MỤC LỤC

[MỤC LỤC 2](#_Toc162989449)

[DANH SÁCH HÌNH VẼ 4](#_Toc162989450)

[DANH SÁCH BẢNG BIỂU 5](#_Toc162989451)

[DANH SÁCH TỪ VIẾT TẮT 6](#_Toc162989452)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 7](#_Toc162989453)

[1.1 Lý do chọn đề tài 7](#_Toc162989454)

[1.2 Mục tiêu của đề tài 7](#_Toc162989455)

[1.3 Giới hạn và phạm vi của đề tài 8](#_Toc162989456)

[1.4 Nội dung thực hiện 8](#_Toc162989457)

[1.5 Phương pháp tiếp cận 8](#_Toc162989458)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 9](#_Toc162989459)

[2.1 Tổng quan về ngôn ngữ Asp.NET MVC 9](#_Toc162989460)

[2.1.1 Giới thiệu Asp.NET MVC Framework 9](#_Toc162989461)

[2.1.2 Lý do chọn Asp.NET MVC 15](#_Toc162989462)

[2.1.3 Ưu điểm của Asp.NET MVC 17](#_Toc162989463)

[2.2 Tổng quan về SQL SERVER 18](#_Toc162989464)

[2.2.1 Hệ quản trị CSDL SQL Server 18](#_Toc162989465)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 21](#_Toc162989466)

[3.1 Đối tượng sử dụng website 21](#_Toc162989467)

[3.2 Chức năng chính của Website 21](#_Toc162989468)

[3.2.1 Chức năng đối với khách hàng 21](#_Toc162989469)

[3.2.2 Chức năng của Admin 26](#_Toc162989470)

[3.3 Phân tích thiết kế hệ thống 29](#_Toc162989471)

[3.3.1 Các yêu cầu chức năng chính của hệ thống 29](#_Toc162989472)

[3.3.2 Các biểu đồ Use Case 32](#_Toc162989473)

[3.3.3 Đặc tả Use Case 34](#_Toc162989474)

[3.3.4 Các yêu cầu phi chức năng của hệ thống 51](#_Toc162989475)

[3.4 Biểu đồ tuần tự 54](#_Toc162989476)

[3.4.1 Biểu đồ tuần tự Use case đăng kí thành viên 54](#_Toc162989477)

[3.4.2 Biểu đồ tuần tự Use case đăng nhập 55](#_Toc162989478)

[3.4.3 Biểu đồ tuần tự Use case thay đổi thông tin tài khoản 55](#_Toc162989479)

[3.4.4 Biểu đồ tuần tự Use case xóa tài khoản 56](#_Toc162989480)

[3.4.5 Biểu đồ tuần tự Use case thêm sản phẩm 56](#_Toc162989481)

[3.4.6 Biểu đồ tuần tự Use case thay đổi thông tin sản phẩm 57](#_Toc162989482)

[3.4.7 Biểu đồ tuần tự Use case xóa sản phẩm 58](#_Toc162989483)

[3.4.8 Biểu đồ tuần tự Use case đặt hàng 59](#_Toc162989484)

[3.4.9 Biểu đồ tuần tự Use case thêm nhà phân phối 60](#_Toc162989485)

[3.4.10 Biểu đồ tuần tự Use case thay đổi thông tin nhà phân phối 60](#_Toc162989486)

[3.4.11 Biểu đồ tuần tự Use case xóa nhà phân phối 61](#_Toc162989487)

[3.5 Thiết kế cơ sở dữ liệu 62](#_Toc162989488)

[3.5.1 Mô hình cơ sở dữ liệu 62](#_Toc162989489)

[3.5.2 Danh sách các lớp đối tượng 63](#_Toc162989490)

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG BIỂU

DANH SÁCH TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Từ đầy đủ | Giải thích |
| CLR | Common Language Runtime |  |
| HTML | Hyper Text Markup Language |  |
| HTTP | [Hyper Text Transfer Protocol](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) |  |
| OOP | Object Oriented Programming |  |
| MSIL | Microsoft Intermediate Language |  |
| IL | Intermediate Language |  |
| JIT | Just-In-Time |  |
| RDMS | Relational Database Management System |  |
| SQL | Structured Query Language | ngôn ngữ truy vấn mang tính cấu trúc |

# TỔNG QUAN VỀ ĐỀ TÀI

1.1 Lý do chọn đề tài

Thương mại điện tử từ lâu đã không còn là khái niệm mới. Năm 1979 đã chào đón sự ra đời của website mua sắm trực tuyến đầu tiên. Xong so với bây giờ,sau hơn 30 năm phát triển thì sự thay đổi là đáng kể. Những bước chuyển tiến mạnh mẽ về các tính năng như chuyển tiền điện tử, quản lý chuỗi dây chuyền cung ứng,tiếp thị Internet, quá trình giao dịch trực tuyến, trao đổi dữ liệu điện tử, các hệ thống quản lý hàng tồn kho, và các hệ thống tự động thu thập dữ liệu đã được nâng cấp và hợp lý hóa rất nhiều.

Nhận thức rõ tầm quan trọng của thương mại điện tử trong quảng bá thương hiệu hình ảnh, sản phẫm cho doanh nghiệp, đưa đến cho khách hàng cái nhìn chi tiết nhất về doanh nghiệp nên em chọn đề tài ***“Xây dựng bán quần áo thời trang sử dụng asp.net mvc”.***

Website giúp cho khách hàng không mất quá nhiều thời gian, công sức để có thể tìm được một sản phẩm ưng ý. Các thông tin sản phẩm, hình thức thanh toán cũng như giỏ hàng được xây dựng trên hệ thống giúp cho người sử dụng có thể dễ dàng thao tác và thực hiện trực tiếp trên website.

1.2 Mục tiêu của đề tài

* Nâng cao kiến thức, kỹ năng sử dụng trong Asp.NET MVC và cách thức truy vấn trong SQL Server.
* Ứng dụng xây dựng website giới thiệu và bán quần áo thời trang online với các tính năng như sau:
* Giao diện thân thiện, dễ sử dụng.
* Tìm kiếm sản phẩm dễ dàng.
* Cho phép mua hàng trực tuyến qua mạng.
* Quản lý sản phẩm đơn hàng trực quan, sinh động.

1.3 Giới hạn và phạm vi của đề tài

* Phạm vi ứng dụng: Sử dụng trong việc quản lý quần áo thời trang của các cửa hàng, siêu thị, doanh nghiệp vừa và nhỏ thực hiện việc nhập và xuất đồ quần áo thời trang…
* Đối tượng sử dụng: Phần mềm có thể được sử dụng bởi nhân viên bán hàng, nhân viên nhập hàng, người quản lý, khách hàng mua quần áo.

1.4 Nội dung thực hiện

* Xây dựng được website bán quần áo thời trang đáp ứng được yêu cầu đề tài.
* Xây dựng được hệ thống quản lý bán quần áo thời trang đầy đủ chức năng.
* Giải quyết tối ưu hóa yêu cầu của khách hàng trong quá trình mua hàng.
* Đưa ra được các báo cáo, thống kê tổng hợp.

1.5 Phương pháp tiếp cận

* Cách tiếp cận : Nghiên cứu Asp.Net MVC
* Sử dụng các phương pháp nghiên cứu:
* Phương pháp đọc hiểu tài liệu : kiến thức ASP.NET MVC5 hiểu về linq, html, jQuery và các framework và teamplate khác..
* Phương pháp thực nghiệm.

# CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về ngôn ngữ Asp.NET MVC

Mẫu kiến trúc Model – View – Controller được sử dụng nhằm chi ứng dụng thành ba thành phần chính: model, view và controller. Nền tảng ASP.NET MVC giúp cho chúng ta có thể tạo được các ứng dụng web áp dụng mô hình MVC thay vì tạo ứng dụng theo mẫu ASP.NET Web Forsm. Nền tảng ASP.NET MVC có đặc điểm nổi bật là nhẹ (lighweigt), dễ kiểm thử phần giao diện (so với ứng dụng Web Forms), tích hợp các tính năng có sẵn của ASP.NET. Nền tảng ASP.NET MVC được định nghĩa trong namespace System.Web.Mvc và là một phần của name space System.Web.

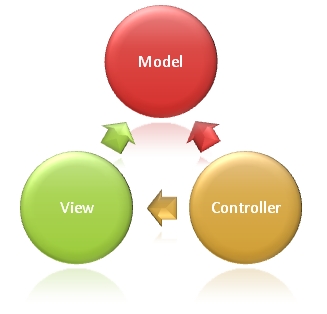
MVC là một mẫu thiết kế (design pattern) chuẩn mà nhiều lập trình viên đã quen thuộc. Một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC. Một số khác vẫn thích hợp với ASP.NET Web Forms và cơ chế postbacks. Đôi khi có những ứng dụng kết hợp cả hai kiến trúc trên.

2.1.1 Giới thiệu Asp.NET MVC Framework

MVC là tên phương pháp chia nhỏ một ứng dụng thành ba thành phần để cài đặt, mỗi thành phần đóng một vai trò khác nhau ảnh hưởng lẫn nhau, đó là model, views, controllers.

MVC là một mẫu thiết kế chuẩn mà nhiều lập trình viên đã quen thuộc. một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC. một số khác vẫn thích hợp với ASP.NET webform và cơ chế postbacks. Đôi khi có những ứng dụng kết hợp với cả hai kiến trúc trên.

Nền tảng MVC bao gồm các thành phần dưới đây :



Hình 2.1: Mẫu model -View - Controller

**Models :**Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.

Trong các ứng dụng nhỏ, model thường là chỉ là một khái niệm nhằm phân biệt hơn là được cài đặt thực thụ, ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gởi chúng đến view, ứng dụng khong cần phải có tầng model và các lớp liên quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model)

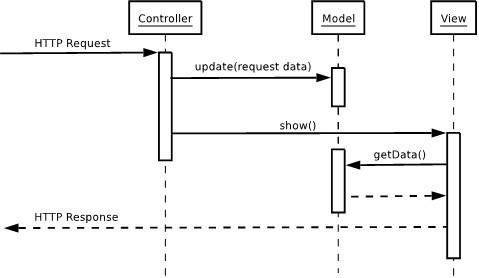
**Views:**  là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.

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

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

Để quản lý sự phức tạp của ứng dụng, mẫu MVC giúp cho chúng ta có thể kiểm thử ứng dụng dễ dàng hơn hẳn so với khi áp dụng mẫu Web Forms. Ví dụ, trong một ứng dụng ASP.NET Web Forms, một lớp thường được sử dụng để hiển thị thông tin xuất ra cho người dùng và đồng thời xử lý thông tin người dùng nhập. Việc xây dựng các bộ test tự động cho ứng dụng Web Forms là rất phức tạp, bởi để kiểm thử mỗi trang web, bạn phải khởi tạo đối tượng trang, khởi tạo tất cả các control được sử dụng trong trang và các lớp phụ thuộc trong ứng dụng. Và bởi vì có quá nhiều lớp cần được khởi tạo để chạy được trang, thật khó để có thể viết các test chỉ tập trung vào một khía cạnh nào đó của ứng dụng. Và vì thế, kiểm thử đối với các ứng dụng dứa trên nền tảng Web Forms sẽ khó khăn hơn nhiều so với khi áp dụng trên ứng dụng MVC. Hơn thế nữa, việc kiểm thử trên nền tảng Web Forms yêu cầu phải sử dụng đến web server. Nền tảng MVC phân tách các thành phần và sử dụng các interface (khái niệm giao diện trong lập trình hướng đối tượng), và nhờ đó có thể kiểm thử các thành phần riêng biệt trong tình trạng phân lập với các yếu tố còn lại của ứng dụng

Sự phân tách rạch ròi ba thành phần của ứng dụng MVC còn giúp cho việc lập trình diễn ra song song. Ví dụ như một lập trình viên làm việc với view, lập trình viên thứ hai lo cài đặt logic của controller và lập trình viên thứ ba có thể tập trung vào logic tác vụ của model tại cùng một thời điểm.

****

Hình 2.2 :mô hình tuần tự của MVC

Lấy ví dụ một GUI Component (thành phần đồ họa người dùng ) đơn giản là checkbox. Checkbox có thành phần Model để quản lý trạng thái của nó là check hay uncheck, thành phần View để thể hiện nó với trạng thái tương ưng lên màn hình, và thành phần Controller để xử lý nhưng sự kiện khi có sự tương tác của người sử dụng hoặc các đối tượng khác lên checkbox.

Khi ngươi sử dụng nhần chuột vào Check box , thành phần Controller của Checkbox sẽ xử lý sự kiện này, yêu cầu thành phần Model thay đổi dữ liệu trạng thái. Sau khi thay đổi trạng thái, thành phần Model phát thông điệp đến thành phần View và Controller. Thành phần View của Checkbox nhận được thông điệp sẽ cập nhật lại thể hiện của Checkbox, phản ánh chính sác trạng thái Checkbox do Model lưu giữ. Thành phần Controller nhận được thông điệp do Model gởi tới sẽ có nhưng tương tác phản hòi với người sử dụng nếu cần thiết.

Controller có trách nhiệm chính là điều hướng các yêu cầu của người sử dụng. Như vậy trên toàn ứng dụng của ta, tất cả các request đều sẽ phải đi tới controller. Và tại đây, ứng với các tham số người sử dụng truyền mà ta đưa họ đến một tác vụ nào đó trên ứng dụng.

Tại các tác vụ này, chúng sẽ thông qua lớp model để làm việc và trả kết quả trở về controller. Cuối cùng controller sẽ đẩy dữ liệu thao tác tới view. View là thành phần cuối cùng mà người sử dụng nhận được khi họ giở request tới ứng dụng.

Có thể hiểu controller, giống với kỹ thuật đa cấp với các tác vụ chạy ứng dụng phân cấp theo từng nhánh riêng biệt như: Module, action,….​  
Qua hình vẽ này, ta hiểu rằng. Để có quần áo thời trang tác với các action (hành động) ta cần phải đi qua file index.php. Lúc này file index đóng vai trò như một controller được dùng để điều hướng các request.

Ứng với các request thì nó sẽ trả về một controller khác để xử lý tác vụ một cách cụ thể. Tại controller con, nó sẽ gọi các action riêng biệt.  
Ở đó, thông qua các action mà nó gọi tới các file xử lý giản đơn.

-Model là thành phần chủ yếu được sử dụng để thao tác xử lý dữ liệu. Trong các framework, Model vẫn thường sử dụng theo phương thức Active Record. Một trong những design pattern. Chúng có tác dụng rút ngắn thời gian viết câu truy vấn cho người sử dụng. Biến những câu truy vấn phức tạp trở nên gần gũi và thân thiện với người sử dụng thông qua các thư viện được định nghĩa sẵn.

Model thường sẽ là các phương thức có trách nhiệm xử lý các tác vụ như: select, insert, update, delete các record trong database. Ứng với các lấy dữ liệu, model thường sử dụng mảng để gởi trả kết quả về. Vì mảng có thể cho phép model lưu trữ nhiều thông tin hơn, nên thường các record khi bóc tách chúng sẽ mang các dữ liệu của database một cách chi tiết.  
Khi sử dụng models, ta cũng cần tuân theo nguyên tắc chính của chúng là không xuất giá trị trực tiếp trong model. Mà tất cả những dữ liệu ấy, phải đưa vào mảng và trả về theo phương thức. Và tiếp tục ở view ta sẽ sử dụng nó để lấy dữ liệu ra.

**Ví dụ:**

Mã:

function getuser($id){

$sql="select \* from info where id='$id'";

$this->query($sql);

$row=$this->fetch();

return $row;}

**Tìm hiều về View**

View là phần hiển thị thông tin tương phản khi gởi và nhận request. Trước đây, khi người lập trình chưa nghĩ tới view. Họ thường thao tác xử lý dữ liệu ngay trực tiếp trên ứng dụng và đổ cả dữ liệu ngay trên file PHP đó. Điều này làm cho ứng dụng trở nên cồng kềnh, và đặc biệt rất khó cho việc bảo trì nâng cấp sau này. Nhất là đối với designer, việc thay đổi giao diện của một website luôn làm cho họ cảm thấy đau đầu vì phải vọc thẳng vào core.  
Trước đây, để giải quyết tình huống này. Người ta thường sử dụng template để phân tách website thành 2 mảng riêng biệt. Một là giao diện và một là core. Việc chỉnh sửa giao diện trở nên đơn giản hơn đối với họ so với cách viết thập cẩm kia. Tuy nhiên, các thư viện này thực chất sẽ làm cho ứng dụng của chúng ta trở nên chậm chạp hơn bao giờ hết. Bởi chúng phải phiên dịch nhiều lần các kịch bản.

Chẳng hạn: Để dễ thao tác, smarty sẽ dịch ngược các yêu cầu của bạn sang ngôn ngữ của nó. Sau đó chúng sẽ chuyển ngôn ngữ đó sang PHP và thao tác xử lý trên nó.

Việc này sẽ làm ứng dụng chậm chạp, do cứ phải dịch qua, dịch lại một kịch bản. Trong khi, với sự kết hợp của PHP thuần, ứng dụng của bạn sẽ nhanh và ổn định hơn nhiều.

Và view cũng là một phần trong việc nâng cấp những hạn chế ấy. Chúng giúp giảm thiểu tối đa quá trình biên dịch nhiều lần. Và làm cho ứng dụng trở nên mạnh mẽ và chuyên nghiệp hơn nhiều so với cách lập trình thuần.

2.1.2 Lý do chọn Asp.NET MVC

.Các tính năng của Asp.NET MVC:

* Tách bạch các tác vụ của ứng dụng (logic nhập liệu, business logic, và logic giao diện), dễ dàng kiểm thử và mặc định áp dụng hướng phát triển TDD. Tất cả các tính năng chính của mô hình MVC được cài đặt dựa trên interface và được kiểm thử bằng cách sử dụng các đối tượng mocks, mock object là các đối tượng mô phỏng các tính năng của những đối tượng thực sự trong ứng dụng. Bạn có thể kiểm thử unit-test cho ứng dụng mà không cần chạy controller trong tiến trình ASP.NET, và điều đó giúp unit test được áp dụng nhanh chóng và tiện dụng. Bạn có thể sử dụng bất kỳ nền tảng unit-testing nào tương thích với nền tảng .NET
* MVC là một nền tảng khả mở rộng (extensible) & khả nhúng (pluggable). Các thành phần của ASP.NET MVC được thiết kể để chúng có thể được thay thế một cách dễ dàng hoặc dễ dàng tùy chỉnh. Bạn có thể nhúng thêm view engine, cơ chế định tuyến cho URL, cách kết xuất tham số của action-method và các thành phần khác. ASP.NET MVC cũng hỗ trợ việc sử dụng Dependency Injection (DI) và Inversion of Control (IoC). DI cho phép bạn gắn các đối tượng vào một lớp cho lớp đó sử dụng thay vì buộc lớp đó phải tự mình khởi tạo các đối tượng. IoC quy định rằng, nếu một đối tượng yêu cầu một đối tượng khác, đối tượng đầu sẽ lấy đối tượng thứ hai từ một nguồn bên ngoài, ví dụ như từ tập tin cấu hình. Và nhờ vậy, việc sử dụng DI và IoC sẽ giúp kiểm thử dễ dàng hơn
* ASP.NET MVC có thành phần ánh xạ URL mạnh mẽ cho phép bạn xây dựng những ứng dụng có các địa chỉ URL xúc tích và dễ tìm kiếm. Các địa chỉ URL không cần phải có phần mở rộng của tên tập tin và được thiết kế để hỗ trợ các mẫu định dạng tên phù hợp với việc tối ưu hóa tìm kiếm (URL) và phù hợp với lập địa chỉ theo kiểu REST.
* Hỗ trợ sử dụng đặc tả (các thẻ) của các trang ASP.NET (.aspx) điều khiển người dùng (.ascx) và trang master page (.mater) Bạn có thể dụng các tình năng có sẵn của ASP.NET như dùng lồng các trang master page sử dụng in-line expression (<%=%>), sử dụng server controls ,mẫu, data -binding, địa phương hóa ( localization) và hơn thế nữa.
* Hỗ trợ các tính năng có sẵn của ASP.NET như cơ chế xác thực người dùng, quản lý thành viên, quyền, output caching và data caching, seession và profile, quản lý tình trạng ứng dụng, hệ thống cấu hình…
* ASP.NET MVC5 còn bổ sung một view engine mới là Razor View Engine cho phép thiết lập các view nhanh chóng, dễ dàng và tốn ít công sức hơn so với việc sử dụng Web Forms view engine.

2.1.3 Ưu điểm của Asp.NET MVC

Nền tảng ASP.NET MVC mang lại những lợi ích sau:

* Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller
* Nó không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
* Nó sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Nhờ đó bạn có thể thiết kế một hạ tầng quản lý định tuyến. Để có nhiều thông tin hơn, bạn nên xem phần [Front Controller](http://go.microsoft.com/fwlink/?LinkId=106357) trên web site MSDN
* Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)
* Nó hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng

2.2 Tổng quan về SQL SERVER

2.2.1 Hệ quản trị CSDL SQL Server

a ) Khái niệm hệ quản trị cơ sở dữ liệu SQL Server

SQL Server là một hệ thống quản lý cơ sở dữ liệu RDBMS sử dụng Transact-SQL để trao đổi dữ liệu giữa Client computer và SQL Server computer. Một RDBMS bao gồm Databases, Database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera -Byte và có thể phục vụ cùng lúc cho hàng ngàn User. SQL Server 2000 có thể kết hợp "ăn ý" với các Server khác như Microsoft Internet Information Server (IIS), ECommerce Server, Proxy Server...

***b) Lịch sử phát triển***

SQL Server được phát triển từ ngôn ngữ SEQUEL2 bởi IBM theo mô hình Codd tại trung tâm nghiên cứu của IBM ở California ,vào những năm 70 cho hệ thống quản trị CSDL lớn.

Đầu tiên SQL Server được sử dụng trong các ngôn ngữ quản lý CSDL và chạy trên các máy đơn lẻ. Song do sự phát triển nhanh chóng của nhu cầu xây dựng những CSDL lớn theo mô hình khách chủ (trong mô hình này toàn bộ CSDL được tập trung trên máy chủ (Server)). Mọi thao tác xử lý dữ liệu được thực hiện trên máy chủ bằng các lệnh SQL máy trạm chỉ dùng để cập nhập hoặc lấy thông tin từ máy chủ). Ngày nay trong các ngôn ngữ lập trình bậc cao đều có sự trợ giúp của SQL. Nhất là trong lĩnh vực phát triển của Internet ngôn ngữ SQL càng đóng vai trò quan trọng hơn. Nó được sử dụng để nhanh chóng tạo các trang Web động.

SQL đã được viện tiêu chuẩn quốc gia Mỹ (ANSI) và tổ chức tiêu chuẩn quốc tế (ISO) chấp nhận như một ngôn ngữ chuẩn cho CSDL quan hệ .Nhưng cho đến nay chuẩn này chưa đưa ra đủ 100%. Nên các SQL nhúng trong các ngôn ngữ lập trình khác nhau đã được bổ xung mở rộng cho SQL chuẩn cho phù hợp với các ứng dụng của mình. Do vậy có sự khác nhau rõ ràng giữa các SQL.

***c) Đặc điểm của SQL Server và đối tượng làm việc***

Đặc điểm

* SQL là ngôn ngữ tựa tiếng Anh.
* SQL là ngôn ngữ phi thủ tục, nó không yêu cầu ta cách thức truy nhập CSDL như thế nào. Tất cả các thông báo của SQL đều rất dễ sử dụng và ít khả năng mắc lỗi .
* SQL cung cấp tập lệnh phong phú cho các công việc hỏi đáp dữ liệu
* Chèn, cập nhật, xoá các hàng trong một quan hệ.
* Tạo, sửa đổi, thêm và xoá các đối tượng trong của CSDL.
* Điều khiển việc truy nhập tới cơ sở dữ liệu và các đối tượng của CSDL để đảm bảo tính bảo mật của cơ sở dữ liệu.
* Đảm bảo tính nhất quán và sự ràng buộc của CSDL.
* Yêu cầu duy nhất để sử dụng cho các hỏi đáp là phải nắm vững được các cấu trúc CSDL của mình.

***d) Đối tượng làm việc của SQL Server***

Là các bảng (tổng quát là các quan hệ) dữ liệu hai chiều. Các bảng này bao gồm một hoặc nhiều cột và hàng. Các cột gọi là các trường ,các hàng gọi là các bản ghi. Cột với tên gọi và kiểu dữ liệu (kiểu dữ liệu của mỗi cột là duy nhất) xác định tạo nên cấu trúc của bảng. Khi bảng đã được tổ chức hệ thống cho một mục đích nào đó có một CSDL.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Đối tượng sử dụng website

Có 2 đối tượng sử dụng:

* Đối với khách hàng: Khách hàng có thể xem, tìm kiếm những chủng loại quần áo thời trang luôn được cập nhật và nhận được những chương trình khuyến mại trên Website. Ngoài ra khách hàng có thể mua hàng, gửi những ý kiến đóng góp cho shop, có thể đăng ký làm thành viên ( nếu muốn ).
* Đối với Admin : Quyền Admin quản lý các mặt hàng, nhóm hàng, hóa đơn, liên hệ

Ngoài ra trang Web sẽ hỗ trợ khả năng tìm kiếm hàng.

3.2 Chức năng chính của Website

3.2.1 Chức năng đối với khách hàng

a)Xem sản phẩm

|  |  |
| --- | --- |
| Mô tả | Khách hàng có thể xem chi tiết về các chủng loại quần áo thời trang |
| Nhập | Lựa chọn loại quần áo thời trang cần xem. |
| Xử lý | Tìm kiếm trong CSDL. |
| Xuất | Hiển thị thông tin về sản phẩm mà khách hàng muốn xem. |

Bảng 3-1: Bảng mô tả chức năng xem sản phẩm

***b) Tìm kiếm***

|  |  |
| --- | --- |
| Mô tả | Khách hàng có thể tìm kiếm sản phẩm hoặc 1 nhóm bằng những form tìm kiếm được xây dựng sẵn trên website. |
| Nhập | Khách hàng nhập vào từ khóa |
| Xử lý | Hệ thống sẽ ghi nhận và thực hiện tìm kiếm trong cơ sở dữ liệu tất cả sản phẩm mà tên có chứa từ khóa hay gần đúng với từ khóa |
| Xuất | Hiển thị thông tin về 1 hoặc nhiều loại quần áo thời trang thỏa mãn được tìm kiếm |

Bảng 3-2: Bảng mô tả chức năng tìm kiếm

***c) Lựa chọn sản phẩm vào giỏ hàng***

|  |  |
| --- | --- |
| Mô tả | Khách hàng có thể lựa chọn loại quần áo thời trang bất kỳ vào giỏ hàng. |
| Nhập | Lựa chọn quần áo thời trang cần mua. |
| Xử lý | -Kiểm tra sản phẩm có trong giỏ hàng.  -Nếu giỏ hàng đã có sản phẩm thì tăng số lượng trong giỏ hàng lên một đơn vị, nếu không thì thêm mới vào giỏ hàng |
| Xuất | Hiển thị thông tin về giỏ hàng mà khách hàng đã lựa chọn. |

Bảng 3-3: Bảng mô tả chức năng chọn sản phẩm vào giỏ hàng

***d) Mua hàng***

|  |  |
| --- | --- |
| Mô tả | Sau khi chọn sản phẩm vào giỏ hàng, khách hàng có thể mua hàng. |
| Nhập | Nhập sản phẩm vào giỏ hàng và cung cấp thông tin họ tên, địa chỉ, email, điện thoại, công ty, fax để đặt hàng. |
| Xử lý | -Kiểm tra sản phẩm có trong giỏ hàng.  -Kiểm tra thông tin khách hàng.  -Ghi nhận hóa đơn khách vào CSDL. |
| Xuất | Hiển thị thông tin về sản phẩm mà khách hàng đã đặt hàng. |

Bảng 3-4: Bảng mô tả chức năng mua hàng

***e) Liên hệ***

|  |  |
| --- | --- |
| Mô tả | Khách hàng có thể liên hệ với quản trị website. |
| Nhập | Nhập thông tin khách hàng và nội dung cần trao đổi. |
| Xử lý | -Kiểm tra tính hợp lệ của thông tin nhập.  -Thêm mới liên hệ vào CSDL. |
| Xuất | Hiển thị thông tin về giỏ hàng mà khách hàng đã lựa chọn. |

Bảng 3-5: Bảng mô tả chức năng liên hệ

***f) Đăng kí thành viên***

|  |  |
| --- | --- |
| Mô tả | Khách vãng lai có thể đăng ký làm thành viên thông qua một form được xây dựng sẵn. |
| Nhập | Đưa vào các thông tin cá nhân như user, password, địa chỉ email... |
| Xử lý | - Kiểm tra user không được trùng lặp  - Mật khẩu phải gồm ít nhất 6 ký tự.  - Email có đúng định dạng .  ...  - Thêm mới các thông tin khách hàng, mã hóa mật khẩu vào CSDL nếu đúng. Nếu không yêu cầu nhập lại. |
| Xuất | Thông báo tới khách hàng việc đăng ký đã thành công hoặc trở về trang đăng ký nếu có lỗi. |

Bảng 3-6: Bảng mô tả chức năng đăng kí thành viên

***g) Đăng nhập website***

|  |  |
| --- | --- |
| Mô tả | Đăng nhập vào website bằng cách gõ user và password đã đăng ký. |
| Nhập | Cung cấp tên đăng nhập và mật khẩu |
| Xử lý | Truy cập kiểm tra tính chính xác trong CSDL |
| Xuất | Hiển thị thông báo đăng nhập thành công hoặc lỗi và phải đăng nhập lại. |

Bảng 3-7: Bảng mô tả chức năng đăng nhập website

***h) Đăng xuất website***

|  |  |
| --- | --- |
| Mô tả | Đăng xuất khỏi website |
| Nhập | click vào nút đăng xuất trên website |
| Xử lý | - Hủy toàn bộ session của người dùng hiện tại  - Đưa trở về chức năng của khách vãng lai. |
| Xuất | Đưa về giao diện chính dành cho khách vãng lai |

Bảng 3-8: Bảng mô tả chức năng đăng xuất website

***i)Thay đổi mật khẩu***

|  |  |
| --- | --- |
| Mô tả | Khách hàng có thể thay đổi mật khẩu của mình. |
| Nhập | -Nhập vào mật khẩu cũ  -Nhập mật khẩu mới muốn đổi.  -Xác nhận mật khẩu mới muốn đổi. |
| Xử lý | -Kiểm tra tính chính xác của mật khẩu cũ.  -Nếu đúng thì mã hóa và cập nhậtmật khẩu mới của khách hàng vào CSDL. |
| Xuất | Hiển thị thông báo thay đổi mật khẩu thành công hoặc nếu lỗi thì yêu cầu nhập lại. |

Bảng 3-9: Bảng mô tả chức năng thay đổi mật khẩu

3.2.2 Chức năng của Admin

a)Đăng nhập quản trị

|  |  |
| --- | --- |
| Mô tả | Quản trị viên đăng nhập vào hệ thống bằng cách cung cấp tên đăng nhập và mật khẩu vào form đăng nhập được xây dựng sẵn. |
| Đầu vào | - Tên đăng nhập.  - Mật khẩu. |
| Xử lý | - Kiểm tra thông tin nhập vào có hợp lệ hay không.  - Kết nối đến CSDL để kiểm tra tên đăng nhập và mật khẩu. |
| Đầu ra | - Chuyển đến trang quản trị nếu đăng nhập thành công.  - Quay lại trang đăng nhập nếu đăng nhập thất bại. |

Bảng 3-10: Bảng mô tả chức năng đăng nhập quản trị

***b) Đăng xuất quản trị***

|  |  |
| --- | --- |
| Mô tả | Thoát khỏi tài khoản quản trị viên |
| Đầu vào | Click vào biểu tượng đăng xuất từ giao diện. |
| Xử lý | Tiến hành thoát tài khoản, hủy các biến session liên quan đến chức năng quản trị. |
| Đầu ra | Quay lại trang đăng nhập. |

Bảng 3-11:Bảng mô tả chức năng đăng xuất quản trị

***c) Thay đổi mật khẩu***

|  |  |
| --- | --- |
| Mô tả | Thay đổi mật khẩu admin. |
| Đầu vào | Nhập mật khẩu cũ và mật khẩu mới |
| Xử lý | -Kiểm tra mật khẩu cũ có trùng khớp.  - Kiểm tra tính hợp lệ của mật khẩu mới.  - Update pass mới vào CSDL. |
| Đầu ra | Thông báo cho admin quá trình đổi pass thành công hay không. |

Bảng 3-12:Bảng mô tả chức năng đổi mật khẩu cho Admin

***d) Quản lý các liên hệ***

|  |  |
| --- | --- |
| Mô tả | Quản trị viên được quyền xem và xóa các liên hệ của khách vãng lai và thành viên. |
| Nhập | Chọn chức năng xem hoặc xóa liên hệ |
| Xử lý | -Đối với việc xem:  Hiển thị ngày giờ, thông tin khách hàng, nội dung liên hệ.  - Đối với việc xóa:  Yêu cầu xác nhận xóa.  Xóa toàn bộ nội dung liên hệ. |
| Xuất | Hiện thông báo thành công hoặc lỗi. |

Bảng 3-13:Bảng mô tả chức năng quản lý liên hệ

***e)Quản lý sản phẩm***

|  |  |
| --- | --- |
| Mô tả | Quản trị viên có quyền thêm, sửa hoặc xóa sản phẩm. |
| Nhập | Chọn và sử dụng công cụ quản lý sản phẩm. |
| Xử lý | -Thêm:  Kiểm tra tính hợp lệ của thông tin sản phẩm mới  Kiểm tra có tồn tại hay chưa?  Nếu chưa thêm mới sản phẩm vào CSDL.  -Sửa:  Kiểm tra tính hợp lệ của thông tin sản phẩm mới  Kiểm tra có tồn tại hay chưa?  Nếu chưa cập nhật thông tin sản phẩm vào CSDL.  -Xóa:  Nếu sản phẩm đã thuộc một giỏ hàng hoặc đơn hàng thì không được phép xóa.  Ngược lại cho phép xóa sản phẩm. |
| Xuất | Hiện thông báo thành công hoặc lỗi. |

Bảng 3-14:Bảng mô tả chức năng quản lý sản phẩm

***f) Quản lý đơn hàng***

|  |  |
| --- | --- |
| Mô tả | Quản trị viên có thể xem, xóa và giải quyết đơn hàng. |
| Nhập | Chọn công cụ được thiết kế sẵn để xem, xóa và giải quyết đơn hàng. |
| Xử lý | -Xem:  Hiển thị tất cả các đơn hàng, phân theo 2 loại: chưa giải quyết và đã giải quyết.  -Xóa:  Chỉ được phép xóa các đơn hàng chưa giải quyết.  Các đơn hàng đã giải quyết thì không được xóa.  -Giải quyết đơn hàng  Đơn hàng chờ giải quyết được cập nhật thành đơn hàng đã giải quyết. |
| Xuất | Thông báo cập nhật thành công hoặc lỗi. |

Bảng 3-15: Bảng mô tả chức năng quản lý đơn hàng

3.3 Phân tích thiết kế hệ thống

3.3.1 Các yêu cầu chức năng chính của hệ thống

***a)Yêu cầu lưu trữ***

* Hệ thống cần lưu trữ các thông tin liên quan đến hàng hóa, thông tin đăng nhập, các đơn đặt hàng, các tin tức (tin khuyến mại và các loại tin khác được đăng trên website).
* Thông tin đăng nhập
* Phần quản trị của website cần đảm bảo tính bảo mật do có thể truy cập tới toàn bộ thông tin về hàng hóa và hoạt động kinh doanh.
* Thông tin hàng hóa
* Cần lưu trữ các thông tin sau đây về quần áo thời trang.
* Mã sản phẩm.
* Mã chuyên mục .
* Tên sản phẩm.
* Ảnh đại diện.
* Giá.
* Giá giảm.
* Số lượng.
* Trạng thái.
* Lượt xem.
* Đặc biệt
* Thông tin đơn hàng
* Cần lưu trữ các thông tin sau đây về đơn hàng:
* Mã hóa đơn.
* Trạng thái.
* Ngày tạo.
* Ngày duyệt.
* Tổng giá.
* Tên khách hàng.
* Giới tính.
* Địa chỉ .
* Email.
* .Số điện thoại.
* Địa chỉ giao hàng.
* Thời gian giao hàng.
* Tin tức
* Cần lưu trữ các thông tin sau về tin tức được đăng:
* Mã tin tức.
* Tiêu đề.
* Ảnh đại diện .
* Mô tả.
* Ngày tạo.
* Chi tiết.
* Lượt xem.

***b) Yêu cầu tính toán***

* Tính tiền tự động cho các sản phẩm mà khách hàng mua. Mỗi khi khách hàng mua một sản phẩm, sản phẩm này sẽ được lưu vào giỏ hàng để khách hàng dễ quản lý. Hệ thống tự động tính tổng giá trị giỏ hàng dựa trên việc lấy giá bán của từng sản phẩm lưu trong cơ sở dữ liệu và số lượng mỗi sản phẩm mà khách hàng mua, từ đó tính tổng tiền của tất cả các mặt hàng được mua để khách hàng theo dõi. Việc tính toán phải được thực hiện hoàn toàn tự động. Nếu khách hàng tiến hành đặt hàng, các kết quả tính toán này phải được lưu trong đơn đặt hàng.
* Kết quả tính toán: Tổng số tiền khách hàng phải trả cho mỗi lần mua.

***c)Yêu cầu tra cứu***

* Chọn thông tin tra cứu về sản phẩm.
* Kết quả xuất ra sau khi tra cứu: Danh sách các mặt hàng (nếu có) theo tiêu chuẩn đã được khách hàng đưa ra.

***d)Yêu cầu thống kê, báo cáo***

* Thống kê sách bán chạy.
* Thống kê hàng tồn kho.
* Báo cáo doanh thu.

3.3.2 Các biểu đồ Use Case

***a)Danh sách các Actor của hệ thống***

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Actor** | **Ý nghĩa** |
| 1 | Người quản lý | Admin là người quản trị trang web, có quyền hạn thực hiện tất cả các chức năng của hệ thống như quản lý hàng hóa, quản lý đơn hàng, tìm kiếm hàng hóa, cập nhật tin tức, đăng nhập, quản lý người dùng. |
| 2 | Khách hàng | Khách hàng lựa chọn hàng cần mua, yêu cầu mua hàng, xem tin tức, xem thông tin hàng hóa. |

Bảng 3-16: Bảng danh sách Actor

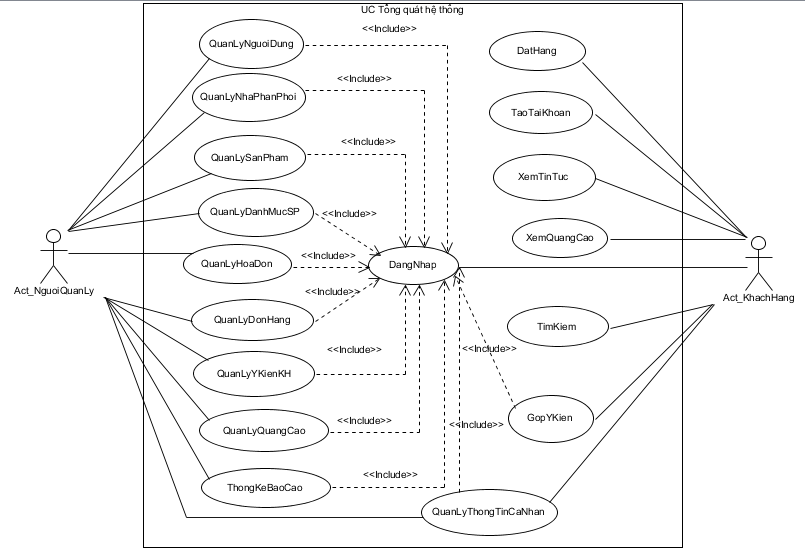
***b) Danh sách các Use Case của hệ thống***

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Usecase** | **Ý nghĩa** |
| 1 | Đăng nhập | Đăng nhập vào chức năng quản trị của trang web. |
| 2 | Quản lý đơn hàng | Quản lý đơn hàng (thêm, xóa) của website. |
| 3 | Quản lý sản phẩm | Quản lý các sản phẩm (thêm, xóa, sửa, tìm kiếm) của trang web. |
| 4 | Quản lý người dùng | Quản lý thông tin người dùng của trang web. |
| 5 | Quản lý danh mục sản phẩm | Thực hiện nhập, sửa, xóa, tìm kiếm thông tin loại sản phẩm. |
| 6 | Quản lý nhà phân phối | Thực hiện nhập, sửa, xóa, tìm kiếm thông tin nhà phân phối. |
| 7 | Quản lý hóa đơn | Quản lý các hóa đơn nhập, xuất của website. |
| 8 | Quản lý ý kiến khách hàng | Quản lý các bình luận, nhận xét về sản phẩm của khách hàng. |
| 9 | Quản lý thông tin cá nhân | Quản lý các thông tin tài khoản(sửa thông tin, đổi mật khẩu,…) trên website |
| 10 | Quản lý quảng cáo | Quản lý các quảng cáo (thêm, xóa, sửa) của trang web. |
| 11 | Đặt hàng | Xác nhận việc mua hàng đã chọn. |
| 12 | Tìm kiếm | Tìm kiếm sản phẩm của trang web. |
| 13 | Xem tin tức | Xem tin tức đăng trên trang web. |
| 14 | Xem quảng cáo | Xem các quảng cáo trên website. |
| 15 | Góp ý kiến | Trực tiếp nhận xét, bình luận về sản phẩm trên website của khách hàng. |
| 16 | Thống kê báo cáo | Thống kê dữ liệu trong website. |

Bảng 3- 17:Danh sách Use Case

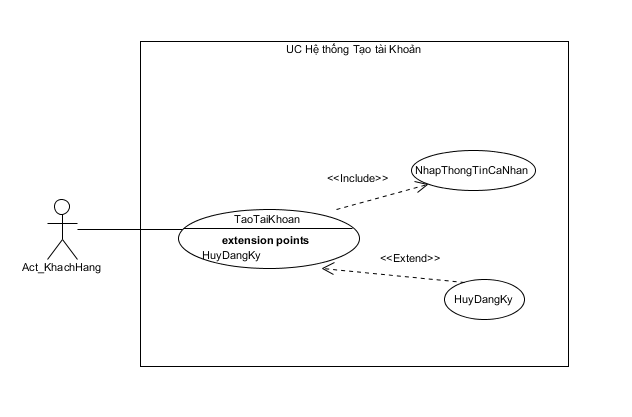
3.3.3 Đặc tả Use Case

***a)Use Case hệ thống***

******

###### Hình 3-1: Biểu đồ Use Case hệ thống

***b) Use Case đăng kí thành viên***

******

###### Hình 3-2: Biểu đồ Use Case đăng kí thành viên

* Tác nhân: Khách hàng.
* Mô tả: Cho phép khách xem đăng ký làm thành viên của hệ thống.
* Tiền điều kiện:
* Dòng sự kiện chính:
* Khách xem chọn mục đăng ký thành viên.
* Form Đăng ký thành viên hiển thị.
* Khách xem nhập thông tin cá nhân cần thiết vào form.
* Nhấn nút Đăng ký.
* Hệ thống thông báo kết quả nhập thông tin cá nhân. Nếu thông tin nhập không chính xác thì thực hiện luồng nhánh A1. Nếu nhập chính xác thì thực hiện bước 6.
* Hệ thống cập nhật thông tin của khách xem vào danh sách thành viên.
* Use case kết thúc.
* Dòng sự kiện rẽ nhánh:

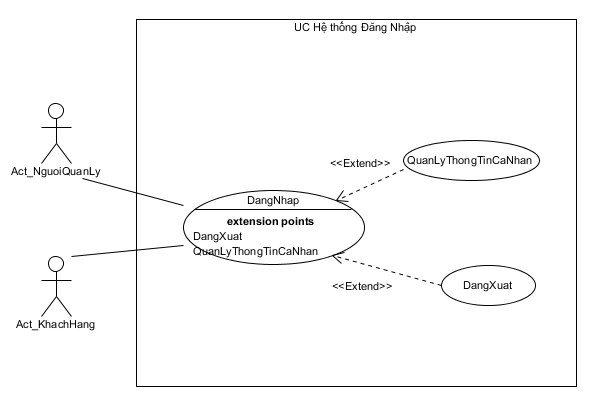
Dòng rẽ nhánh A1: Quá trình nhập thông tin không chính xác.

* Hệ thống thông báo nhập thông tin không chính xác.
* Hệ thống yêu cầu khách xem nhập thông tin lại.

Nếu khách xem đồng ý thì quay lại bước 2 của luồng sự kiện chính.Nếu không đồng ý thì use case kết thúc.

Hậu điều kiện: Khách hàng trở thành thành viên của hệ thống.

***c) Use Case đăng nhập***

******

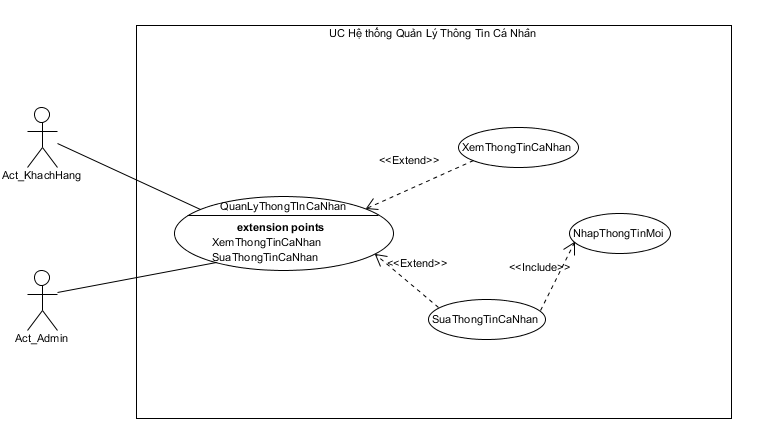
###### Hình 3-3: Biểu đồ Use Case đăng nhập

* Tác nhân: Thành viên.
* Mô tả: Use case cho phép thành viên đăng nhập vào hệ thống.
* Tiền điều kiện: Thành viên chưa đăng nhập vào hệ thống.
* Dòng sự kiện chính:
* Thành viên chọn chức năng đăng nhập.
* Form Đăng nhập hiển thị.
* Nhập tên, mật khẩu vào form Đăng nhập.
* Hệ thống kiểm tra tên, mật khẩu nhập của thành viên.
* Nếu thành viên nhập sai tên, mật khẩu thì chuyển sang dòng sự kiện rẽ nhánh A1. Nếu nhập đúng thì tiếp tục thực hiện các chức năng trong hệ thống.
* Use case kết thúc.
* Dòng sự kiện rẽ nhánh:

Dòng rẽ nhánh A1: Thành viên đăng nhập không thành công.

* Hệ thống thông báo quá trình đăng nhập không thành công.
* Chọn đăng ký hay nhập lại. Nếu chọn đăng ký thì thực hiện A2.
* Hệ thống yêu cầu thành viên nhập lại tên, mật khẩu.
* Nếu khách xem đồng ý thì quay về bước 2 của luồng sự kiện chính, nếu không đồng ý thì use case kết thúc.
* Hậu điều kiện: Thành viên đăng nhập thành công và có thể thực hiện các chức năng mà hệ thống cung cấp.

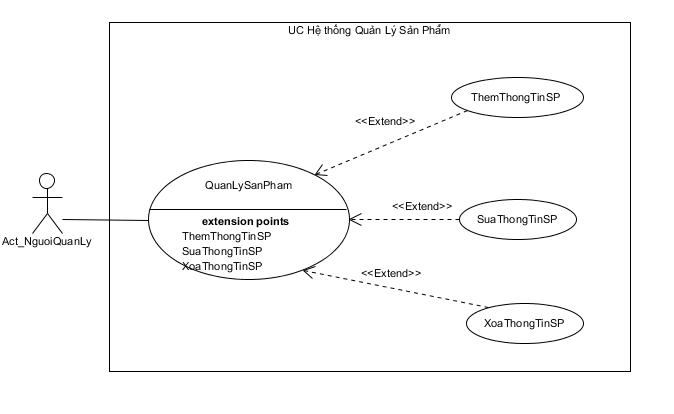
***d) Use Case quản lý thông tin cá nhân***

******

###### Hình 3-4: Biểu đồ Use Case quản lý thông tin cá nhân

* Tác nhân: Thành viên của hệ thống.
* Mô tả: Use case cho phép thành viên thay đổi các thông tin đã đăng ký.
* Tiền điều kiện: Đã đăng nhập vào hệ thống
* Dòng sự kiện chính:
* Thành viên chọn chức năng sửa thông tin cá nhân.
* Hệ thống hiển thị form sửa thông tin với các thông tin cũ của thành viên hiện tại.
* Thành viên nhập các thông tin mới.
* Nhấn nút lưu thông tin.
* Nếu cập nhật thành công thì thực hiện bước 6. Nếu không thì thực hiện luồng rẽ nhánh A1.
* Lưu thông tin.
* Use case kết thúc.
* Dòng sự kiện rẽ nhánh A1: Thông tin nhập không hợp lệ.
* Hệ thống thông báo việc nhập dữ liệu không hợp lệ.
* Thành viên nhập lại thông tin.
* Quay lại bước 4 của dòng sự kiện chính.
* Hậu điều kiện: Thông tin thay đổi của thành viên được lưu vào hệ thống.

***e) Use Case quản lý sản phẩm***

******

###### Hình 3-5: Biểu đồ Use Case quản lý sản phẩm

* Tác nhân: Người quản lý
* Mô tả: Use case cho phép thêm, thay đổi hay xóa danh sách của các sản phẩm trong hệ thống.
* Tiền điều kiện: Đã đăng nhập vào hệ thống.
* Dòng sự kiện chính:
* Người quản lý chọn kiểu tác động lên danh sách sản phẩm: thêm, thay đổi thông tin sản phẩm hay xóa sản phẩm ra khỏi danh sách sản phẩm của hệ thống.

Thêm sản phẩm:

* Hệ thống hiển thị form Nhập thông tin sản phẩm.
* Người quản lý nhập thông tin sản phẩm
* Nhấn nút lưu thông tin.
* Nếu nhập thành công thì thực hiện bước 5. Nếu sai thực hiện dòng sự kiện rẽ nhánh A1.
* Lưu thông tin sản phẩm.

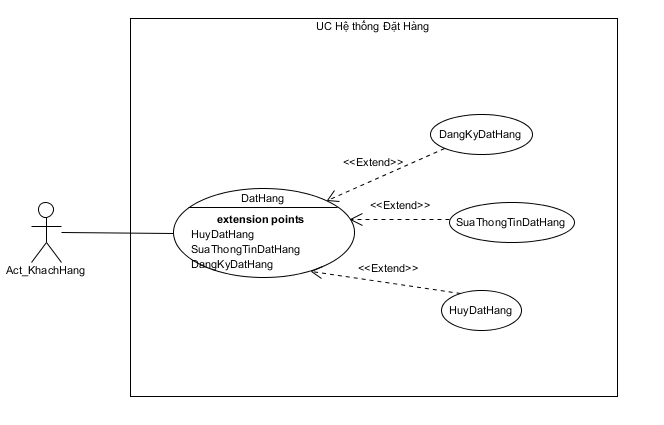
Thay đổi thông tin sản phẩm:

* Hệ thống hiển thị form sửa thông tin sản phẩm.
* Người quản lý nhập các thông tin cần thay đổi.
* Nhấn nút lưu thông tin.
* Nếu việc thay đổi thành công thì thực hiện bước 5. Nếu sai thực hiện dòng rẽ nhánh A1.
* Lưu thông tin về danh sách sản phẩm.

Xóa sản phẩm:

* Người quản lý chọn sản phẩm cần xóa.
* Nhấn nút xóa để thực hiện loại bỏ sản phẩm.
* Hệ thống hiển thị thông báo xác nhận việc loại bỏ. Nếu người quản lý đồng ý thì thực hiện bước 4. Nếu không đồng ý thì thực hiện bước 5.
* Thông báo sản phẩm đã được xóa.
* Hệ thống hiển thị lại danh sách sản phẩm.
* Use case kết thúc.
* Dòng sự kiện rẽ nhánh A1:
* Hệ thống thông báo việc nhập dữ liệu không hợp lệ.
* Người quản lý nhập lại thông tin.
* Quay lại bước 1.3 của dòng sự kiện chính.
* Hậu điều kiện: các thông tin về sản phẩm được cập nhật trong cơ sở dữ liệu.

***f) Use Case Đặt hàng***

******

###### Hình 3-6 : Biểu đồ Use Case đặt hàng

***Use case Đặt hàng***

* Tác nhân: Khách hàng.
* Mô tả: cho phép khách hàng đặt hàng sản phẩm.
* Tiền điều kiện: Sản phẩm cần chọn phải hiển thị trên hệ thống.
* Dòng sự kiện chính:
* Khách hàng chọn vào sản phẩm cần mua.
* Hệ thống hiển thị các sản phẩm có trong mục đã chọn.
* Khách hàng xem thông tin sản phẩm cần mua.
* Nếu hệ thống hiển thị có sản phẩm cần mua, khách hàng chọn vào sản phẩm cụ thể cần mua.

Đăng ký đặt hàng:

* Hệ thống hiển thị giao diện đặt hàng.
* Khách hàng điền thông tin của mình và các thông tin về sản phẩm cần mua (số lượng, màu sắc,…)
* Khách hàng chọn kiểu thanh toán.
* Hệ thống kiểm tra thông tin. Nếu đúng thông báo đặt hàng thành công. Nếu sai yêu cầu khách hàng kiểm tra lại thông tin.

Sửa thông tin đặt hàng:

* Hệ thống hiển thị những mặt hàng đã đặt của khách hàng.
* Khách hàng chọn mặt hàng cần thay đổi thông tin và nhập lại thông tin cần sửa.
* Hệ thống thông báo cập nhật thành công nếu khách hàng nhập đúng thông tin. Ngược lại thì thông báo cập nhật thất bại và yêu cầu khách hàng nhập lại.

Hủy đặt hàng:

* Khách hàng chọn đơn hàng cần hủy.
* Hệ thống yêu cầu xác nhận lại.
* Thông báo đã hủy đơn hàng.
* Nếu không thì thực hiện dòng sự kiện rẽ nhánh A1.
* Use case kết thúc.
* Dòng sự kiện phụ: A1
* Khách hàng chọn mua các sản phẩm ở mục khác.
* Quay lại bước 4 của luồng sự kiện chính.
* Hậu điều kiện: sản phẩm cần mua được chọn xong.

***Use case Thêm sản phẩm vào giỏ hàng***

* Tác nhân: Khách hàng.
* Mô tả: use case cho phép khách hàng đưa sản phẩm đã chọn vào giỏ hàng.
* Tiền điều kiện: sản phẩm đã được chọn.
* Dòng sự kiện chính:
* Chọn chức năng Thêm sản phẩm vào giỏ hàng để đưa sản phẩm đã chọn vào giỏ hàng. Nếu chưa chọn sản phẩm mà nhấn nút thêm vào giỏ hàng thì thực hiện dòng sự kiện rẽ nhánh A1.
* Hệ thống hiển thị thông tin đưa vào giỏ hàng.
* Use case kết thúc.
* Dòng sự kiện rẽ nhánh A1: chưa chọn sản phẩm cần mua
* Hệ thống thông báo sản phẩm chưa được chọn.
* Hiển thị sản phẩm để khách hàng chọn.
* Quay lại bước 1 của dòng sự kiện chính.
* Hậu điều kiện: Thông tin của sản phẩm đã đưa vào giỏ hàng phải được lưu trữ.

***Use case Loại sản phẩm đã chọn ở giỏ hàng***

* Tác nhân: Khách hàng.
* Mô tả: Use case cho phép khách hàng loại bỏ sản phẩm đã đưa vào giỏ hàng.
* Tiền điều kiện: Giỏ hàng đã có sản phẩm.
* Dòng sự kiện chính:
* Khách hàng chọn sản phẩm muốn loại bỏ khỏi giỏ hàng.
* Nhấn loại bỏ để thực hiện loại bỏ.
* Nếu có sản phẩm được chọn, hệ thống hiển thị thông báo để khách hàng xác nhận lại. Ngược lại thực hiện dòng sự kiện rẽ nhánh A1.
* Nếu chấp nhận thì sản phẩm được chọn sẽ bị loại bỏ. Nếu không thì thực hiện luồng sự kiện rẽ nhánh A2.
* Dòng sự kiện rẽ nhánh:

Dòng rẽ nhánh A1: Sản phẩm chưa được chọn:

* Hệ thống thông báo chưa được chọn sản phẩm cần loại bỏ.
* Quay lai bước 1 của luồng sự kiện chính.

Dòng rẽ nhánh A2: Quay lại giỏ hàng:

* Hệ thống hiển thị danh sách trong giỏ hàng.
* Use case kết thúc.
* Hậu điều kiện: sản phẩm được chọn bị loại khỏi giỏ hàng.

***Use case Thanh toán***

* Tác nhân: khách hàng
* Mô tả: use case cho phép khách hàng tạo đơn hàng để tiến hành thanh toán giỏ hàng sản phẩm.
* Tiền điều kiện: khách hàng đã chọn sản phẩm vào giỏ hàng.
* Luồng sự kiện chính:
* Khách hàng nhấn thanh toán để tạo đơn hàng.
* Form thanh toán xuất hiện, hệ thống hiển thị đơn hàng và một số thông tin mặc định của đơn hàng.
* Khách hàng nhập tiếp các thông tin: họ tên, số tài khoản, phương thức thanh toán, địa chỉ giao hàng, thời gian giao hàng, địa chỉ email để hoàn thành đơn hàng
* Chọn gửi đơn hàng. Nếu đơn hàng đã nhập đúng thì thực hiện bước 5. Nếu sai thì thực hiện luồng sự kiện rẽ nhánh A1
* Hệ thống kiểm tra số tài khoản của khách hàng và phương thức thanh toán. Nếu hợp lệ thì thực hiện bước 6. Nếu sai thì thực hiện luồng sự kiện rẽ nhánh A2
* Hệ thống thông báo giao dịch thành công.
* Use case kết thúc
* Luồng sự kiện rẽ nhánh

Luồng nhánh A1: đơn hàng nhập sai

* Hệ thống thông báo việc tạo đơn hàng không thành công
* Hệ thống hiển thị lại form nhập thông tin và các trường nhập sai
* Khách hàng nhập lại thông tin, nếu muốn ngưng giao dịch thì thực hiện luồng sự kiện rẽ nhánh A3.
* Quay lại bước 4 của luồng sự kiện chính

Luồng nhánh A2: tài khoản không hợp lệ

* Hệ thống thông báo phương thức thanh toán và tài khoản của khách hàng không hợp lệ.
* Khách hàng nhập lại thông tin nếu muốn tiếp tục mua hàng, nếu muốn ngưng giao dịch thì thực hiện luồng sự kiện rẽ nhánh A3.
* Quay lại bước 4 của luồng sự kiện chính.

Luồng nhánh A3: khách hàng hủy đơn hàng

* Khách hàng nhấn huỷ bỏ đơn hang.
* Hệ thống hiển thị các sản phẩm để khách hàng thực hiện giao dịch mới
* Use case kết thúc
* Hậu điều kiện: đơn hàng được gửi.

***g) Use Case nhà sản xuất***

******Hình 3-7: Biểu đồ Use Case nhà sản xuất

* Tác nhân: Người quản lý
* Mô tả: Use case cho phép thêm, thay đổi hay xóa danh sách hãng sản xuất trong hệ thống.
* Tiền điều kiện: Đã đăng nhập vào hệ thống.
* Dòng sự kiện chính:
* Người quản lý chọn kiểu tác động lên danh sách hãng sản xuất: thêm, thay đổi thông tin hãng sản xuất hay xóa hãng sản xuất ra khỏi danh sách hãng sản xuất của hệ thống.

***Thêm hãng sản xuất:***

* Hệ thống hiển thị form Nhập thông tin hãng sản xuất.
* Người quản lý nhập thông tin hãng sản xuất.
* Nhấn nút lưu thông tin.
* Nếu nhập thành công thì thực hiện bước cuối. Nếu sai thực hiện dòng sự kiện rẽ nhánh A1.
* Lưu thông tin hãng sản xuất.

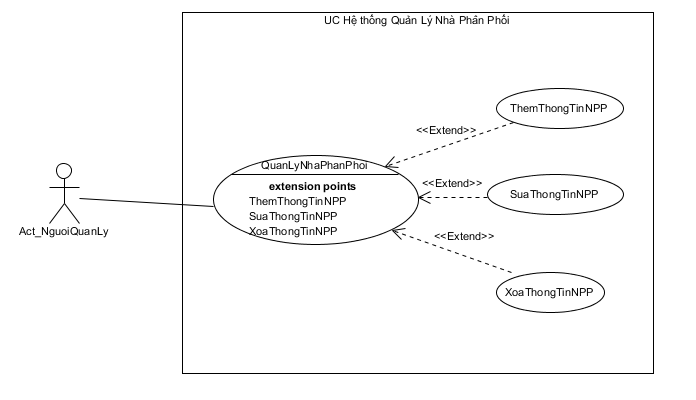
***Thay đổi thông tin hãng sản xuất:***

* Hệ thống hiển thị form sửa thông tin hãng sản xuất.
* Người quản lý nhập các thông tin cần thay đổi.
* Nhấn nút lưu thông tin.
* Nếu việc thay đổi thành công thì thực hiện bước cuối. Nếu sai thực hiện dòng rẽ nhánh A1.
* Lưu thông tin về danh sách hãng sản xuất.

***Xóa hãng sản xuất:***

* Người quản lý chọn hãng sản xuất cần xóa.
* Nhấn nút xóa để thực hiện loại bỏ hãng sản xuất.
* Hệ thống hiển thị thông báo xác nhận việc loại bỏ. Nếu người quản lý đồng ý thì thực hiện bước sau. Nếu không đồng ý thì thực hiện bước cuối.
* Thông báo hãng sản xuất đã được xóa.
* Hệ thống hiển thị lại danh sách hãng sản xuất.
* Use case kết thúc.
* Dòng sự kiện rẽ nhánh A1:
* Hệ thống thông báo việc nhập dữ liệu không hợp lệ.
* Người quản lý nhập lại thông tin.
* Quay lại bước 3 của dòng sự kiện chính.
* Hậu điều kiện: các thông tin về hãng sản xuất được cập nhật trong cơ sở dữ liệu.

***h) Use Case nhà phân phối***

******

Hình 3-8: Biểu đồ Use Case nhà phân phối

* Tác nhân: Người quản lý
* Mô tả: Use case cho phép thêm, thay đổi hay xóa danh sách nhà phân phối trong hệ thống.
* Tiền điều kiện: Đã đăng nhập vào hệ thống.
* Dòng sự kiện chính:
* Người quản lý chọn kiểu tác động lên danh sách nhà phân phối: thêm, thay đổi thông tin nhà phân phối hay xóa nhà phân phối ra khỏi danh sách nhà phân phối của hệ thống.

***Thêm nhà phân phối:***

* Hệ thống hiển thị form Nhập thông tin nhà phân phối.
* Người quản lý nhập thông tin nhà phân phối.
* Nhấn nút lưu thông tin.
* Nếu nhập thành công thì thực hiện bước cuối. Nếu sai thực hiện dòng sự kiện rẽ nhánh A1.
* Lưu thông tin nhà phân phối.

***Thay đổi thông tin nhà phân phối:***

* Hệ thống hiển thị form sửa thông tin nhà phân phối.
* Người quản lý nhập các thông tin cần thay đổi.
* Nhấn nút lưu thông tin.
* Nếu việc thay đổi thành công thì thực hiện bước cuối. Nếu sai thực hiện dòng rẽ nhánh A1.
* Lưu thông tin về danh sách nhà phân phối.

***Xóa nhà phân phối:***

* Người quản lý chọn nhà phân phối cần xóa.
* Nhấn nút xóa để thực hiện loại bỏ nhà phân phối.
* Hệ thống hiển thị thông báo xác nhận việc loại bỏ. Nếu người quản lý đồng ý thì thực hiện bước sau. Nếu không đồng ý thì thực hiện bước cuối.
* Thông báo nhà phân phối đã được xóa.
* Hệ thống hiển thị lại danh sách nhà phân phối.
* Use case kết thúc.
* Dòng sự kiện rẽ nhánh A1:
* Hệ thống thông báo việc nhập dữ liệu không hợp lệ.
* Người quản lý nhập lại thông tin.
* Quay lại bước 3 của dòng sự kiện chính.
* Hậu điều kiện: các thông tin về nhà phân phối được cập nhật trong cơ sở dữ liệu.

3.3.4 Các yêu cầu phi chức năng của hệ thống

***a)Các yêu cầu về hiệu năng***

Phần mềm có thể phản hồi ngay tới người dùng trong quá trình đang sử dụng thông qua các thông báo trợ giúp và có thể hướng dẫn sử dụng các chức năng khác.

***b) Các yêu cầu về sự logic của cơ sở dữ liệu***

* Cơ sở dữ liệu cần có sự logic thông qua việc liên kết các dữ liệu với nhau bằng cách : mỗi bảng dữ liệu trong cơ sở dữ liệu có khóa chính, các bảng này liên kết với nhau thông qua các khóa ngoại và các khóa ngoại này chính là dữ liệu được tham chiếu từ bảng bị tham chiếu.
* Bảng bị tham chiếu được tạo ra trước, có chứa khóa chính. Bảng tham chiếu sinh ra sau, nó cũng có khóa chính và có khi khóa chính của nó chính là khóa chính của bảng bị tham chiếu(trong trường hợp này dữ liệu đó vừa tham gia là khóa chính vừa tham gia là khóa ngoại). Dữ liệu bảng tham chiếu có chứa dữ liệu bảng bị tham chiếu.

***c) Các ràng buộc về thiết kế***

* Lập trình trên môi trường Web, hệ quản trị cơ sở dữ liệu SQL SerVer 2014.
* Công cụ phát triển sử dụng : Visual Studio 2015
* Các form được thiết kế phải có sự nhất quán, được sắp xếp hợp lý...

***d) Các đặc tính của hệ thống phần mềm***

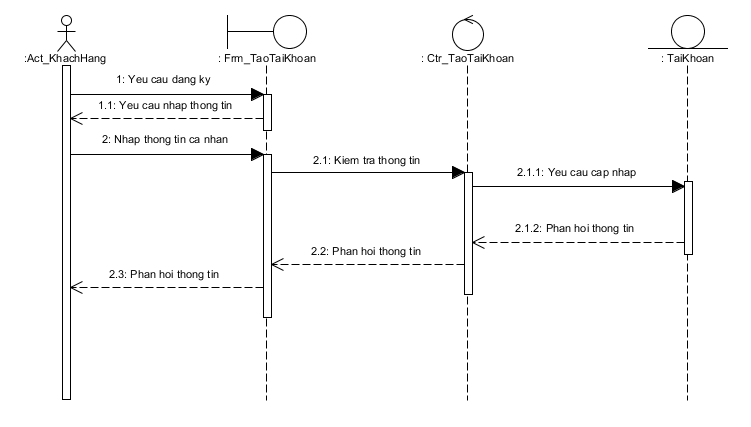
* Yêu cầu về độ tin cậy
* Hệ thống phải giúp người dùng quản lý được hệ thống thông tin một cách chính xác, đồng thời khi thực hiện một công việc cụ thể thì kết quả đưa ra phải tối ưu nhất.
* Yêu cầu về tính khả dụng
* Hệ thống có khả năng áp dụng vào thực tế cho các website quần áo thời trang vừa và nhỏ.
* Website có thể chạy trên môi trường sử dụng cơ sở dữ liệu SQL SerVer 2008, 2014 và Visual Studio 2015.
* Yêu cầu về tính bảo mật
* Website cần có bảo mật giúp người quản lý hệ thống quản lý được những thông tin về công việc quản lý của mình và mỗi người dùng trong hệ thống cũng có những thông tin cá nhân riêng cần bảo mật. Vì vậy, người quản lý hệ thống có tài khoản riêng và những thành viên có tài khoản riêng.
* Yêu cầu về bảo trì
* **Bảo trì** bao gồm điều chỉnh các lỗi mà chưa được phát hiện trong các giai đoạn trước của chu kỳ sống của hệ thống, nâng cấp tính năng sử dụng và an toàn vận hành của phần mềm.
* Yêu cầu về tính khả chuyển
* Website có thể được nâng cấp, cải tiến hơn về sau.
* Yêu cầu về tính khả dụng
* Giao diện website phải bắt mắt, trực quan và dễ sử dụng.
* Yêu cầu về tính ổn định
* Hệ thống phải hoạt động liên tục 24h/ngày, 7 ngày/tuần, vấn đề sự cố của hệ thống phải được hạn chế tối đa.
* Yêu cầu về hiệu suất
* Hệ thống phải cho phép thao tác và xuất kết quả nhanh chóng, truy vấn cơ sở dữ liệu và hiển thị kết quả nhanh chóng.

***e) Các giao tiếp***

* Font chữ : Times New Roman, cỡ chữ 10.
* Kích thước màn hình : tùy thuộc vào từng máy tính.

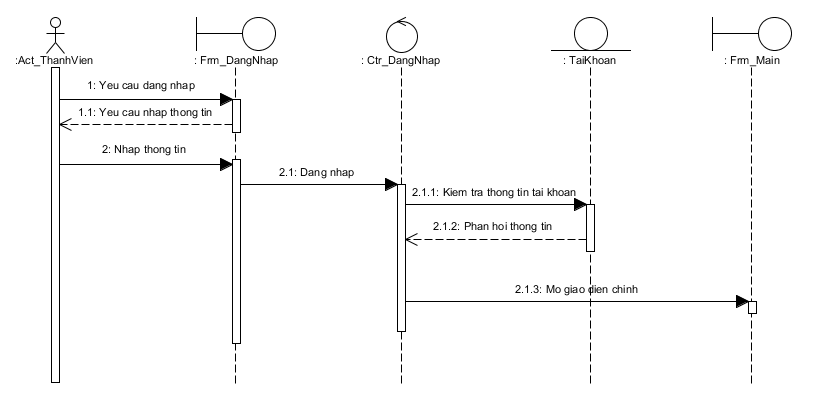
3.4 Biểu đồ tuần tự

3.4.1 Biểu đồ tuần tự Use case đăng kí thành viên



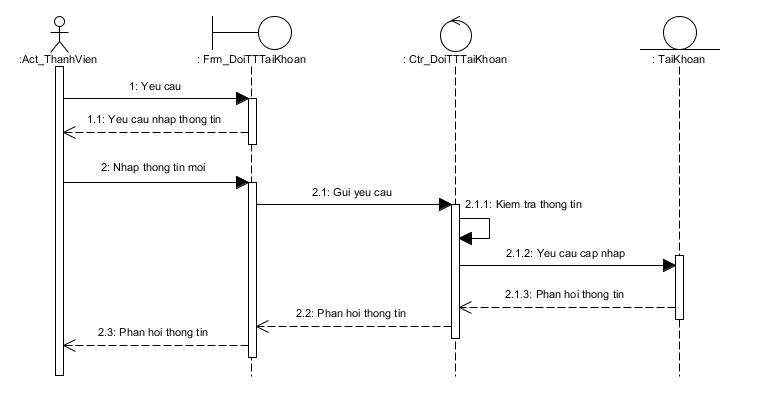
###### Hình 3-9: Biểu đồ tuần tự cho chức năng đăng ký thành viên

3.4.2 Biểu đồ tuần tự Use case đăng nhập



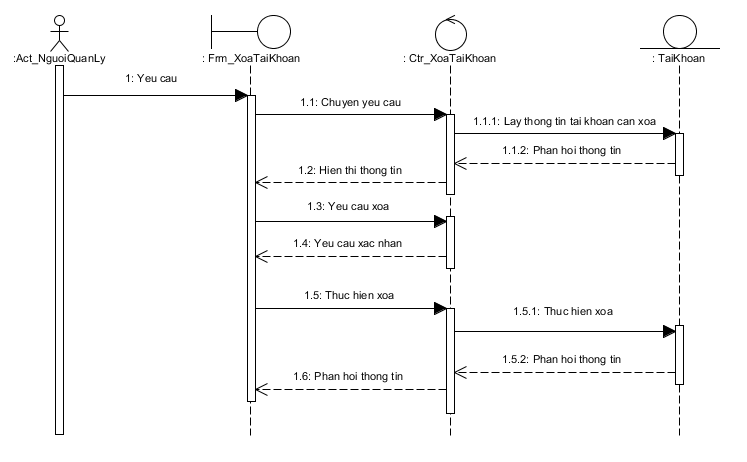
###### Hình 3-10: Biểu đồ tuần tự cho chức năng đăng nhập

3.4.3 Biểu đồ tuần tự Use case thay đổi thông tin tài khoản



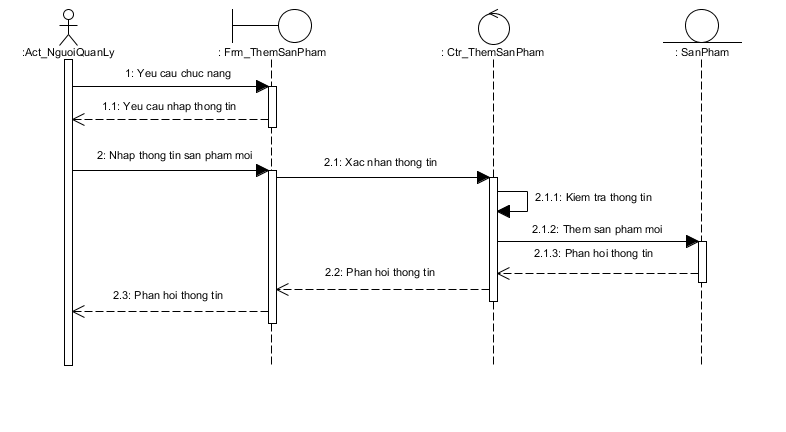
###### Hình 3-11: Biểu đồ tuần tự cho chức năng sửa thông tin tài khoản

3.4.4 Biểu đồ tuần tự Use case xóa tài khoản



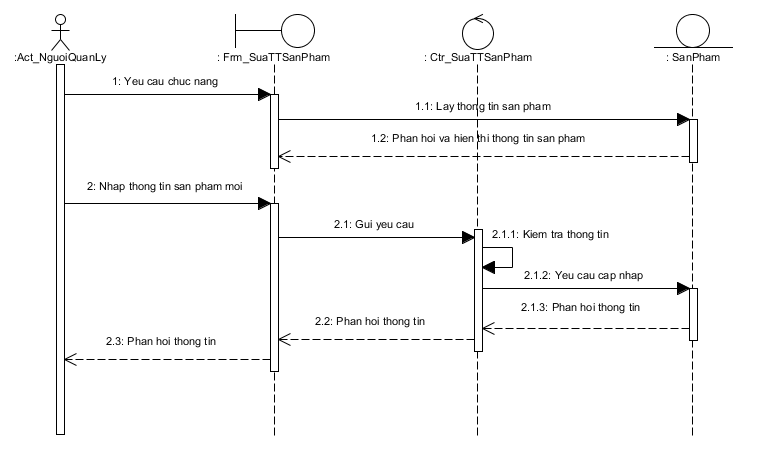
###### Hình 3-12: Biểu đồ tuần tự cho chức năng xóa tài khoản

3.4.5 Biểu đồ tuần tự Use case thêm sản phẩm



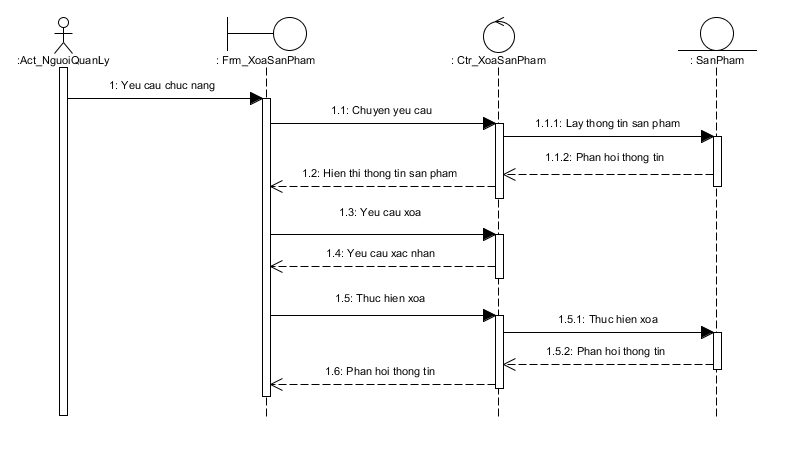
###### Hình 3-13: Biểu đồ tuần tự cho chức năng thêm sản phẩm

3.4.6 Biểu đồ tuần tự Use case thay đổi thông tin sản phẩm



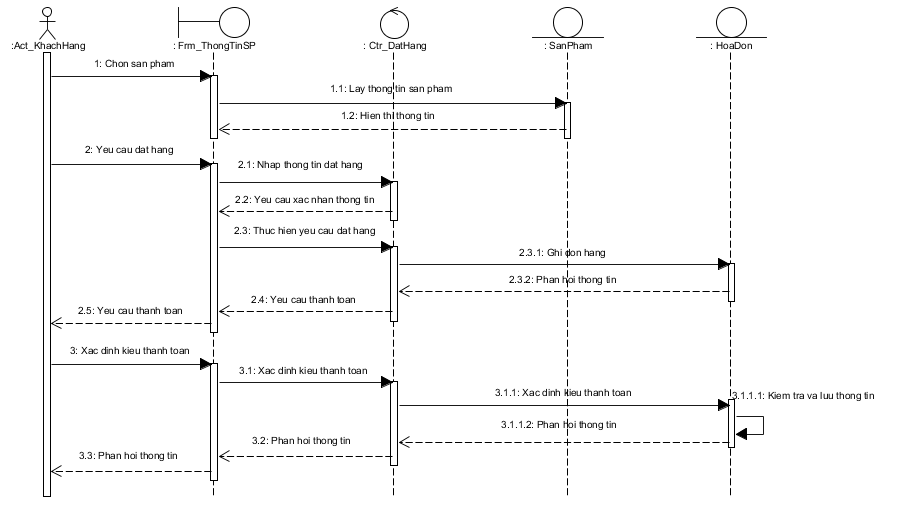
###### Hình 3-14: Biểu đồ tuần tự cho chức năng sửa thông tin sản phẩm

3.4.7 Biểu đồ tuần tự Use case xóa sản phẩm



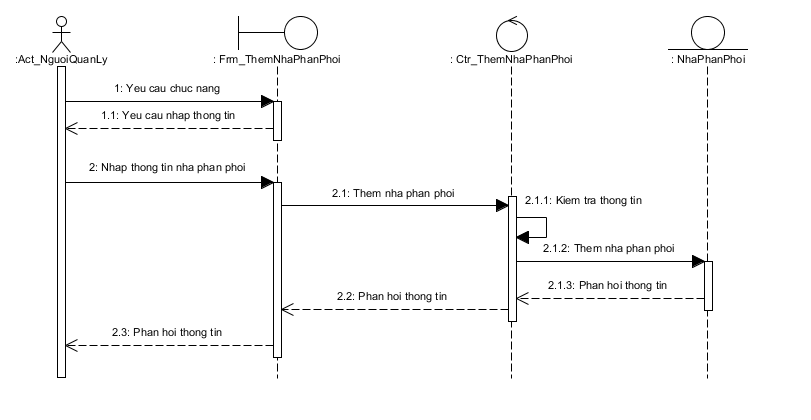
###### Hình 3-15: Biểu đồ tuần tự cho chức năng xóa sản phẩm

3.4.8 Biểu đồ tuần tự Use case đặt hàng



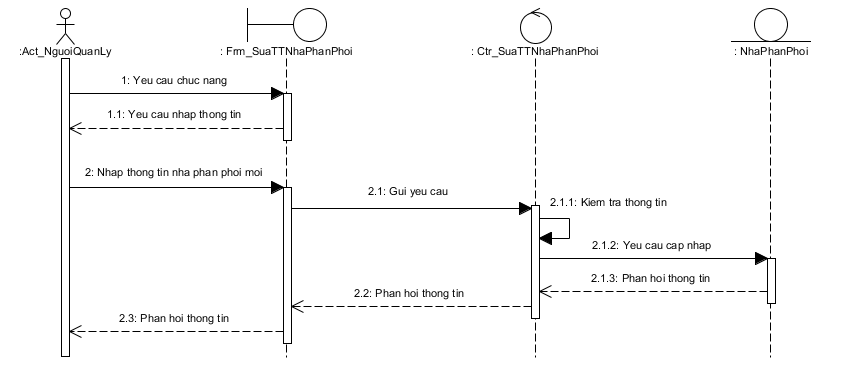
###### Hình 3-16: Biểu đồ tuần tự cho chức năng đặt hàng

3.4.9 Biểu đồ tuần tự Use case thêm nhà phân phối

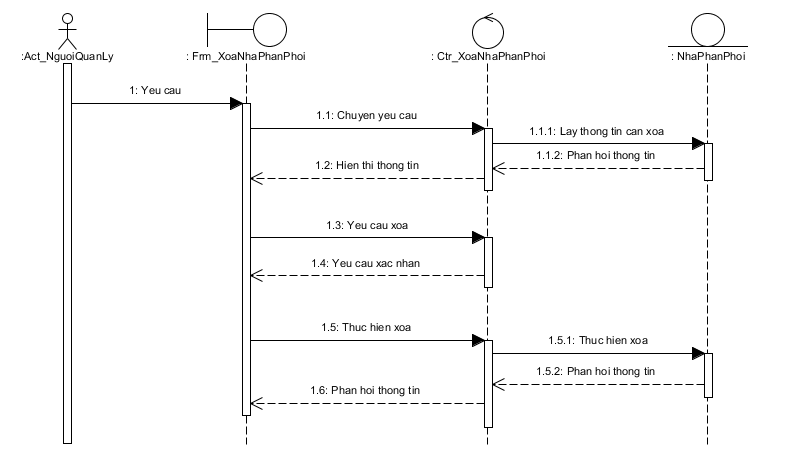


###### Hình 3-17: Biểu đồ tuần tự cho chức năng thêm nhà phân phối

3.4.10 Biểu đồ tuần tự Use case thay đổi thông tin nhà phân phối

***Hình 3-18: Biểu đồ tuần tự cho chức năng sửa thông tin nhà phân phối***

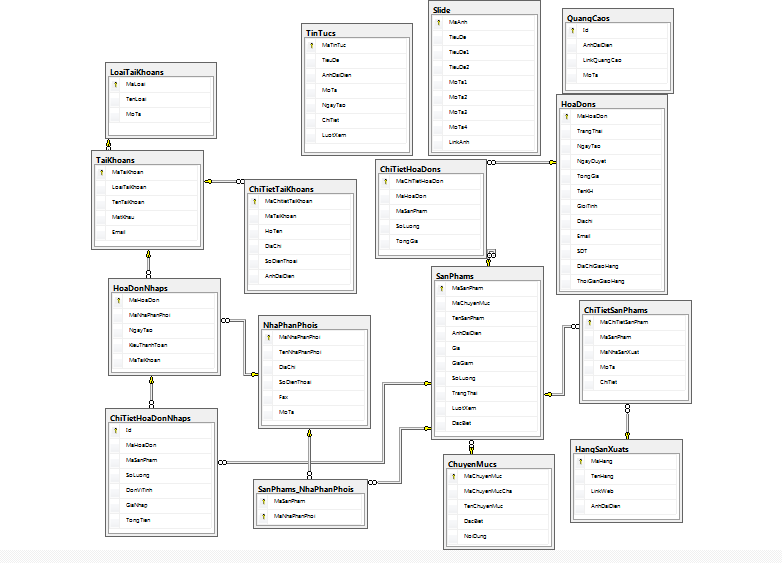
3.4.11 Biểu đồ tuần tự Use case xóa nhà phân phối



###### Hình 3-19: Biểu đồ tuần tự cho chức năng xóa nhà phân phối

3.5 Thiết kế cơ sở dữ liệu

3.5.1 Mô hình cơ sở dữ liệu



###### Hình 3-20: Biểu đồ dữ liệu quan hệ

3.5.2 Danh sách các lớp đối tượng

***a)Lớp khách hàng***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaKH | Varchar(10) | Khóa chính | Mã khách hàng |
| **2** | TenKH | Nvarchar(50) |  | Tên khách hàng |
| **3** | DiaChi | Nvarchar(50) |  | Địa chỉ |
| **4** | DienThoai | int |  | Điện thoại của khách hàng |
| **5** | TenDangNhap | Nvarchar(50) |  | Tên đăng nhập |
| **6** | MatKhau | Nvarchar(50) |  | Mật khẩu |

Bảng 3-18: Danh sách các thuộc tính Lớp Khách hàng

***b) Lớp hãng sản xuất***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaNSX | Varchar(10) | Khóa chính | Mã nhà sản xuất |
| **2** | TenNSX | Nvarchar(50) |  | Tên nhà sản xuất |
| **3** | linkweb | Nvarchar(200) |  | link website |
| **4** | AnhDaiDien | Nvarchar(200) |  | ảnh đại diện |

Bảng 3-19: Danh sách các thuộc tính Lớp Hãng sản xuất

c) Lớp nhà phân phối

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaNPP | Varchar(10) | Khóa chính | Mã nhà phân phối |
| **2** | TenNPP | Nvarchar(50) |  | Tên nhà phân phối |
| **3** | DiaChiNPP | Nvarchar(50) |  | Địa chỉ nhà phân phối |
| **4** | SoDienThoai | Int |  | Điện thoại nhà phân phối |
| **5** | Email | Nvarchar(50) |  | Email nhà phân phối |

Bảng 3-20: Danh sách các thuộc tính Lớp Nhà phân phối

***d) Lớp sản phẩm***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaSanPham | Varchar(10) | Khóa chính | Mã sản phẩm |
| **2** | TenSanPham | Nvarchar(50) | Không được rỗng. | Tên sản phẩm. |
| **3** | MaChuyenMuc | Varchar(10) | Khóa ngoại | Mã chuyên mục |
| **4** | AnhDaiDien | Nvarchar(200) |  | ảnh đại diện |
| **5** | Gia | Float | Không được rỗng | Đơn giá |
| **6** | GiaGiam | Float | Không được rỗng | Đơn giá |
| **7** | SoLuong | Int | Không được rỗng | Số lượng |
| **8** | TrangThai | Nvarchar(50) |  | Trạng Thái |
| **9** | LuotXem | int |  | lượt xem. |
| **10** | DacBiet | Int | Không được rỗng | Đặc biệt |

Bảng 3-21: Danh sách các thuộc tính Lớp Sản phẩm

***) Lớp hóa Nhập***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaHoaDon | Varchar(10) | Khóa chính | Mã hóa đơn xuất |
| **2** | MaNhaPhanPhoi | Varchar(10) | Khóa ngoại | Mã nhà phân phối |
| **3** | NgayTao | DateTime |  | Ngày Tạo |
| **4** | KieuThanhToan | Varchar(10) |  | Kiểu thanh toán |
| **5** | MaTaiKhoan | Nvarchar(20) | khóa ngoại | Mã tài khoản |

Bảng 3-22: Danh sách các thuộc tính Lớp Hóa đơn nhập

***g) Lớp chi tiết hóa đơn***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaChiTietHoaDon | Int | Khóa ngoại | Mã chi tiết hóa đơn |
| **2** | MaHoaDon | Varchar(10) | Khóa ngoại | Mã sản phẩm |
| **3** | MaSanPham | varchar(10) | Khóa Ngoại | Mã sản phẩm |
| **4** | SoLuong | int |  | số lượng |
| **5** | TongGia | Float |  | tổng giá |

Bảng 3-23: Danh sách các thuộc tính Lớp Chi tiết hóa đơn

***i)Lớp người dùng***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | UserName | Varchar(10) | Khóa chính | Tên đăng nhập |
| **2** | PassWord | Varchar(10) | Không được rỗng | Mật khẩu |
| **3** | HoTen | Nvarchar(50) |  | Họ tên người dùng |
| **4** | MaNhom | Varchar(10) | Khóa ngoại | Mã nhóm |
| **5** | SoDienThoai | Int |  | Số điện thoại |
| **6** | SoCMND | Int |  | Số chứng minh nhân dân |
| **7** | GioiTinh | Nvarchar (10) |  | Giới tính |
| **8** | DiaChi | Nvarchar(50) |  | Địa chỉ |
| **9** | Email | Nvarchar(50) |  | Email |
| **10** | NgaySinh | Datetime |  | Ngày sinh |

Bảng 3-24: Danh sách các thuộc tính Lớp Người dùng