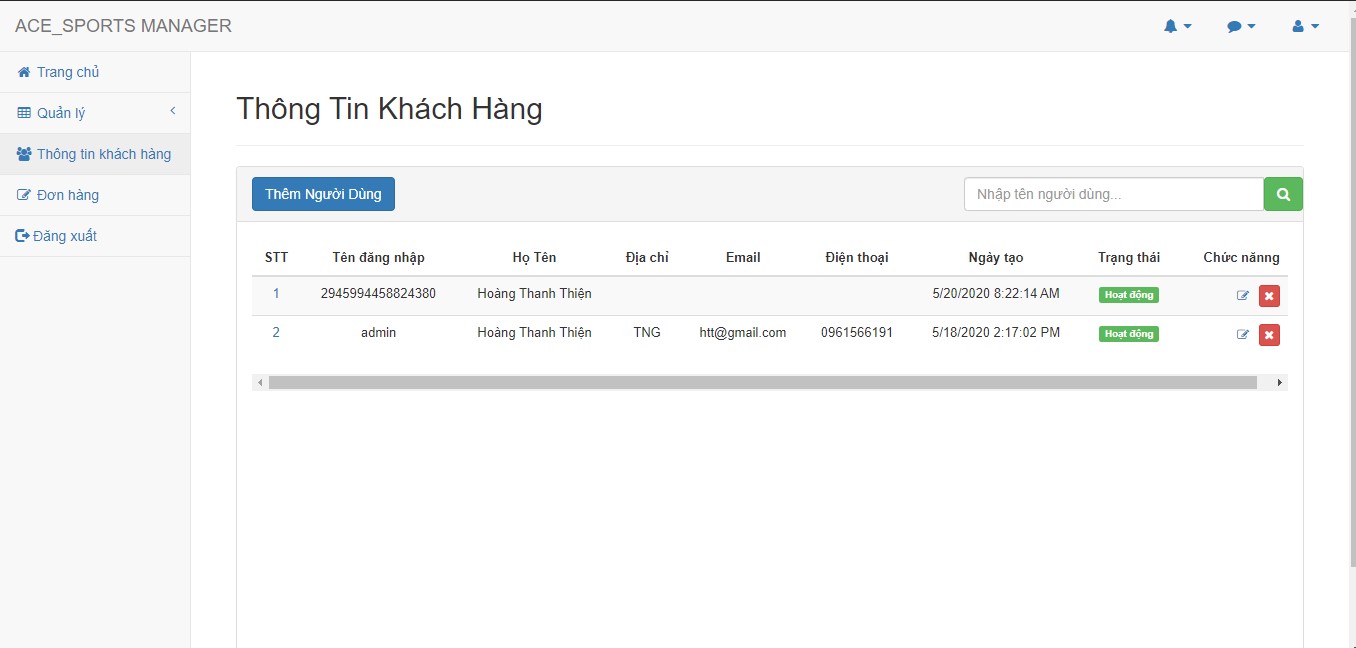
# 3.2 Giao diện trang quản lý

## 3.2.1 Trang quản lý sản phẩm



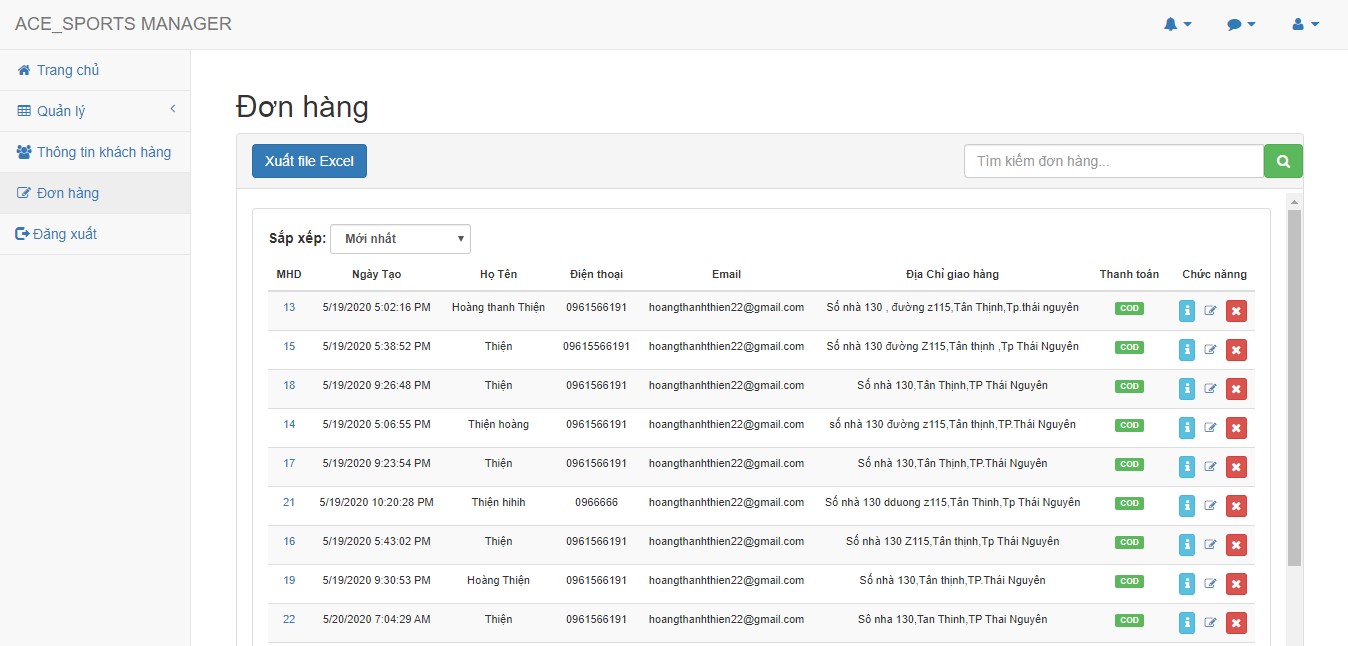
Hình 45. Giao diện Trang quản lý sản phẩm

## 3.2.2 Trang quản lý khách hàng



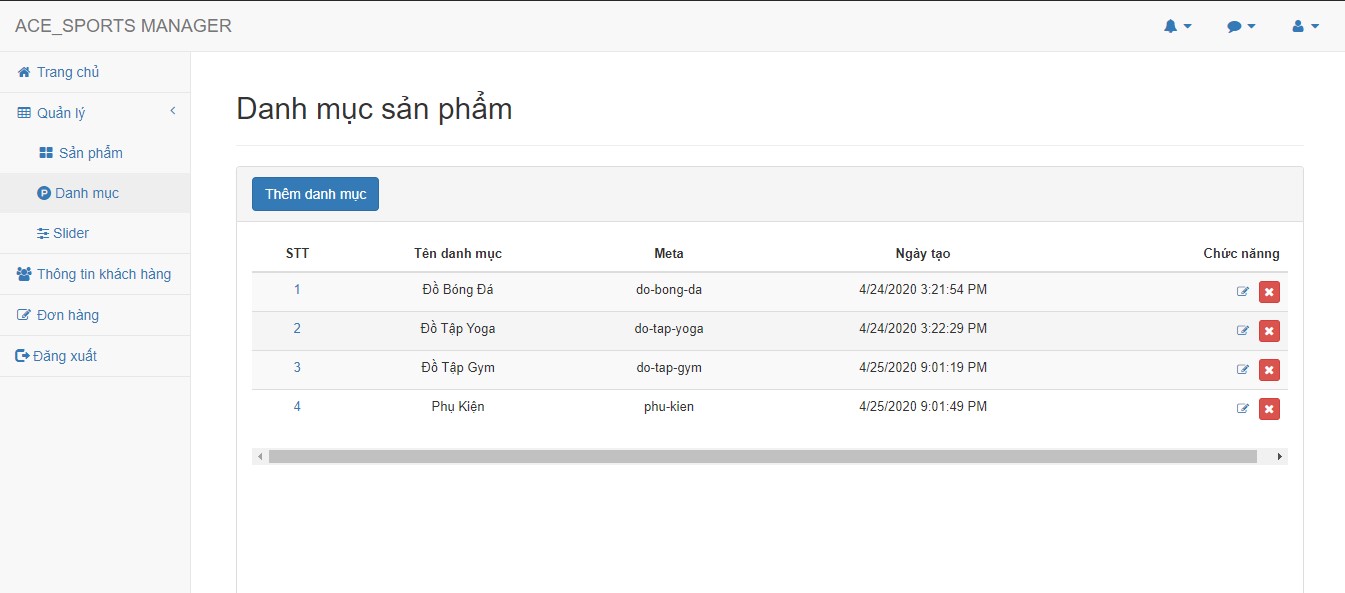
Hình 46. Giao diện Trang quản lý khách hàng

## 3.2.3 Trang quản lý đơn hàng



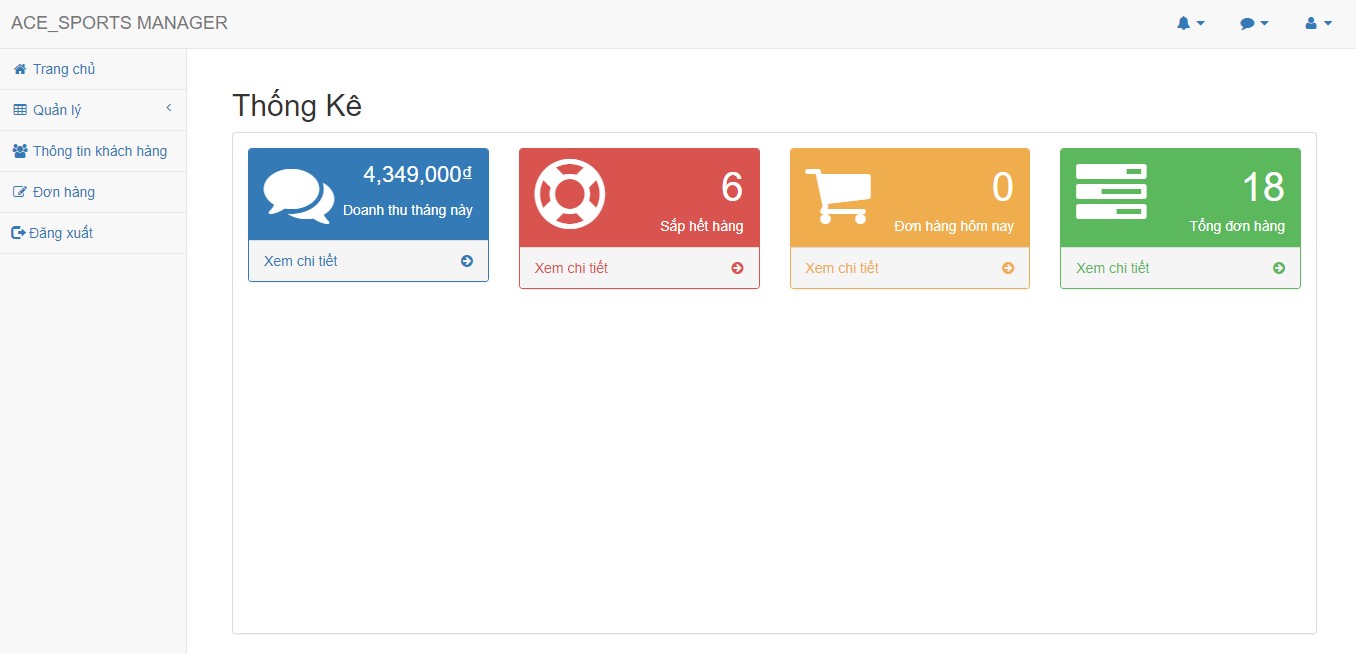
Hình 47. Giao diện Trang quản lý khách hàng

## 3.2.4 Trang quản lý danh mục sản phẩm



Hình 48. Giao diện Trang quản lý danh mục sản phẩm

## 3.2.5 Trang thống kê



Hình 49. Giao diện Trang quản lý danh mục sản phẩm

# KẾT LUẬN

***1. Kết quả đạt được***

- Nắm vững được kiến thức trong phân tích thiết kế hệ thống.

- Nắm vững cách xây dựng một Website sử dụng công nghệ .NET, hệ quản trị cơ sở dữ liệu SQL Server.

- Website hoạt động tốt trên môi trường Localhost .

- Địa chỉ website: <http://acesport.somee.com/>.

- Thực hiện được một số tính năng chính cho Website ,thực hiện các chức năng quản lí nội dung.

- Cửa hàng có thể đưa Website vào sử dụng nhằm hỗ trợ tốt hơn cho công việc của cửa hàng.

- Hiểu được cách cài đặt và sử dụng các package hỗ trợ trong ASP.NET MVC.

- Hiểu được các chức năng để phát triển một website .

- Hiểu được nguyên lý hoạt động trong ASP.NET MVC.

***2. Hạn chế***

- Giao diện còn chưa được tối ưu, chưa thực sự hấp dẫn người xem.

- Các tính năng của Website còn chưa được tối ưu, đôi khi gây khó khăn cho người sử dụng mới.

- Tính năng thanh toàn bằng tài khoản Ngân lượng chưa được tích hợp.

***3. Hướng phát triển đề tài***

- Thay đổi một số tính năng giúp người sử dụng dễ dàng thao tác hơn.

- Thay đổi giao diện để Website nâng cao tính thẩm mĩ hơn.

- Tích hợp ví điện tử vào website.

- Phát triển thêm hệ thống lớn trên website, đưa vào ứng dụng thực tế.

# TÀI LIỆU THAM KHẢO

[1] *Bài giảng phân tích thiết kế hướng đối tượng****,*** Trường ĐH CNTT & TT

[2] Sử dụng DHTML & CSS, *Thiết kế Web Động,* NXB Lao động xã hội, 2005.

[3] Lê Minh Hoàng, *Thiết Kế Web với CSS*, NXB Lao động xã hội, 04/2007.

[4] Beginning ASP.NET : From Novice to Professional

[5] [http://www.w3schools.com](http://www.w3schools.com/)

[6] <https://stackoverflow.com/>

[7] <https://docs.microsoft.com/en-us/aspnet/mvc/>