|  |  |
| --- | --- |
|  |  |
| TẠ ĐỨC MẠNH | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| **ĐỒ ÁN TỐT NGHIỆP**  **NGÀNH CÔNG NGHỆ THÔNG TIN** |
|  |
| **NGHIÊN CỨU VÀ XÂY DỰNG WEBSITE BÁN ĐỒ TENNIS BẰNG ASP.NET MVC** |
|  |
|  |
| **CBHD :** **TS. Đặng Trọng Hợp** |
| CÔNG NGHỆ THÔNG TIN | **Sinh viên : Tạ Đức Mạnh** |
| **Mã sinh viên: 2020600026** |
|  |
|  |
|  |
| **Hà Nội - 2024** |
|  |

|  |  |
| --- | --- |
|  |  |
| TẠ ĐỨC MẠNH | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| **ĐỒ ÁN TỐT NGHIỆP**  **NGÀNH CÔNG NGHỆ THÔNG TIN** |
|  |
| **NGHIÊN CỨU VÀ XÂY DỰNG WEBSITE BÁN ĐỒ TENNIS BẰNG ASP.NET MVC** |
|  |
|  |
| **CBHD :** **TS.** **Đặng Trọng Hợp** |
| CÔNG NGHỆ THÔNG TIN | **Sinh viên : Tạ Đức Mạnh** |
| **Mã sinh viên: 2020600026** |
|  |
|  |
|  |
| **Hà Nội - 2024** |
|  |

# LỜI CẢM ƠN

Sau khoảng thời gian học tập tại trường và được tham gia các hoạt động phát triển phần mềm được sự truyền đạt kiến thức, lời khuyên trong công việc và giúp đỡ tận tình của quý thầy cô, em đã có cơ hội ra ngoài thực tập ở công ty, được tiếp xúc với môi trường làm việc thực tế, tiếp xúc với các công cụ quản lý dự án hiện đại,… Em xin chân thành cảm ơn các thầy cô, giảng viên trường Đại học Công Nghiệp Hà Nội nói chung và khoa Công nghệ thông tin nói riêng - những người đã ân cần giảng dạy, truyền đạt kiến thức và hướng dẫn em trong quá trình học tập tại trường.

Bên cạnh đó, em cũng xin chân thành cảm ơn sự hướng dẫn và chỉ bảo tận tình của thầy Đặng Trọng Hợp – người đã luôn quan tâm, chỉ bảo và tạo điều kiện cho em trong suốt quá trình thực tập tốt nghiệp và quá trình thực hiện đề tài. Các định hướng và sự hiểu biết về khoa học của thầy chính là tiền đề để em có thể hoàn thành được đề tài này một cách tốt nhất.

Tuy còn một số hạn chế nhất định về chuyên môn và thời gian nên bài báo cáo và ứng dụng còn nhiều hạn chế. Kính mong quý thầy cô góp ý để đề tài của em hoàn thiện hơn.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Tạ Đức Mạnh

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc166530397)

[MỤC LỤC 2](#_Toc166530398)

[MỤC LỤC HÌNH ẢNH 5](#_Toc166530399)

[Chương 1. CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 6](#_Toc166530400)

[1.1. Giới thiệu chung về hệ thống 6](#_Toc166530401)

[1.2. Đặt vấn đề 7](#_Toc166530402)

[1.3. Phạm vi đề tài 7](#_Toc166530403)

[1.4. Phương pháp nghiên cứu/Hướng giải quyết đề tài 8](#_Toc166530404)

[**CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG** 11](#_Toc166530405)

[2.1. Hệ quản trị cơ sở dữ liệu SQL Server 11](#_Toc166530407)

[2.2. Tổng quan về ASP.NET MVC 13](#_Toc166530408)

[2.3. Tổng quan về Web API 16](#_Toc166530409)

[2.4. Tổng quan kiến trúc của Entity Framework 20](#_Toc166530410)

[2.5. Tổng quan về Entity Framework Code First 22](#_Toc166530411)

[**2.6. HTML, CSS VÀ JAVASCRIPT** 24](#_Toc166530412)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 26](#_Toc166530413)

[3.1. Các bước hoàn thành dự án 26](#_Toc166530414)

[*1.1* 3.2. Thu thập yêu cầu 27](#_Toc166530415)

[3.1.1. Yêu cầu doanh nghiệp 27](#_Toc166530419)

[3.1.2. Yêu cầu chức năng 28](#_Toc166530420)

[3.1.3. Yêu cầu phi chức năng 29](#_Toc166530423)

[3.1.4. Yêu cầu hệ thống 30](#_Toc166530424)

[3.3. Danh sách chức năng 30](#_Toc166530429)

[3.3.1. Phân hệ Khách (Guest) 30](#_Toc166530430)

[3.3.2. Phân hệ thành viên 31](#_Toc166530431)

[3.3.3. Phân hệ Administrator (Quản trị viên) 32](#_Toc166530432)

[3.4. Phân tích thiết kế hệ thống 33](#_Toc166530433)

[3.4.1. Danh sách các Usecase của hệ thống 33](#_Toc166530437)

[3.4.2. Sơ đồ use case 34](#_Toc166530438)

[3.4.3. Mô tả chi tiết một số use case 40](#_Toc166530439)

[3.4.4. Sơ đồ tuần tự 46](#_Toc166530440)

[3.5. Thiết kế cơ sở dữ liệu 50](#_Toc166530442)

[3.5.1. Cơ sở dữ liệu và các mối quan hệ 50](#_Toc166530443)

[3.5.2. Mô tả chi tiết các bảng trong cơ sở dữ liệu 50](#_Toc166530444)

[**CHƯƠNG 4: KẾT QUẢ CÀI ĐẶT CHƯƠNG TRÌNH** 54](#_Toc166530445)

[4.1 Giao diện chính trang hiển thị 54](#_Toc166530449)

[4.2 Giao diện trang đăng nhập 54](#_Toc166530450)

[4.3 Giao diện trang đăng kí 55](#_Toc166530451)

[4.4 Giao diện trang chi tiết sản phẩm 55](#_Toc166530452)

[4.5 Giao diện trang giỏ hàng 56](#_Toc166530453)

[4.6 Giao diện trang đặt hàng và thanh toán 56](#_Toc166530454)

[4.7 Giao diện trang quản trị 57](#_Toc166530455)

[4.8 Giao diện trang quản lý sản phẩm 57](#_Toc166530456)

[4.9 Giao diện trang thêm sản phẩm 58](#_Toc166530457)

[4.10 Giao diện trang quản lý bài viết 58](#_Toc166530458)

[4.11 Giao diện trang quản lý tài khoản 59](#_Toc166530459)

[4.12 Giao diên cài đặt slide show 59](#_Toc166530460)

[**4.19. Kiểm thử** 59](#_Toc166530461)

[**4.19.1. Kiểm thử hộp đen** 59](#_Toc166530462)

[**4.19.2.** **Kiểm thử tương thích** 61](#_Toc166530463)

[**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 63](#_Toc166530464)

[Kết luận 63](#_Toc166530467)

[**Hướng phát triển** 63](#_Toc166530468)

[**TÀI LIỆU THAM KHẢO** 65](#_Toc166530469)

# MỤC LỤC HÌNH ẢNH

**No table of figures entries found.**

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

## Giới thiệu chung về hệ thống

* Hệ thống bán sản phẩm bán đồ tennis qua mạng chạy trên nền web giúp công việc trao đổi hàng hóa được tiến hành một cách nhanh chóng, tiện lợi và ít tốn kém nhất. Cung cấp những dịch vụ và giải pháp quảng cáo bán sản phẩm bán đồ tennishàng đầu trên internet. Hệ thống Bán sản phẩm bán đồ tennis ra đời nhằm đáp ứng mọi nhu cầu mua bán trao đổi những sản phẩm, mặt hàng và cả dịch vụ cho những tổ chức công ty và cá nhân có nhu cầu.
* Thông tin sẽ được truyền tải tới người sử dụng một cách nhanh chóng và hiệu quả nhất giúp người dùng dễ dàng tìm kiếm thông tin sản phẩm mình cần và giúp quảng bá sản phẩm tới tay người tiêu dùng một cách rộng rãi nhất.
* Website cho phép mọi người đăng kí làm thành viên và đăng tải tin bán sản phẩm bán đồ tennis của mình. Tin bán sản phẩm bán đồ tennis có thể là mua, bán, trao đổi hoặc xem các thông tin về công nghệ … Website sẽ giúp người dùng phát tán tin bán sản phẩm bán đồ tennislên các website khác và giúp quảng bá tin bán sản phẩm bán đồ tennis tới tất cả mọi người ghé thăm website. Tin bán sản phẩm bán đồ tenniscủa người dùng sẽ có cơ hội rất cao để được một số lượng người dùng lớn đang sử dụng các dịch vụ mua bán trực tuyến biết đến và theo dõi.
* Hệ thống website còn được quản lý chuyện nghiệp cùng với hệ thống đánh giá bài viết, đánh giá khách hàng giúp tạo niềm tin vào các thông tin bán sản phẩm bán đồ tennis trên internet.
* Hệ thống gồm 3 phân hệ: Phân hệ khách, phân hệ thành viên (người dùng) thông thường, và phân hệ Administrator.

## Đặt vấn đề

* Để quản lý quy trình mua hàng của khách nhân viên kỹ thuật phải ghi chép các biểu mẫu hằng ngày và lưu trữ dưới dạng file thô, khả năng lưu trữ kém và có thể sai sót và gây nhiều bất lợi cho nhân viên kỹ thuật về việc quản lý mua hàng của khách hàng. Ngoài ra khi các bộ phận, nhà máy đặt hàng với số lượng lớn và các mặt hàng khác nhau thì việc giải quyết các nhu cầu mua hàng đó vô cùng phức tạp và mất quá nhiều thời gian nhưng hiệu quả công việc vẫn còn rất thấp.
* Vấn đề đặt ra cho bài toán là làm thế nào xây dựng một website cho phép nhân viên kỹ thuật có thể quản lý quy trình mua hàng của từng khách riêng biệt nhờ vậy mà công ty dễ dàng quản lý được các đơn đặt hàng, thống kê được doanh số bán hàng và lợi nhuận của từng tháng.
* Bên cạnh đó đề tài cần đảm bảo được các yêu cầu sau:
* Tính tiện dụng: Hệ thống phải có tính thân thiện với người dùng, đảm bảo khả năng hỗ trợ tối đa cho người dùng trong việc sử dụng và khai thác hệ thống.
* Tính bảo mật: Hệ thống phải đảm bảo khả năng bảo mật cho thông tin trong quá trình truy xuất thông tin từ phía người dùng. Mọi truy xuất của người dùng phải được kiểm soát cả từ phía trình khách lẫn trình chủ nhầm đảm bảo khả năng vận hành tốt nhất cho hệ thống.
* Tính tùy biến: Hệ thống phải hỗ trợ khả năng tùy biến nhằm giúp cho người quản trị dễ dàng kiểm soát các nguồn thông tin của mình tùy thuộc vào nhu cầu của họ.

## Phạm vi đề tài

Đề tài nghiên cứu xoay quanh vấn đề quản lý quy trình mua hàng của khách hàng và hỗ trợ cho nhân viên kỹ thuật để giúp công ty quản lý tốt quy trình mua hàng của của khách hàng được hiệu quả hơn.

* Hệ thống được xây dựng trên nền website và người dùng là khách hàng có nhu cầu mua hàng trực tuyến trên hệ thống website của công ty. Hệ thống sẽ gồm các chức năng chính như sau:
* Quản trị viên (admin) quản lý sản phẩm, đơn hàng, thành viên người dùng
* Khách hàng xem, tìm kiếm và mua sản phẩm
* Khách hàng có thể đăng nhập và xem danh sách đơn hàng đã mua
* Khách hàng có thể gửi liên hệ và xem thông tin công ty
* Có 2 view cho khách hàng và cho quản trị, quản trị yêu cầu đăng nhập còn khách hàng thì mặc định là không.

## Phương pháp nghiên cứu/Hướng giải quyết đề tài

* Cấu trúc dự án
* Dự án được xây dựng theo sự kết hợp của mô hình 3 lớp và MVC.
* Mục đích là để có thể sử dụng lại được toàn bộ code ở phần server side vì AngularJS cũng là MVC.
* Một số khái niệm:
* Backend: Toàn bộ code c# xử lý WebAPI, EF, MVC5
* Frontend: Phần code giao diện bên ngoài như AngularJS, Bootstrap,HTML, CSS, JavaScript
* Trang admin: Trang dành cho Admin
* Trang customer: Trang dành cho khách hàng Database -> Repository -> Service -> Web API, Controller-> ASP.NET MVC
* Yêu cầu cài đặt:
* Microsoft SQL Server 2022:

https://www.microsoft.com/en-us/download/details.aspx?id=30438

* Visual Studio 2022 Community:

https://www.visualstudio.com/downloads/download-visual-studio-vs

* Node Package Manage: https://nodejs.org/en/
* Bower: https://bower.io/
* Công nghệ sử dụng
* Truy xuất dữ liệu: Entity Framework
* Resfull API: Web API
* Hiển thị cho khách hàng: ASP.NET MVC thuần
* Phần quản trị:
* Phần chứng thực: ASP.NET Identity
* Định hướng công nghệ:
* Có thể sử dụng ADO.NET nếu cần tốc độ nhưng lại vất vả trong khi codevà cần có kỹ năng quản lý code tốt.
* Có thể dùng Dapper hỗ trợ bạn quản lý kết nối và mapping data.
* Ngoài ra có thể dùng các thư viện khác.
* Tất cả các thư viện này đều phải base trên ADO.NET Provider. Nên ADO.NET tốc độ vẫn là vô địch vì nó nằm gần SQL nhất.
* Dựng cấu trúc solution cho dự án
* Các thành phần của Solution
* Demo.Common: Chứa các lớp tiện ích dùng chung cho dự án
* Demo.Model: Chứa các lớp Domain Entities của dự án
* Shop.Data: Chứa tầng truy cập dữ liệu sủ dụng Entity Framework
* Codefirst
* Demo.Service: Chứa các service xử lý Business logic
* Demo.Web: Project chính dùng để hiển thị giao diện và tương tác
* người dùng.
* Demo.UnitTest: Chứa các class Text sử dụng cho việc Unit Test
* Các package cần cài đặt
* Shop.Common
* Shop.Model
* Entity Framework
* Microsoft.AspNet.Identity.Core
* Microsoft.AspNet.Identity.EntityFramework
* Shop.Data
* Entity Framework
* Shop.Service
* Shop.Web
* Autofac, AutoMapper, Entity Framework...
* Shop.UnitTest
* Moq, Entity Framwork

**CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG**



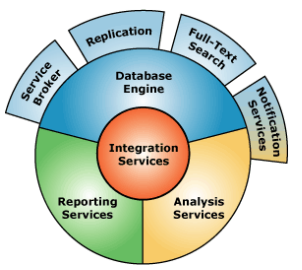
## 2.1. Hệ quản trị cơ sở dữ liệu SQL Server

****

*Hình 2.1: Hệ quản trị CSDL SQL Server*

* Giới Thiệu SQL Server
* SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (Transact-SQL) để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. 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 có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server....
* Các thành cơ bản trong SQL Server

SQL Server được cấu tạo bởi nhiều thành phần như Database Engine, Reporting Services, Notification Services, Integration Services, Full-Text Search Service.... Các thành phần này khi phối hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc lưu trữ và phân tích dữ liệu một cách dễ dàng.

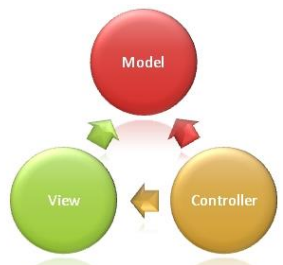


*Hình 2.2: Các thành phần cơ bản trong SQL Server*

* Database Engine: Đây là mộtengine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh ví dụ: trả lại tài nguyên cho ệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.
* Integration Services: là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu. Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Oracle, SQL Server, DB2, Microsoft Access,...và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.
* Analysis Services: Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Dữ liệu khi được lưu trữ vào trong database mà bạn không thể lấy được những thông tin bổ ích thì coi như không có ý nghĩa gì. Chính vì thế, công cụ này ra đời giúp bạn trong việc phân tích dữ liệu một cách hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimendion cubes.
* Notification Services: Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo dịch thời đến hàng ngàn người đăng ký sử dụng trên nhiều loại thiết bị khác nhau.
* Reporting Services: là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.
* Full Text Search Service: là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các cơ sở dữ liệu SQL Server.
* Service Broker: là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

## 2.2. Tổng quan về 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.
* Nền tảng MVC bao gồm các thành phần dưới đây:



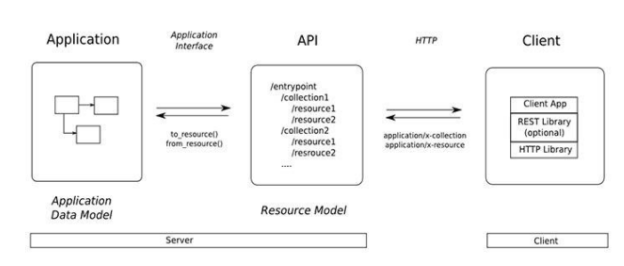
*Hình 2.3: Mô hình ASP.NET MVC*

* Models: Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.
* Views: Views là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các 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.
* 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 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
* Lợi ích của ứng dụng được xây dựng trên nền tảng Web Forms
* Nó hỗ trợ cách lập trình hướng sự kiện, quản lý trạng thái trên giao thức HTTP, tiện dụng cho việc phát triển các ứng dụng Web phục vụ kinh doanh. Các ứng dụng trên nền tảng Web Forms cung cấp hàng tá các sự kiện được hỗ trợ bởi hàng trăm các server controls.
* Sử dụng mẫu Page Controller. Xem thêm ở mục Page Controller trên MSDN
* Mô hình này sử dụng view state hoặc server-based form, nhờ đó sẽ giúp cho việc quản lý trạng thái các trang web dễ dàng.
* Nó rất phù hợp với các nhóm lập trình viên quy mô nhỏ và các thiết kế, những người muốn tận dụng các thành phần giúp xây dựng ứng dụng một cách nhanh chóng.
* Nói tóm lại, áp dụng Web Forms giúp giảm bớt sự phức tạp trong xây dựng ứng dụng, bởi vì các thành phần (lớp Page, controls,...) được tích hợp chặc chẽ và thường thì giúp bạn viết ít code hơn là áp dụng theo mô hình MVC.

## 2.3. Tổng quan về Web API

API là viết tắt của Application Programming Interface (giao diện lập trình ứng dụng) phương thức kết nối với các thư viện và ứng dụng khác. Windows có nhiều API, và Twitter cũng có web API, tuy nhiên chúng thực hiện các chức năng khác nhau, với mục tiêu khác nhau. Nó chính là một phần mềm giao tiếp được sử dụng bởi các ứng dụng khác nhau. Nó cũng giống như bàn phím là thiết bị dùng để giao tiếp giữa người sử dụng và máy tính, API là một phần mềm giao tiếp giữa chương trình và hệ điều hành.

* Những điểm nổi bật của API:
* Đây là một trong những framework mới sẽ giúp ít cho bạn trong việc xây dựng các HTTP service một cách rất đơn giản và nhanh chóng.
* Mã nguồn mở nên bạn có thể được sử dụng bởi bất kì một client nào hỗ trợ XML, JSON.
* Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma.
* Bạn có thể sử dụng các host nằm trong phần ứng dụng hoặc trên IIS.
* Một kiểu kiến trúc vô cùng phù hợp dành cho các thiết bị trang bị băng thông giới hạn như smartphone, tablet.
* Thường nó có định dạng dữ liệu là JSON, XML hoặc một kiểu dữ liệu bất kỳ.
* REST For Web Developers
* Hiện nguyên lý REST và cấu trúc dữ liệu RESTful được biết đến rất rộng rãi trong giới lập trình, nhưng vẫn có nhiều người hay nhầm lẫn. Bản thân REST không phải là một loại công nghệ, mà là phương thức tạo API với nguyên lý tổ chức nhất định. Những nguyên lý này nhằm hướng dẫn lập trình viên tạo môi trường xử lý API request được toàn diện hơn.
* REST là viết tắt của Representational State Transfer. Giải thích đơn giản, REST là một loạt hướng dẫn và dạng cấu trúc dùng cho việc chuyển đổi dữ liệu. Thông thường, REST hay được dùng cho ứng dụng web, nhưng cũng có thể làm việc được với dữ liệu phần mềm.
* Tóm lại, RESTful API là các API theo cấu trúc REST.
* Đặc điểm
* Sự nhất quán trong cả API
* Tồn tại không trang thái (ví dụ, không có server-side session)
* Sử dụng HTTP status code khi cần thiết
* Sử dụng URL endpoint với logical hierarchy



*Hình 2.4: Cấu trúc REST*

* Versioning trong URL chứ không phải trong HTTP header
* REST là một phương thức nhỏ gọn. Nên rất được ưa chuộng cho dữ liệu HTTP. Cũng vì vậy nên REST dần phổ biến trên web, và được xem là lựa chọn “số một” cho phát triển API.
* Có 4 lệnh dùng để truy cập RESTful API:
* GET để truy vấn object
* POST để tạo object mới
* PUT để sửa đổi hoặc thay thế một object
* DELETE để loại bỏ một object
* Mỗi phương thức trên phải được API call thông qua để gửi chỉ thị cho server phải làm gì.
* Sự khác nhau giữa Web Service, WCF, WCF REST, Web API

Hiện nay trong kỷ nguyên hậu PC, smartphone và tablet lên ngôi, nhu cầu xây dựng các ứng dụng mobile trên Windows Phone, iOS, Android đồng bộ dữ liệu đến server theo mô hình điện toán đám mây đang trở nên rất hot. Nếu quyết định xây dựng phía server sử dụng .NET Framework (với phiên bản mới nhất 4.5.3) thì lập trình viên có thể tạo ra các HTTP service bằng rất nhiều công nghệ khác nhau như là: Web Service (ASMX), WCF Service, WCF REST Service,

* Web API Service
* Web Service
* Đây là công nghệ cũ nhất của .NET Framework
* Nó dựa trên SOAP (Simple Object Access protocol) dữ liệu trả về dạng XML
* Chỉ hỗ trợ giao thức HTTP
* Không phải Open Source nhưng có thể sử dụng được với bất cứ client
* nào hỗ trợ XML
* Chỉ có thể host trên IIS

Ưu điểm:

* Code và Test đơn giản

Nhược điểm:

* Chỉ hỗ trợ giao thức SOAP để truyền nhận dữ liệu nên performance
* không cao
* Không thể tạo ra service dạng REST hỗ trợ định dạng dữ liệu JSON
* WCF (.NET 3.0 trở lên)
* Là bản nâng cấp đáng giá của WCF với việc trên .NET 3.5 Microsoft
* bổ sung webHttpBinding để hỗ trợ RESTful service
* Hỗ trợ 2 HTTP verb GET, POST để truyền nhận dữ liệu với 2 thuộc
* tính tương ứng là WebGet và WebInvoke
* Muốn sử dụng các HTTP verb khác như PUT, DELETE cần cấu hình
* thêm trên IIS
* Hỗ trợ các định dạng dữ liệu XML, ATOM, JSON

Ưu điểm:

* Bổ sung hỗ trợ RESTful service với định dạng dữ liệu JSON nhẹ hơn SOAP với dữ liệu XML rất nhiều
* Cho phép cấu hình tham số WebGet qua URI sử dụng UriTemplate

Nhược điểm:

* Chưa hoàn toàn phải là RESTful service, mới chỉ hỗ trợ mặc định GET, POST
* Cấu hình khó nhớ (cố hữu của WCF)
* Web API (.NET 4 trở lên)
* Đây là một framework mới giúp cho việc xây dựng các HTTP service rất đơn giản và nhanh chóng
* Open Source và có thể được sử dụng bởi bất kì client nào hỗ trợ XML, JSON
* Hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content formats
* Có thể host trong ứng dụng hoặc trên IIS
* Kiến trúc lý tưởng cho các thiết bị có băng thông giới hạn như smartphone, tablet
* Định dạng dữ liệu có thể là JSON, XML hoặc một kiểu dữ liệu bất kỳ

Ưu điểm:

* Cấu hình hết sức đơn giản khi so với WCF
* Performance cao
* Hỗ trợ RESTful đầy đủ
* Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test
* Open Source

Nhược điểm:

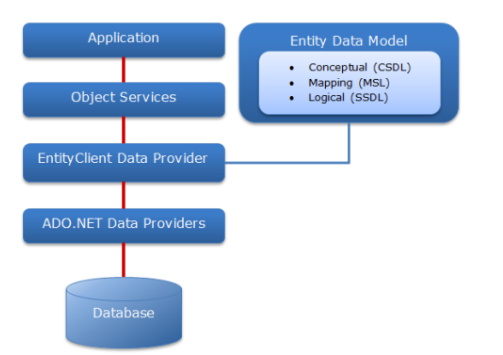
* Còn rất mới nên chưa có nhiều đánh giá về nhược điểm của Web API

- Tại sao dùng Web API:

* Service RESTful hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content formats
* Service cung cấp dữ liệu cho nhiều client khác nhau với băng thông giới hạn như: browser, mobile, tablet...

## 2.4. Tổng quan kiến trúc của Entity Framework

ADO.NET Entity Framework là một nền tảng được sử dụng để làm việc với database thông qua cơ chế ánh xạ Object/Relational Mapping (ORM). Nhờ đó, bạn có thể truy vấn, thao tác với database gián tiếp thông qua các đối tượng lập trình.



*Hình 2.5: Kiến trúc của Entity Framework*

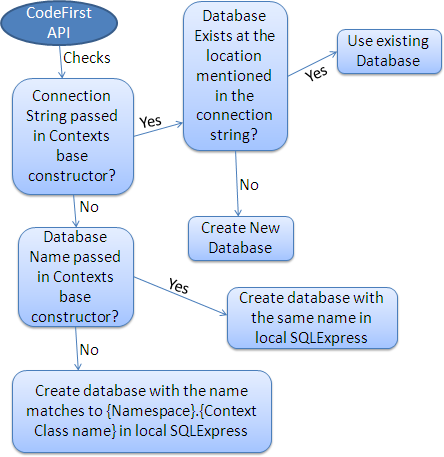
* Kiến trúc của Entity Framework được minh họa như sau:
* Object Services

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<TEntity> 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<TEntity> và EntityReference<TEntity>: 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.

## 2.5. Tổng quan về Entity Framework Code First

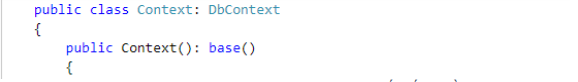
* Entity Framework Code First được giới thiệu từ Entity Framework 4.1. Code First chủ yếu là hữu ích trong Domain Driven Design (DDD). Trong cách tiếp cận Code First, bạn có thể tập trung vào việc thiết kế Domain và bắt đầu tạo ra các lớp theo yêu cầu của Domain của bạn chứ không phải thiết kế cơ sở dữ liệu trước rồi sau đó tạo ra các lớp phù hợp với thiết kế cơ sở dữ liệu đó. Code First API sẽ tạo ra cơ sở dữ liệu dựa trên các lớp thực thể và lớp cấu hình của bạn.
* Vì vậy, đầu tiên bạn bắt đầu viết các lớp thay vì tập trung vào thiết kế cơ sở dữ liệu, sau đó khi bạn chạy ứng dụng, Code First API sẽ tạo ra cơ sở dữ liệu mới hoặc ánh xạ các lớp của bạn vào cơ sở dữ liệu đã tồn tại trước khi chạy ứng dụng của bạn.
* Vì vậy, các công việc cơ bản sẽ là: Viết các lớp → Nhấn F5 để chạy ứng dụng → Code First API tạo ra cơ sở dữ liệu mới hoặc ánh xạ các lớp với cơ sở dữ liệu đã tồn tại → Thêm dữ liệu mặc định vào cơ sở dữ liệu hoặc kiểm tra dữ liệu trong cơ sở dữ liệu → Cuối cùng khởi chạy ứng dụng.
* Hình dưới đây thể hiện luồng khởi tạo database dựa trên đối số truyền vào phương thức khởi tạo của lớp context kế thừa từ lớp DbContext:



*Hình 2.6: Luồng khởi tạo database*

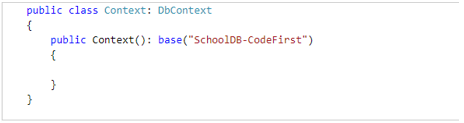
Có thể truyền các đối số vào phương thức khởi tạo của lớp context như sau:

* Không có đối số truyền vào: Nếu bạn không truyền đối số cho phương thức khởi tạo thì Code First API sẽ tạo database của bạn trong SQLEXPRESS với tên được đặt theo quy tắc sau {Namespace}.{Tên\_lớp\_Context}. Ví dụ: database sẽ có tên là SchoolDataLayer.Context nếu chúng ta định nghĩa lớp Context như sau:

**

*Hình 2.7: Định nghĩa lớp Context không có đối số truyền vào*

* Truyền vào tên của database: Nếu bạn truyền tên của database vào đối số của phương thức khởi tạo, Code First API sẽ tạo database có tên như vậy trong SQLEXPRESS. Ví dụ: lớp Context dưới đây sẽ tạo database tên là "SchoolDB-CodeFirst" trong SQLEXPRESS.



*Hình 2.8: Định nghĩa lớp Context truyền vào tên của database*

**2.6. HTML, CSS VÀ JAVASCRIPT**

* **HTML (Hypertext Markup Language)**

HTML là chữ viết tắt của cụm từ HyperText Markup Language (dịch là Ngôn ngữ đánh dấu siêu văn bản) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML. Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

* **CSS (Cascading Style Sheets)**

CSS là từ viết tắt của Cascading Style Sheets: công dụng dùng để trang trí trang web và thường được gắn với các ngôn ngữ như HTML, PHP dùng làm nổi bật trang web và hình ảnh trang web. Trong Style Sheet này chứa những câu lệnh CSS. Mỗi câu lệnh CSS sẽ định dạng một phần nhất định của HTML, ví dụ như: font của chữ, đường viền, màu nền, căn chỉnh hình ảnh,…

* **JAVASCRIPT**
* JavaScript là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu được tạo ra vào năm 1995 bởi Brendan Eich tại Netscape. Mục đích ban đầu để nâng cao khả năng tương tác của trang web với người sử dụng.
* JavaScript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome,... thậm chí các trình duyệt trên thiết bị di động cũng hỗ trợ.
* JavaScript là một ngôn ngữ lập trình dự trên nguyên mẫu với cú pháp phát triển từ C. JavaScript có khái niệm từ khóa, do đó gần như JavaScript không được mở rộng. Cũng giống như C, JavaScript không có bộ xử lý xuất/nhập riêng. Trong khi C sử dụng thư viện xuất/nhập chuẩn, JavaScript dựa vào phần mềm ngôn ngữ được gắn vào để thực hiện xuất/nhập.
* Trên trình duyệt, rất nhiều trang web sử dụng JavaScript để thiết kế trang web động và một số hiệu ứng hình ảnh thông qua DOM. JavaScript được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ HTML như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh,... Ở Việt Nam, JavaScript còn được ứng dụng để làm bộ gõ tiếng Việt giống như bộ gõ hiện đang sử dụng trên trang Wikipedia tiếng Việt. Tuy nhiên, mỗi trình duyệt áp dụng JavaScript khác nhau và không tuân theo chuẩn W3C DOM, do đó trong rất nhiều trường hợp lập trình viên phải viết nhiều phiên bản của cùng một đoạn mã nguồn để có thể hoạt động trên nhiều trình duyệt. Một số công nghệ nổi bật dòng JavaScript để tương tác với DOM bao gồm DHTML, Ajax và SPA.

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 3.1. Các bước hoàn thành dự án

*Bảng 3-1: Các bước hoàn thành dự án*

|  |  |
| --- | --- |
| **Tiến trình** | **Mô tả** |
| Khởi tạo dự án | Tìm hiểu các tài liệu liên quan đến dự án.  Tìm hiểu cách thức hoạt động của việc bán hàng và giao  hàng trên mạng.  Xem xét tính khả thi của dự án, các khía cạnh liên quan  đến dự án. |
| Lập kế hoạch | Mô tả tổng quát quá trình thực hiện dự án.  Ràng buộc dự án với các điều kiện chủ quan và khách  quan. Xác định tính khả thi của dự án.  Xác định rủi ro và các vấn đề phát sinh trong quá trình  xây dựng dự án có thể gặp.  Lập kế hoạch hệ thống mới. |
| Thu thập yêu cầu | Thu thập yêu cầu của khách hàng đối với hệ thống một  cách chi tiết và chính xác.  Khảo sát, tìm hiểu các hệ thống đã có thể đưa ra các yêu  cầu, các chức năng cho hệ thống của mình. |
| Phân tích và kiểm tra  yêu cầu | Phân tích các yêu cầu chức năng và phi chức năng.  Kiểm tra tính khả thi của yêu cầu.  Bổ sung các yêu cầu còn thiếu sót. |
| Phân tích | Mô hình hóa các yêu cầu hệ thống và đặc tả các chức  năng. |
| Thiết kế | Thiết kế cơ sở dữ liệu. Thiết kế giao diện. |
| Kiểm thử | Kiểm thử cho từng giai đoạn của dự án từ lúc bắt đầu đến lúc kết thúc dự án. |
| Cài đặt | Xây dựng website cho kế hoạch đã đặt ra.  Viết tài liệu hướng dẫn sử dụng hệ thống người dùng. |
| Thực thi | Đưa hệ thống vào hoạt động thử. Theo dõi hoạt động của hệ thống. |

## 3.2. Thu thập yêu cầu



### Yêu cầu doanh nghiệp

* Hệ thống quản lý mua bán bán sản phẩm bán đồ tennis trực tuyến của khách hàng tại website:
* Khi hách hàng có nhu cầu mua hàng thì khách hàng sẽ trực tiếp truy cập vào website để tham khảo mặt hàng, giá cả của tất cả các loại sản phẩm mà website đang kinh doanh đã được đưa lên trang web. Thông tin mỗi sản phẩm gồm: tên sản phẩm, đơn giá, hình ảnh chi tiết, thông tin chi tiết. Sau khi lựa chọn được sản phẩm muốn mua thì khách hàng chọn thêm vào giỏ hàng. Bạn không cần phải có tài khoản đăng nhập vào hệ thống vẫn có thể đặt mua hàng được. Khách hàng kiểm tra lại giỏ hàng một lần nữa, tại đây khách hàng có thể hủy bỏ giỏ hàng nếu không muốn đặt hàng hoặc chỉnh sửa lại số lượng, sản phẩm trong giỏ hàng. Tiếp theo, khách hàng điền đầy đủ các thông tin: họ tên, số sản phẩm bán đồ tennis, email, địa chỉ giao hàng.
* Hệ thống sẽ tự động gửi mail thông báo cho khách hàng biết đã đặt hàng thành công và thông tin chi tiết về đơn đặt hàng đến địa chỉ email mà khách hàng đã cung cấp.
* Nhân viên sau khi nhận được đơn đặt hàng sẽ chủ động liên lạc đến khách hàng thông qua email hoặc số trà sữađể xác nhận lại lần cuối. Nếu khách hàng đồng ý, nhân viên sẽ duyệt đơn hàng, chuẩn bị hàng giao cho khách, ngược lại, nếu khách hàng không muốn đặt hàng thì nhân viên sẽ hủy đơn đặt hàng.
* Sau khi đơn đặt hàng được duyệt, số lượng sản phẩm trong kho sẽ giảm tương ứng với số lượng sản phẩm trong đơn đặt hàng.
* Các chức năng của quản trị viên:
* Quản trị viên có quyền hạn cao nhất, thực hiện được tất cả thao tác trênweb. Trang quản trị sẽ hiển thị những thông tin theo sự phân quyền sử dụng khác nhau được quản lý bởi thông tin sau: tên đăng nhập, mật khẩu.
* Quản lý danh mục sản phẩm: xem danh sách danh mục sản phẩm, thêm, sửa, xóa danh mục.
* Quản lý sản phẩm: xem danh sách sản phẩm, thêm, sửa, xóa sản phẩm.
* Quản lý thành viên: xem danh sách thành viên, thêm, sửa, xóa thành viên (chỉ admin, shop manager mới có quyền thực hiện).
* Quản lý đơn đặt hàng: xem, xác nhận đơn đặt hàng, xóa đơn đặt hàng
* Quản lý bình luận: xem, xóa bình luận.

### Yêu cầu chức năng

* Đối với người dùng:

*Bảng 3-2: Mô tả yêu cầu chức năng đối với người dùng*

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Loại chức năng** |
| 1 | Xem thông tin sản phẩm | Truy cập |
| 2 | Tìm kiếm sản phẩm | Tìm kiếm |
| 3 | Xem những thông tin trên website | Truy cập |
| 4 | Đặt hàng | Giao dịch mua bán |
| 5 | Bình luận, liên hệ | Lưu trữ, hỏi đáp |

* Đối với ban quản trị:

*Bảng 3-3: Mô tả yêu cầu chức năng đối với quản trị*

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Loại chức năng** |
| 1 | Quản lý danh mục | Thêm, xóa, sửa, tìm kiếm danh mục |
| 2 | Quản lý sản phẩm | Thêm, xóa, sửa, tìm kiếm sản phẩm |
| 3 | Quản lý thành viên | Thêm, xóa, sửa, thay đổi quyền của thành viên. |
| 4 | Quản lý đơn đặt hàng | Xem, duyệt, xóa đơn đặt hàng |
| 5 | Quản lý bình luận | Xem danh sách, xóa |
| 6 | Đăng nhập, đăng xuất | Truy cập, quản trị hệ thống |
| 7 | Thống kê – báo cáo | Thống kê doanh số bán hàng và lợi nhuận của  công ty qua từng tháng |



### Yêu cầu phi chức năng

*Bảng 3-3: Mô tả yêu cầu phi chức năng*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Nội dung** | **Hiệu quả** | **Mức độ chi tiết** |
| 1 | Cho phép người dùng truy cập | Tốt | Website cho phép nhiều người truy cập không bị treo hay xung đột |
| 2 | Giao diện đẹp mắt, dễ nhìn, thân thiện với người dùng | Tiện lợi | Người dùng và người quản trị hệ thống không cần có trình độ tin học cao vẫn có thể sử dụng hiệu quả |
| 3 | Tốc độ xử lý công việc nhanh chóng, chính xác | Hiệu quả | Xem, tìm kiếm, đăng ký, mua hàng,… |
| 4 | Tính bảo mật | Tốt | Thông tin cá nhân, tài khoản, mật khẩu, phân quyền thành viên |

### Yêu cầu hệ thống

*Bảng 3-4: Mô tả yêu của hệ thống*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Chức năng** | **Đơn đặt hàng** | **Liên hệ** | **Quản lý thông tin thành viên** |
| 1 | Thêm | Quản trị, khách hàng | Khách hàng | Quản trị |
| 2 | Xóa | Quản trị, khách hàng | Khách hàng | Quản trị |
| 3 | Sửa | Quản trị, khách hàng | Khách hàng | Quản trị |
| 4 | Xem | Quản trị, khách hàng | Khách hàng | Quản trị |



## Danh sách chức năng

### Phân hệ Khách (Guest)

Khách (Guest) ở đây là một khách hàng, người dùng đang xem các thông tin trang web (không nhất thiết phải có tài khoản trong hệ thống).

Danh sách chức năng

Phân hệ này bao gồm các chức năng sau:

* Xem danh sách tất cả các mục bán sản phẩm bán đồ tennis chính :
  + Tất cả
  + Sản phẩm mới
  + Sản phẩm nổi bật
  + …
* Xem danh sách các mục bán sản phẩm bán đồ tennis con trong 1 mục bán sản phẩm bán đồ tennis chính
* Tìm kiếm cơ bản bài bán sản phẩm bán đồ tennis theo từ khóa bất kì :
* Tìm kiếm nâng cao theo từ khóa với các tiêu chí :
  + Địa điểm
  + Thời gian
  + Gía tiền
  + ….
* Đăng kí tài khoản thành viên thông thường
* Xem thông tin liên hệ người dùng khác
* Đăng kí Email nhận tin bán sản phẩm bán đồ tennis mới theo từng mục hay tất cả.
* Liệt kê các bài bán sản phẩm bán đồ tennis theo các nội dung ( một mục hay nhiều mục bán sản phẩm bán đồ tennis):
  + Các bài mới nhất
  + Các bài đáng tin cậy nhất ( đánh giá tốt nhất )
  + Các tài khoản đáng tin nhất
  + Các bài trả lời nhiều nhất
  + Các bài xem nhiều nhất
  + …
* Xem nội dung bài bán sản phẩm bán đồ tennis bất kì bao gồm các thông tin:
  + Tiêu đề , ngày giờ đăng, số người xem, số người trả lời,đánh giá, tác giả…
  + Nội dung bài bán sản phẩm bán đồ tennis ( bao gồm text + hình ảnh + link )
  + Nội dung các bài trả lời
  + …
    1. Phân hệ thành viên

Giới thiệu

Thành viên thường là người dùng có đăng kí tài khoản trên website và tài khoản đã được kích hoạt.

Danh sách chức năng

Phân hệ này bao gồm các chức năng của phân hệ khách và các **chức năng** sau:

* Đăng nhập
* Đăng Xuất
* Lấy lại mật khẩu tài khoản
* Xem tin nhắn của mình hay nhắn tin cho người dùng khác.
* Đăng bài bán sản phẩm bán đồ tennis ở mục bất kì
* Được xem và Đăng bài trả lời cho bài bán sản phẩm bán đồ tennis của mình hoặc của người khác ( up bài )
* Báo cáo bài viết vi phạm nội quy.
* Báo cáo thành viên vi phạm.
* Xem và chình sửa thông tin cá nhân trong tài khoản
* Chỉnh sửa bài bán sản phẩm bán đồ tennis hoặc bài trả lời của mình.
* Đánh giá người dùng khác hoặc bài bán sản phẩm bán đồ tennis của người khác ( tin cậy, ko tin cậy)
* Đánh dấu bài bán sản phẩm bán đồ tennis bất kì để ghi nhớ giúp theo dõi nhanh và thuận tiện hơn
* Liệt kê bài viết của mình.
* Tự động phát tán , lấy tin bán sản phẩm bán đồ tennis từ website khác

### Phân hệ Administrator (Quản trị viên)

Giới thiệu

Administrator là những người quản lí website, phụ trách theo dõi và quản lí toàn bộ người dùng và hoạt động của website. Administrator có tài khoản riêng đặc biệt có thể truy cập vào trang quản lí và được phép sử dụng các công cụ quản lí website tại trang này.

Danh sách chức năng

Phân hệ này bao gồm các chức năng sau:

* Xem và duyệt tin bán hàng
* Thêm, Xóa tin bán
* Thêm, Xóa, Sửa chuyên mục chính/ con
* Thêm, Xóa, Sửa banner, tin tức quảng cáo :người quản trị có thể thực hiện các thao tác như:
  + Thêm banner, tin tức quảng cáo mới vào vị trí cụ thể trên trang web
  + Xóa banner không hợp lệ hoặc đã hết hạn ra khỏi trang
  + Sửa lại banner cho phù hợp hoặc theo yêu cầu đối tác
* Thêm, Xóa, Sửa banner, logo, theme…
* Cấu hình mail Server

## Phân tích thiết kế hệ thống



### Danh sách các Usecase của hệ thống

#### 3.4.1.1 Phân hệ Guest

* Xem danh sách tất cả các mục bán sản phẩm đồ tennis chính
* Xem danh sách các mục bán sản phẩm đồ tenniscon trong 1 mục bán sản phẩm đồ tennis chính
* Liệt kê các bài bán sản phẩm đồ tennis
* Liệt kê các bài bán sản phẩm đồ tennis mới nhất
* Liệt kê các bài bán sản phẩm đồ tennis đáng tin cậy nhất
* Liệt kê các bài bán sản phẩm đồ tennis có người đăng đáng tin cậy nhất
* Liệt kê các bài bán sản phẩm đồ tennis có bài trả lời nhiều nhất
* Liệt kê các bài bán sản phẩm đồ tennis có nhiều người xem nhất
* Xem nội dung bài bán sản phẩm đồ tennis bất kì
* Tìm kiếm cơ bản bài bán sản phẩm đồ tennis theo từ khóa bất kì
* Tìm kiếm nâng cao theo từ khóa …
* Đăng kí tài khoản thành viên thông thường
* Xem thông tin liên hệ người dùng khác
* Đăng kí Email nhận tin bán sản phẩm đồ tennismới theo từng mục hay tất cả.

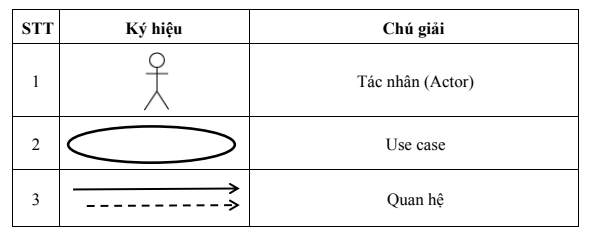
#### 3.4.1.2 Phân hệ thành viên

* Đăng nhập
* Đăng Xuất
* Lấy lại mật khẩu tài khoản
* Xem thông tin cá nhân trong tài khoản
* Chỉnh sửa thông tin cá nhân trong tài khoản
* Đánh giá người dùng khác
* Đăng bài bán sản phẩm đồ tennis ở mục bất kì
* Đăng bài trả lời cho bài bán sản phẩm đồ tennis của mình hoặc của người khác (up bài)
* Chỉnh sửa bài bán sản phẩm đồ tennis hoặc bài trả lời của mình.
* Đánh giá hoặc bài bán sản phẩm đồ tennis của người khác
* Đánh dấu bài bán sản phẩm đồ tennis bất kì để ghi nhớ giúp theo dõi nhanh và thuận tiện hơn.
* Liệt kê bài viết của mình.
* Tự động phát tán , lấy tin bán sản phẩm đồ tennis từ website khác

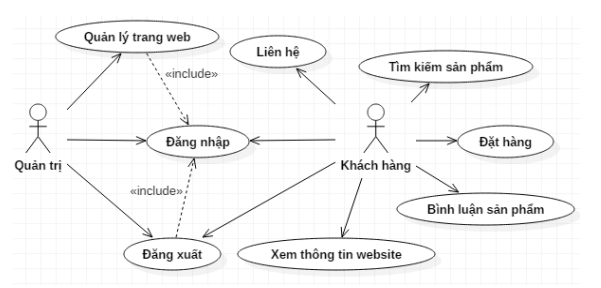
#### 3.4.1.3 Phân hệ Administrator

* Xem và duyệt tin bán sản phẩm đồ tennis
* Thêm, Xóa tin bán sản phẩm đồ tennis
* Thêm, Xóa, Sửa chuyên mục chính/ con
* Thêm, Xóa, Sửa banner, tin tức quảng cáo
* Thêm, Xóa, Sửa banner, logo, theme…
  + 1. Sơ đồ use case

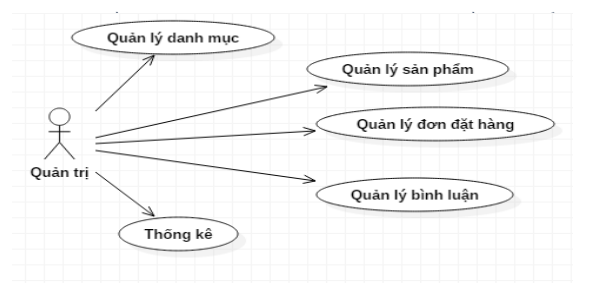
*Bảng 3-5: Các ký hiệu*



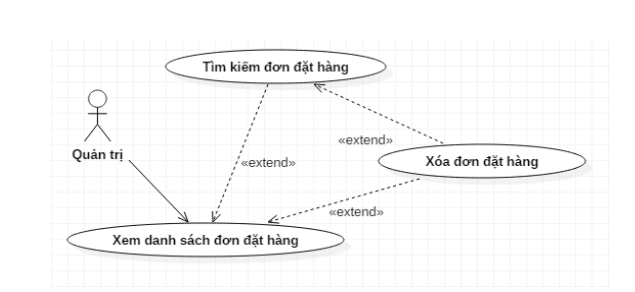
* Các sơ đồ use case



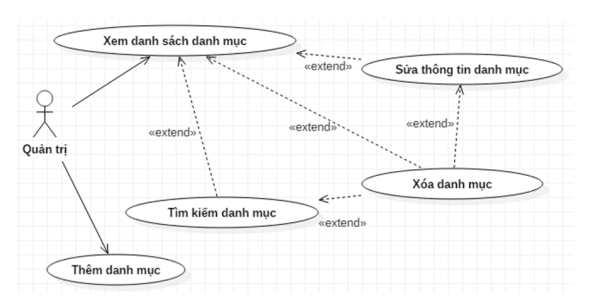
*Hình 3.1: Sơ đồ use case tổng quát*



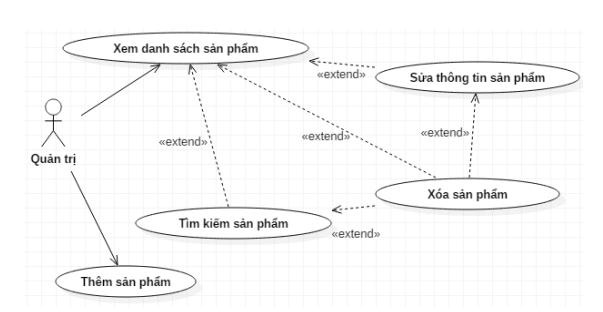
*Hình 3.2: Sơ đồ use case quản lý phía người quản trị*



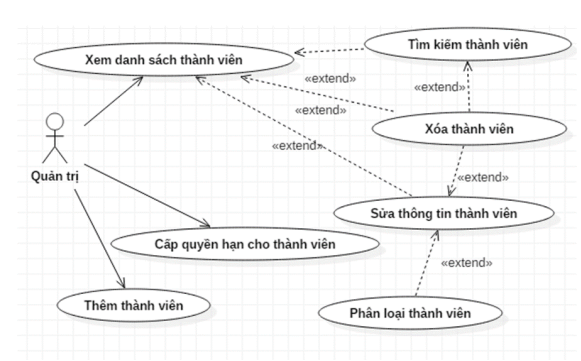
*Hình 3.3: Sơ đồ use case quản lý đặt hàng*



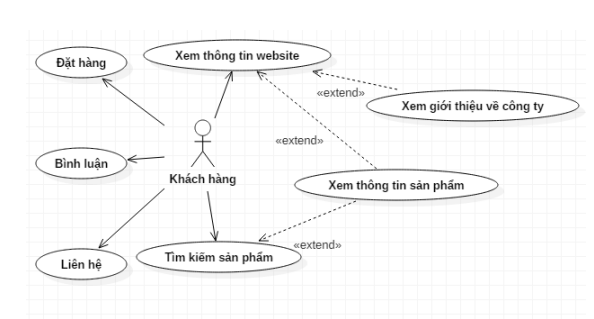
*Hình 3.4: Sơ đồ use case quản lý danh mục*



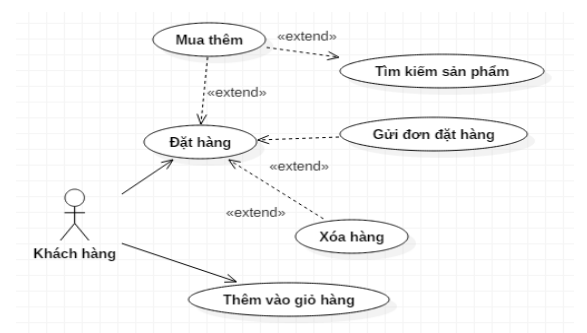
*Hình 3.5: Sơ đồ use case quản lý sản phẩm*



*Hình 3.6: Sơ đồ use case quản lý thành viên*



*Hình 3.7: Sơ đồ use case quản lý phía khách hàng*



*Hình 3.8: Sơ đồ use case đặt hàng*

### Mô tả chi tiết một số use case

#### 3.4.3.1. Phân hệ Khách (Guest)

#### 3.4.3.2. Xem danh sách tất cả các mục bán sản phẩm đồ tennischính

1. Điều kiện tiên quyết: Không có.
2. Luồng sự kiện chính: Use case này được thực hiện khi người dùng vào trang chủ của hệ thống.

* Hệ thống hiển thị màn hình danh sách tất cả các mục bán sản phẩm đồ tennischính.
* Hệ thống hiển thị các thông tin liên quan đến các mục bán sản phẩm đồ tennischính như: Các mục bán sản phẩm đồ tenniscon, các thuộc tính phân loại, tổng số tin bán sản phẩm đồ tennisthuộc mục chính đó, hình ảnh đại diện cho mục chính đó.
* Kết thúc Use-Case.

1. Điều kiện kết thúc Use-Case

* Toàn bộ các mục bán sản phẩm đồ tennischính được hiển thị thành công.

1. Các luồng sự kiện khác

* Khi người dùng chọn vào 1 mục bán sản phẩm đồ tennischính thì danh sách các mục bán sản phẩm đồ tennisđược thực hiện.
* Khi người dùng chọn chức năng tìm kiếm mục bán sản phẩm đồ tennisthì tìm miếm cơ bản bài bán sản phẩm đồ tennistheo từ khóa được thực hiện.

#### 3.4.3.3. Xem danh sách các mục bán sản phẩm đồ tenniscon trong một mục bán sản phẩm đồ tennis chính

Mô tả

1. Điều kiện tiên quyết: *không có*
2. Luồng sự kiện chính: Use case này được thực hiện khi người dùng chọn xem danh sách các mục bán sản phẩm đồ tenniscon của một mục bán sản phẩm đồ tennischính cụ thể.

* Hệ thống hiển thị màn hình danh sách các mục bán sản phẩm đồ tenniscon có trong mục bán sản phẩm đồ tennischính.
* Kết thúc Use-Case.

1. Điều kiện kết thúc Use-Case

* Toàn bộ mục bán sản phẩm đồ tenniscon của mục bán sản phẩm đồ tennischính đã chọn được hiển thị thành công.

1. Các luồng sự kiện khác

* Không có.

#### 3.4.3.4. Liệt kê các bài bán sản phẩm đồ tennistheo nội dung

Mô tả

* 1. Điều kiện tiên quyết: usecase xảy ra khi người dùng click vào bản tin bất kì
  2. Luồng sự kiện chính:
* Use case này được thực hiện khi người dùng vào trang chủ của hệ thống.
* Người dùng chọn chức năng liệt kê các bài bán sản phẩm đồ tennistheo nội dung.
* Hệ thống hiển thị kết quả liệt kê.
* Kết thúc Use-Case.
  1. Điều kiện kết thúc Use-Case
* Hệ thống hiển các kết quả sau khi liệt kê xong.
  1. Các luồng sự kiện khác
* Khi người dùng chọn chức năng liệt kê theo bài mới nhất thì “Liet ke theo bai moi nhat”được thực hiện.
* Khi người dùng chọn chức năng liệt kê theo bài đáng tin cậy nhất thì “Liet ke theo bai dang tin cay nhat”được thực hiện.
* Khi người dùng chọn chức năng liệt kê theo tài khoản đáng tin cậy nhất thì “Liet ke theo tai khoan dang tin cay nhat”được thực hiện.
* Khi người dùng chọn chức năng liệt kê theo bài mới nhất thì “Liet ke theo bai tra loi nhieu nhat”được thực hiện.
* Khi người dùng chọn chức năng liệt kê theo bài được xem nhiều nhất thì “Liệt kê theo bai xem nhieu nhất”được thực hiện.
* Khi người dùng chọn chức năng xem nội dung bài bán sản phẩm đồ tennisthì “Xem noi dung bai rao vat”được thực hiện.

#### 3.4.3.5. Xem nội dung bài bán sản phẩm đồ tennis

Mô tả

* 1. Điều kiện tiên quyết: usecase xảy ra khi người dùng click vào bản tin bất kì
  2. Luồng sự kiện chính: Use case này được thực hiện khi người dùng click vào bản tin bất kì.
* Hệ thống hiển thị màn hình bao gồm nội dung của bản tin, người đăng tin, thời gian đăng tin, số người đã xem và danh sách các bản tin cùng danh mục.
* Kết thúc Use-Case.
  1. Điều kiện kết thúc Use-Case
* Toàn bộ nội dung bản tin được hiển thị thành công.
  1. Các luồng sự kiện khác
* Khi người dùng chọn các tùy chọn tìm kiếm
* Khi người dùng chọn bản tin khác trong danh sách

#### 3.4.3.6. Tìm kiếm nâng cao

Mô tả

1. Điều kiện tiên quyết: *(điều kiện để Usecase này xảy ra)*

Người dùng phải kết nối thành công đến trang web

1. Luồng sự kiện chính:

* Use case này được thực hiện khi người dùng vào trang chủ của trang web, chọn chức năng tìm kiếm sau đó chọn chức năng tìm kiếm nâng cao.
* Trang web sẽ chuyển đến một trang mới, cho phép người dùng chọn gõ từ khóa tìm kiếm và chọn các mục tìm kiếm như bán sản phẩm đồ tennis, đại điểm, thời gian, người đăng …
* Sau khi người dùng đã nhập xong nội dung tìm kiếm và chọn các mục tìm kiếm , trang web sẽ tự động tìm kiếm trong CSDL
* Kết thúc Use-Case: Kết quả tìm kiếm được hiển thị trên web

1. Điều kiện kết thúc Use-Case

* Toàn bộ kết quả tìm kiếm được lưu lại và hiển thị lần lượt qua các trang.

1. Các luồng sự kiện khác

* Người dùng chọn Trang chủ thì sẽ chuyển về trang chủ

#### 3.4.3.7. Đăng ký thành viên

Mô tả

1. Điều kiện tiên quyết: *một trang bất kỳ của web bán sản phẩm đồ tennisđược mở.*
2. Luồng sự kiện chính: Use case này được thực hiện khi người dùng ở bất kỳ trang nào của web bán sản phẩm đồ tennis.

* Trên hệ thống menu ngang của trang bất kỳ, người dùng nhấn chuột chọn vào item “**Đăng ký**”
* Trang đăng ký thành viên thông thường thay thế trang hiện tại.
* Người dùng điền đẩy đủ thông tin bắt buộc (thông tin có dấu \* trên màn hình) và thông tin bổ sung (nếu có).
* Kết thúc Use-Case.

1. Điều kiện kết thúc Use-Case

* Người dùng đăng ký thành công hoặc người dùng hủy việc đăng ký vào các trang khác.

1. Các luồng sự kiện khác

* Không có.

#### 3.4.3.8. Xem thông tin liên hệ người dùng khác

Mô tả

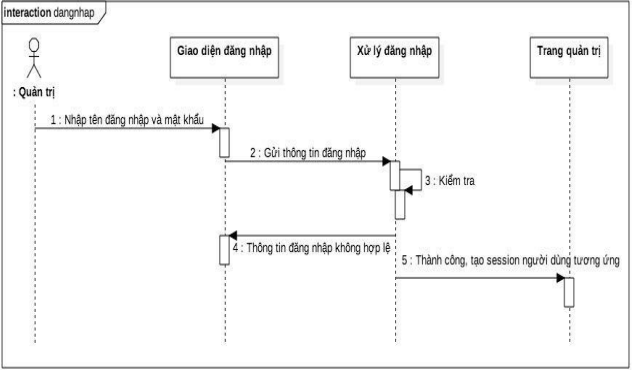
* 1. Điều kiện tiên quyết: Người dùng đang xem danh sách các tin của một chuyên mục.
  2. Luồng sự kiện chính: Use case này được thực hiện khi người dùng chọn xem chi tiết một tin.
* Hệ thống hiển thị màn hìnhthông tin chi tiết của tin, cùng với thông tin liên hệ của người đăng bài ở phía dưới như: tên người đăng, địa chỉ, số sản phẩm đồ tennis, email…
* Kết thúc Use-Case.
  1. Điều kiện kết thúc Use-Case
* Thông tin chi tiết của tin và người đăng tin được hiển thị thành công.
  1. Các luồng sự kiện khác (*không có*)

#### 3.4.3.9. Đăng kí Email nhận tin bán sản phẩm đồ tennismới theo từng mục hay tất cả

Mô tả

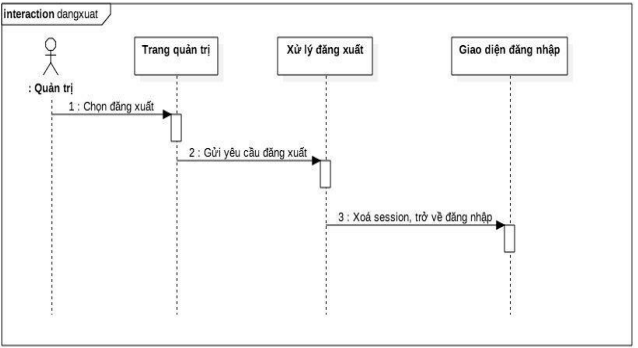
1. Điều kiện tiên quyết:
   * Không điều kiện.
2. Luồng sự kiện chính:
   * Usecase bắt đầu khi người dùng click vào mục “Đăng kí nhận tin bán sản phẩm đồ tennisqua Email”.
   * Màn hình “Đăng kí nhận tin bán sản phẩm đồ tennisqua Email” xuất hiện, người dùng gõ vào email muốn nhận tin bán sản phẩm đồ tennistạo mục “Your email address”.
   * Người dùng nhấn nút “Complete subscription request” để xác nhận việc đăng kí email.
   * Màn hình thông báo việc đăng kí thành công.
   * Usecase kết thúc.
3. Điều kiện kết thúc Use-Case:
   * Usecase kết thúc khi màn hình thông báo việc đăng kí thành công.
4. Các luồng sự kiện khác:
   * Không có luồng sự kiện khác.

### Sơ đồ tuần tự

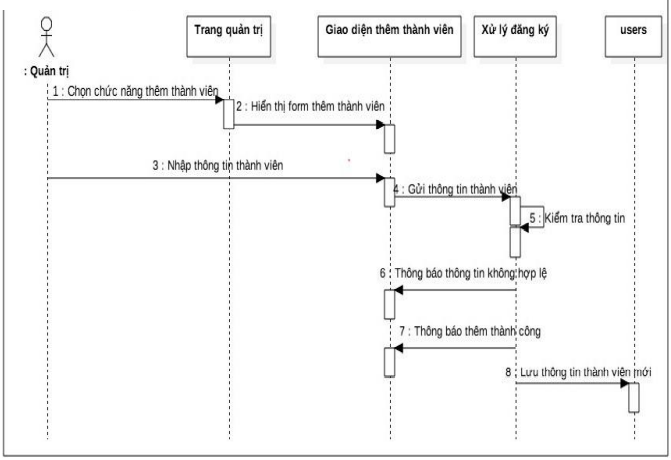


*Hình 3.9: Sơ đồ tuần tự đăng nhập*

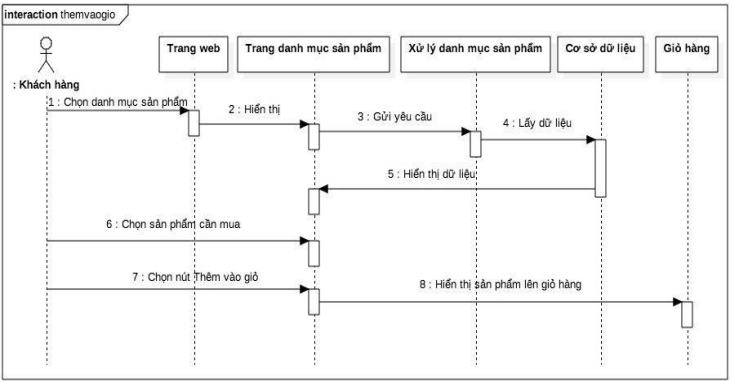




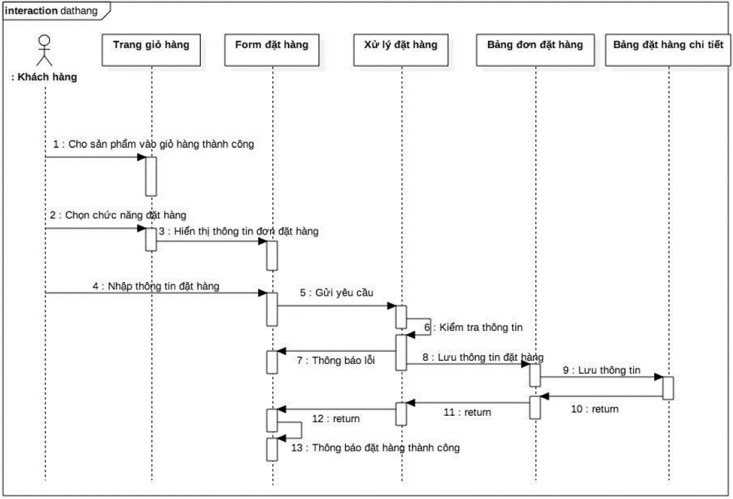
*Hình 3.10: Sơ đồ tuần tự đăng xuất*



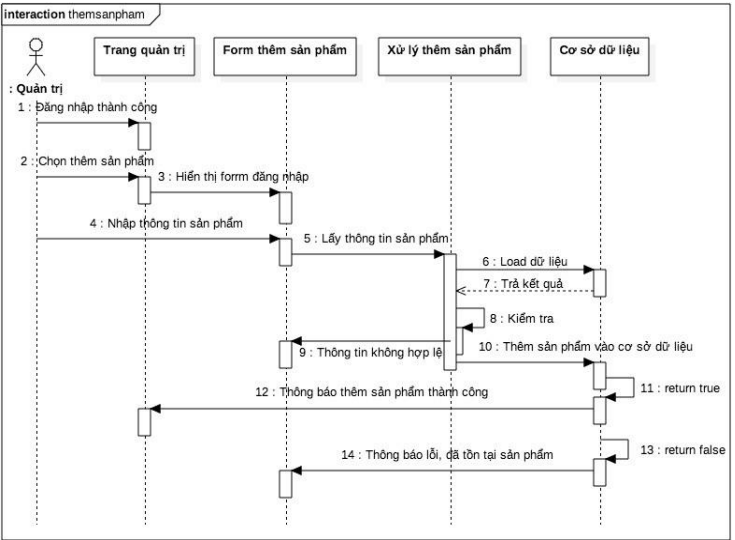
*Hình 3.11: Sơ đồ tuần tự thêm thành viên*



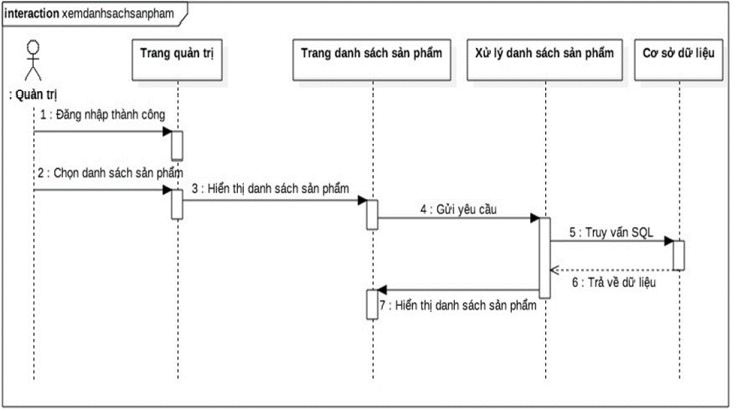
*Hình 3.12: Sơ đồ tuần tự thêm vào giỏ hàng*

**

*Hình 3.13: Sơ đồ tuần tự đặt hàng*

**

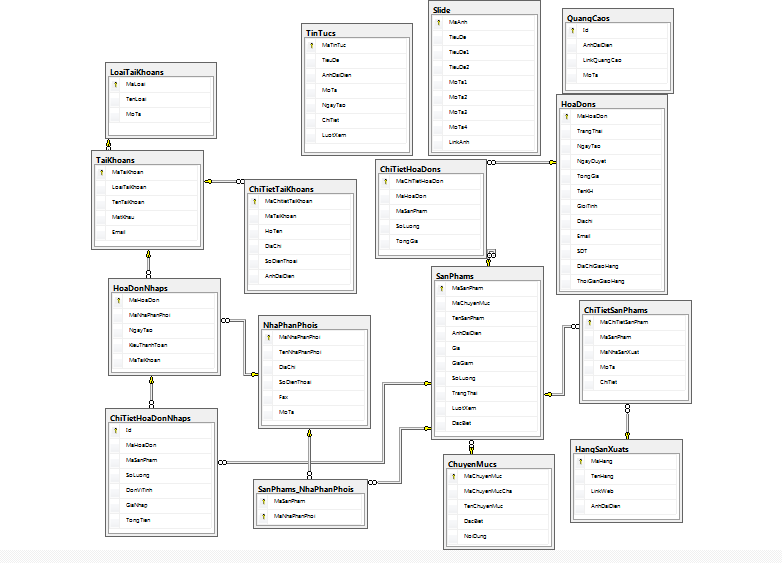
*Hình 3.14: Sơ đồ tuần tự thêm sản phẩm*

**

*Hình 3.15: Sơ đồ tuần tự xem danh sách sản phẩm*

## Thiết kế cơ sở dữ liệu

### Cơ sở dữ liệu và các mối quan hệ



*Hình 3.16: Biểu đồ dữ liệu quan hệ*

* + 1. Mô tả chi tiết các bảng trong cơ sở dữ liệu

***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 3-6: 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 3-7: Danh sách các thuộc tính Lớp Hãng sản xuất

***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 3-9: Danh sách các thuộc tính Lớp Sản phẩm

***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 3-11: Danh sách các thuộc tính Lớp Chi tiết hóa đơn

***i)Lớp người dùng***

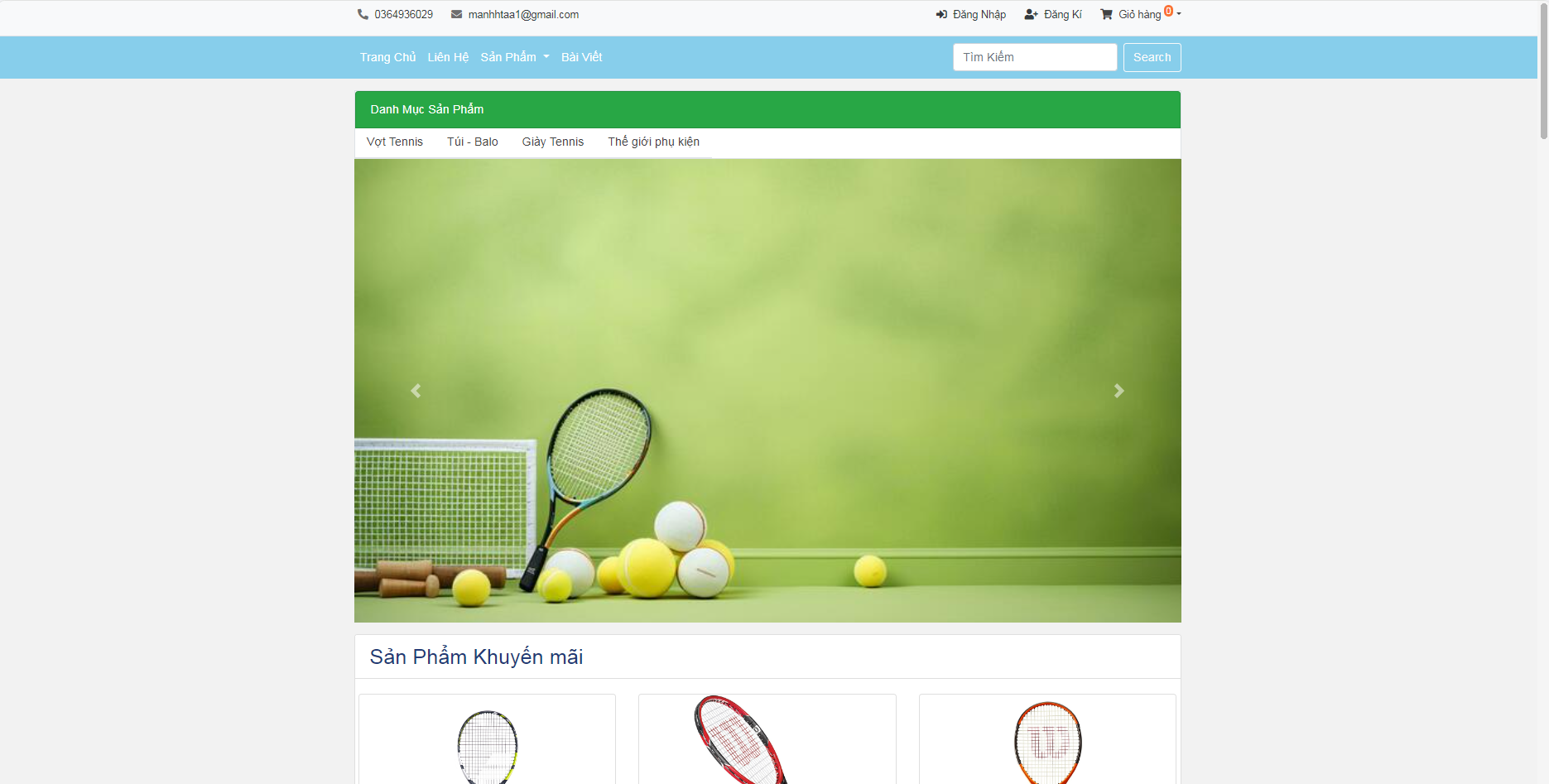
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | UserName | Varchar(10) | Khóa chính | Tên đăng nhập |
| **2** | PassWord | Varchar(10) | Không được rỗng | Mật khẩu |
| **3** | HoTen | Nvarchar(50) |  | Họ tên người dùng |
| **4** | MaNhom | 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 3-12: Danh sách các thuộc tính Lớp Người dùng

**CHƯƠNG 4: KẾT QUẢ CÀI ĐẶT CHƯƠNG TRÌNH**

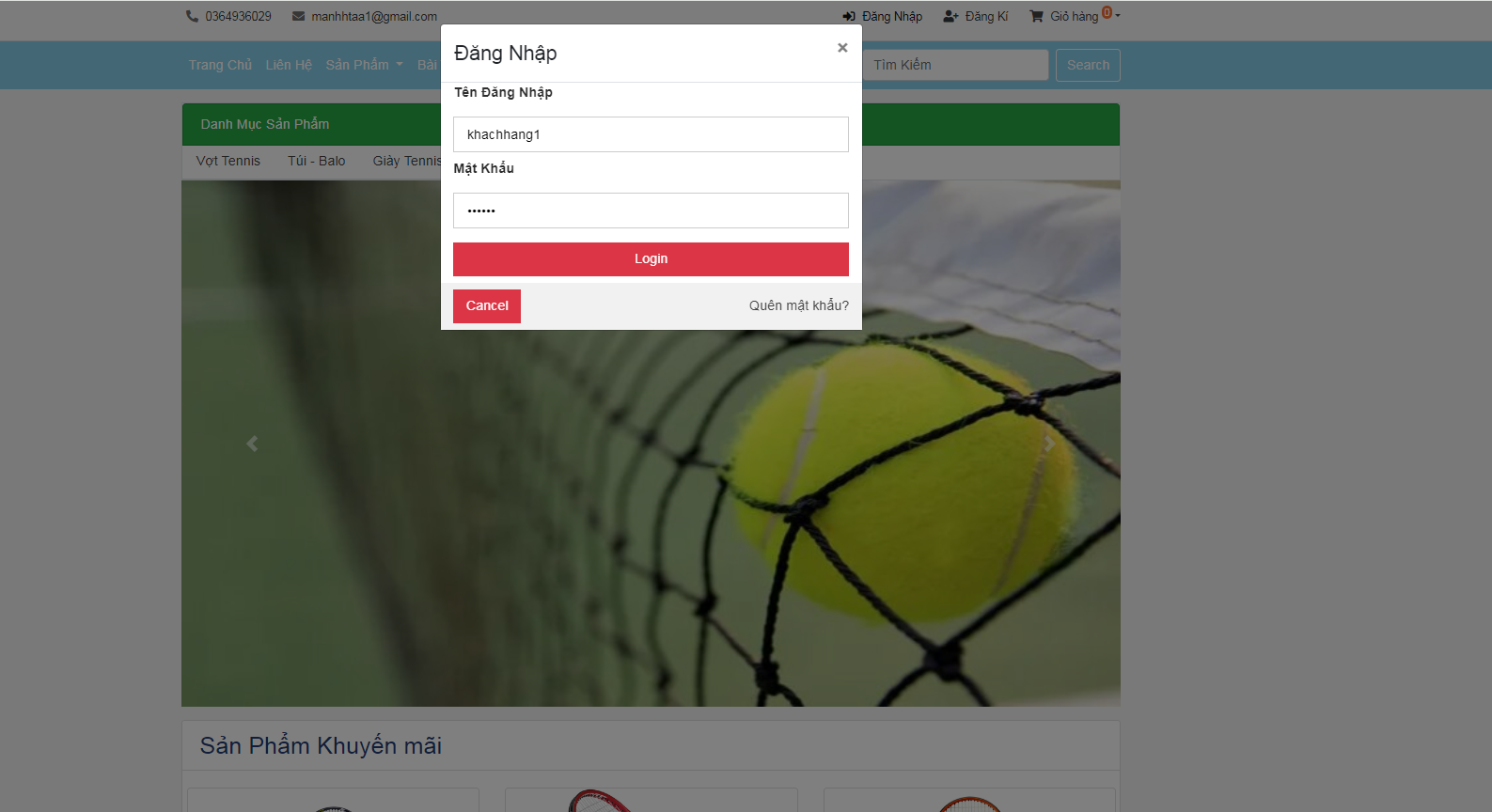


## Giao diện chính trang hiển thị



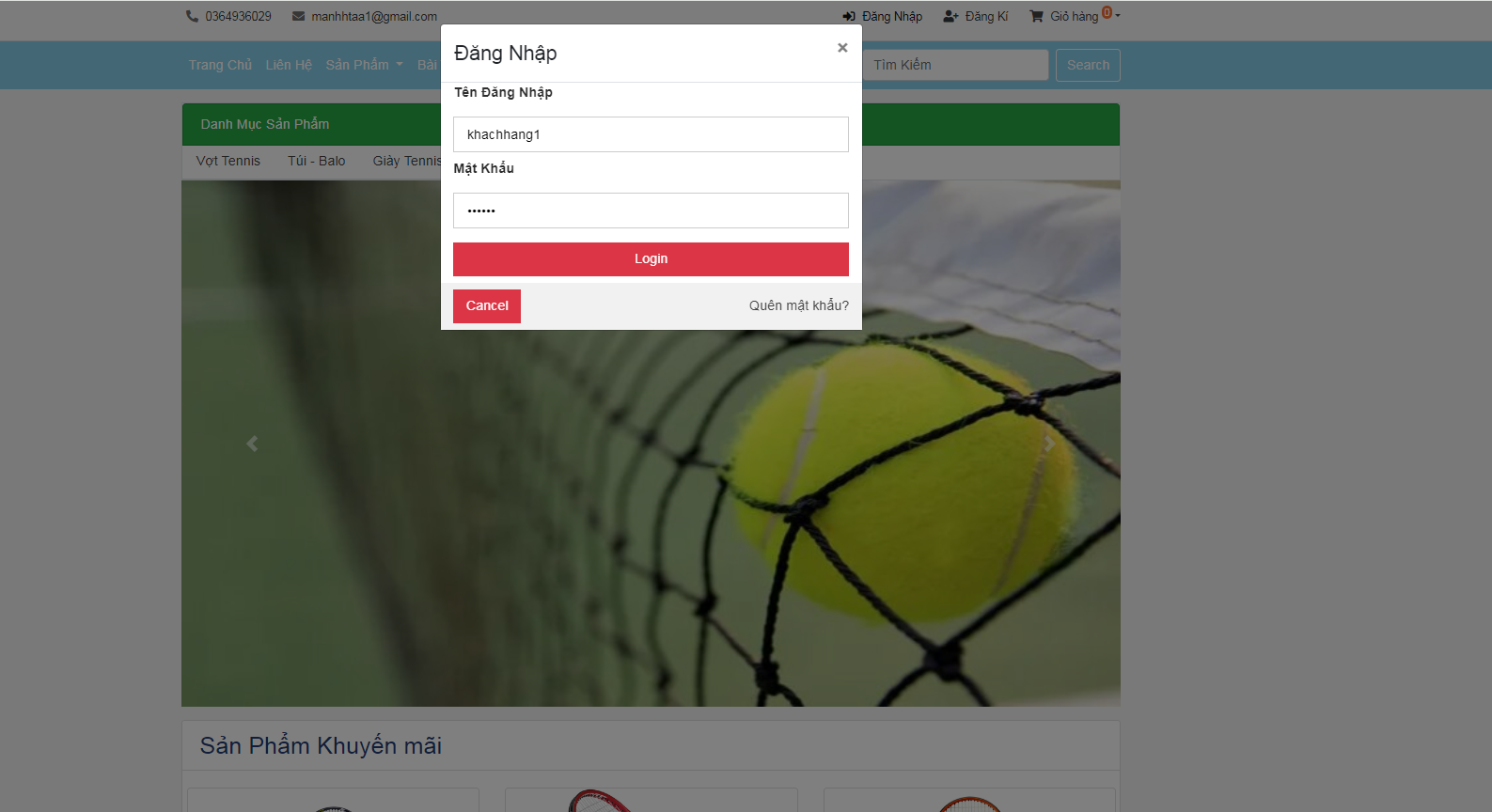
Hình 4.1: Giao diện chính trang hiển thị

## Giao diện trang đăng nhập



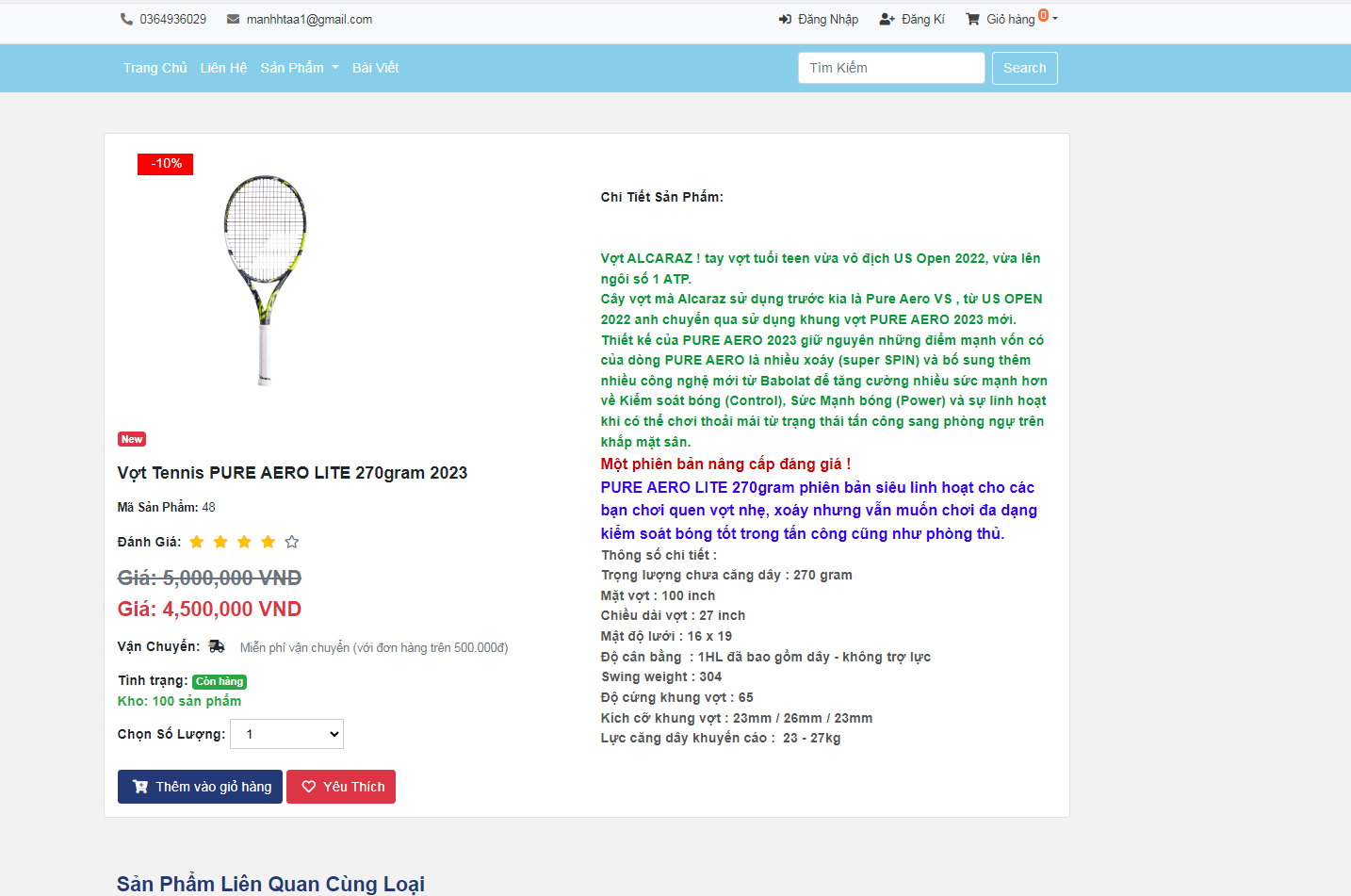
Hình 4.2: Giao diện đăng nhập

## Giao diện trang đăng kí

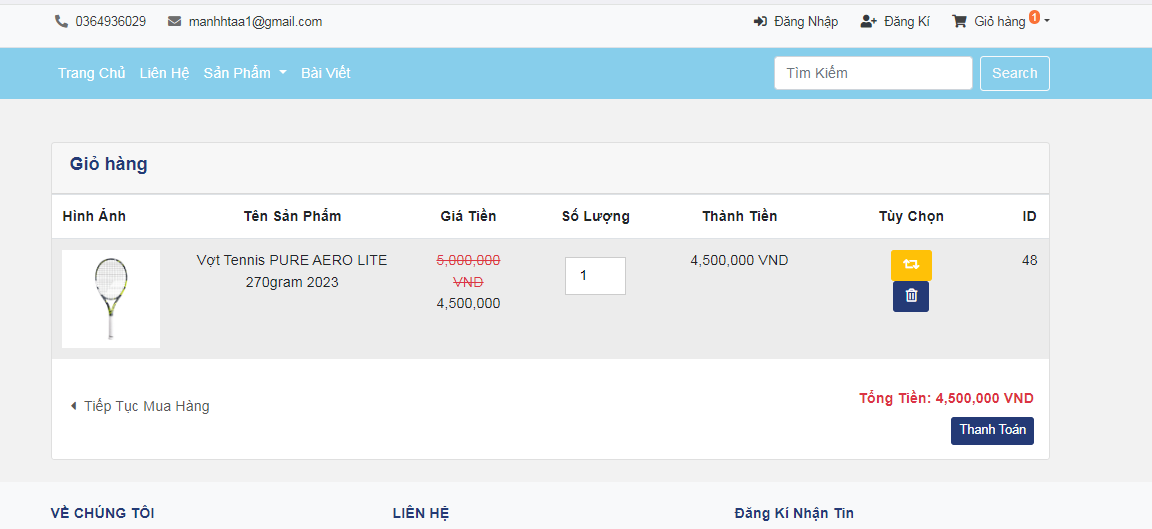


Hình 4-3: Giao diện trang đăng ký

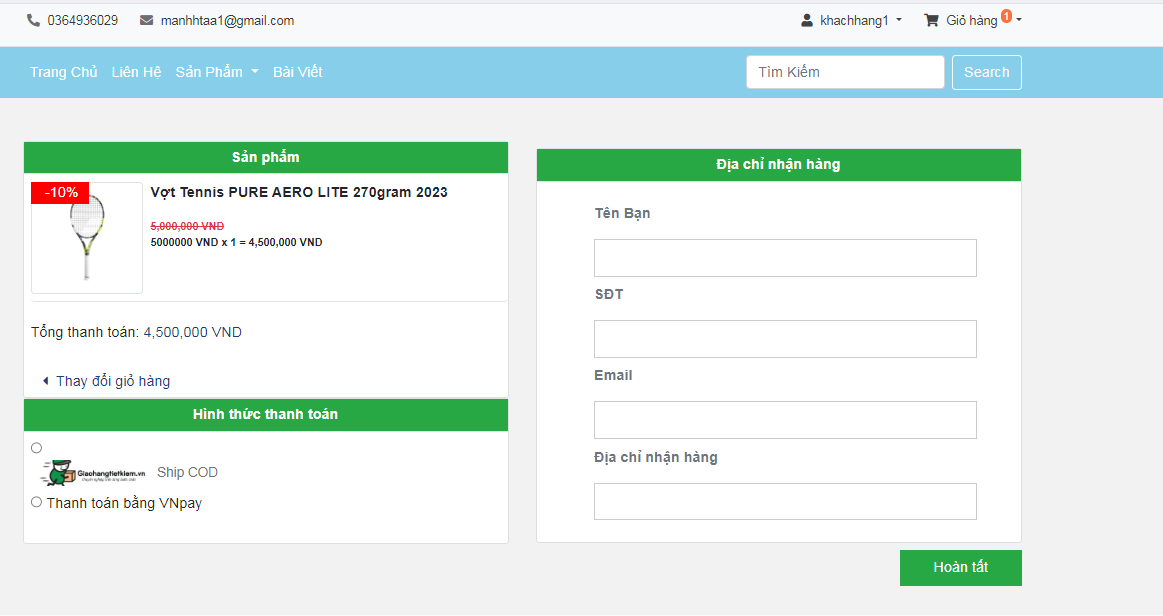
## Giao diện trang chi tiết sản phẩm



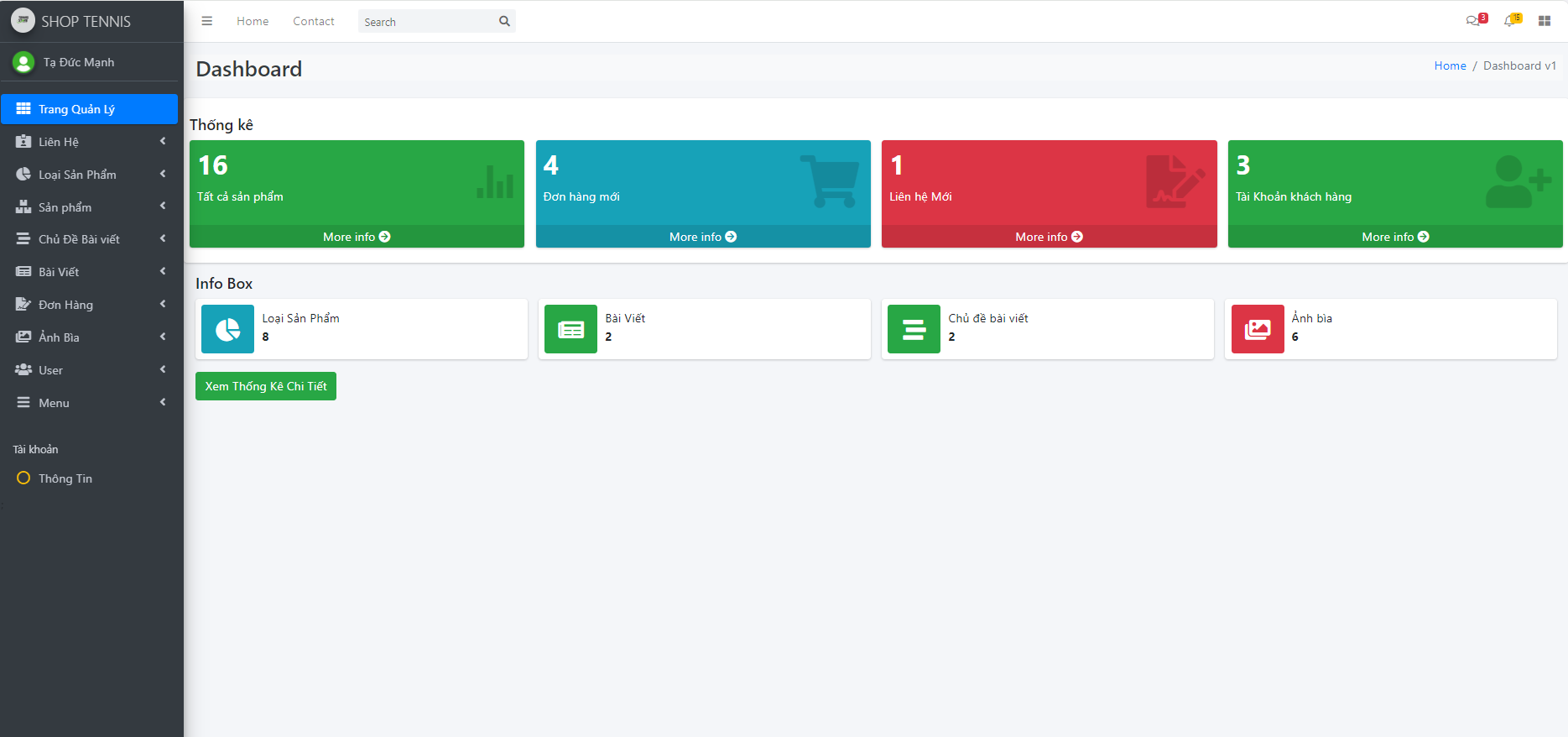
## Giao diện trang giỏ hàng



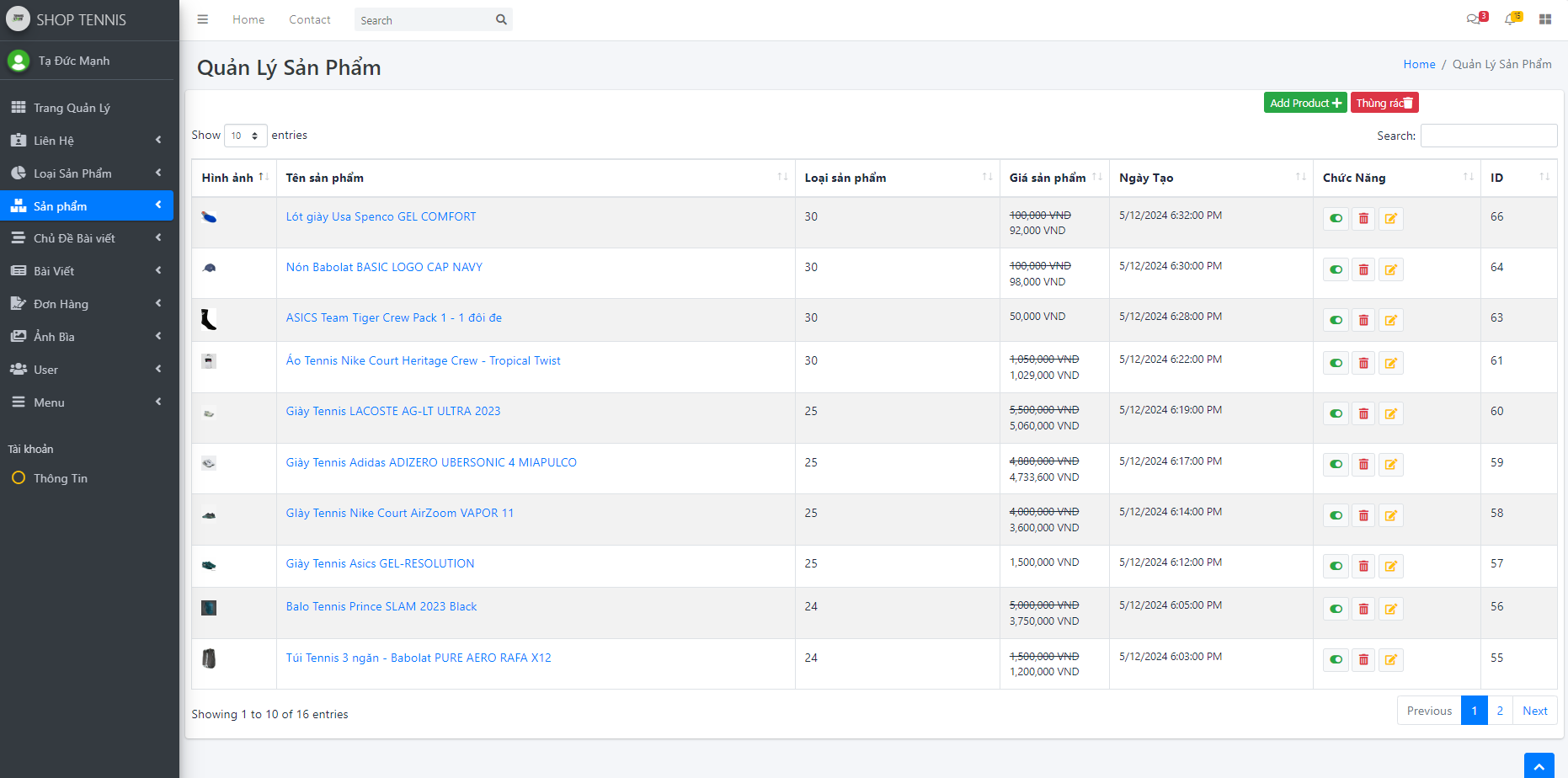
## Giao diện trang đặt hàng và thanh toán



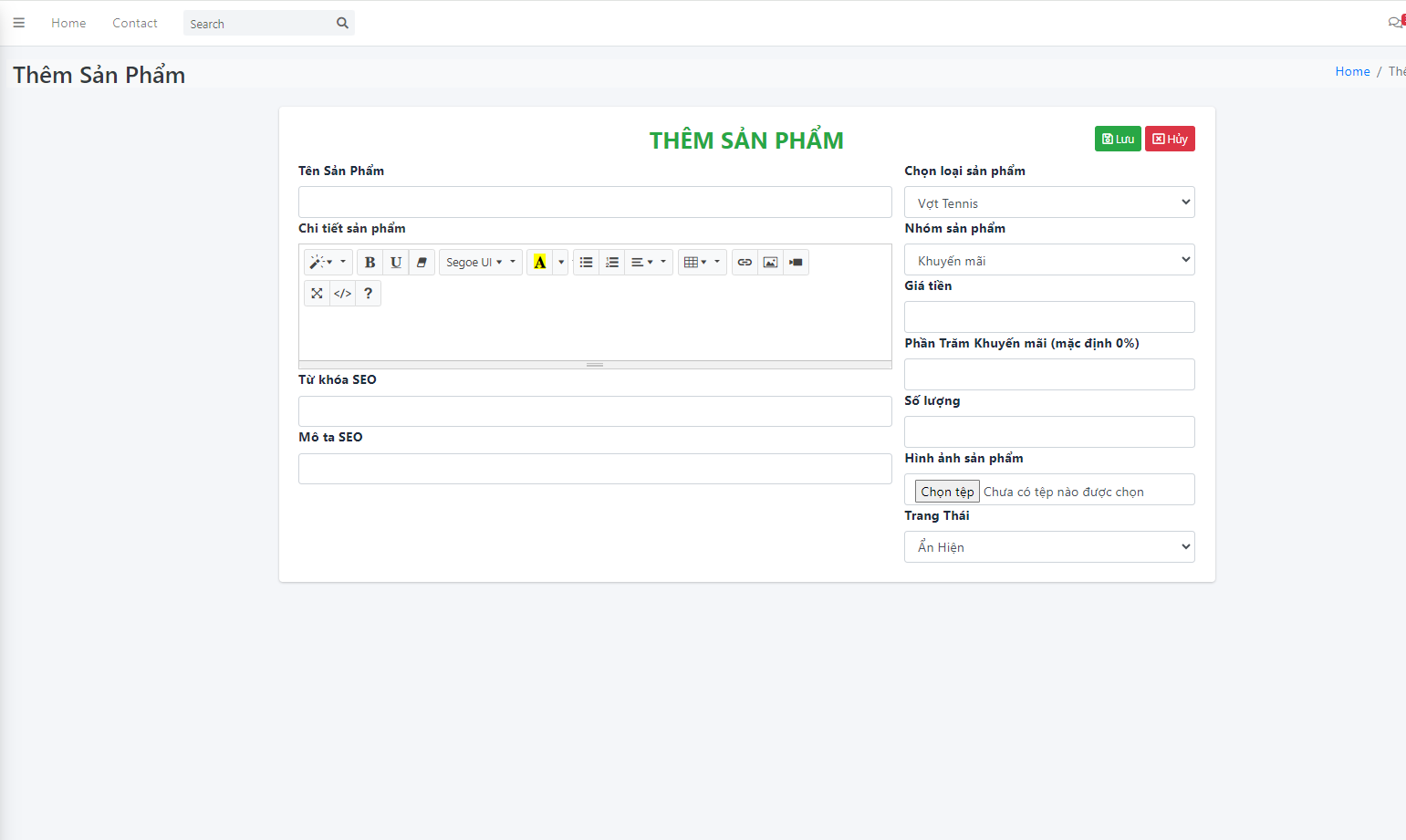
## Giao diện trang quản trị

 Giao diện trang chủ quản trị

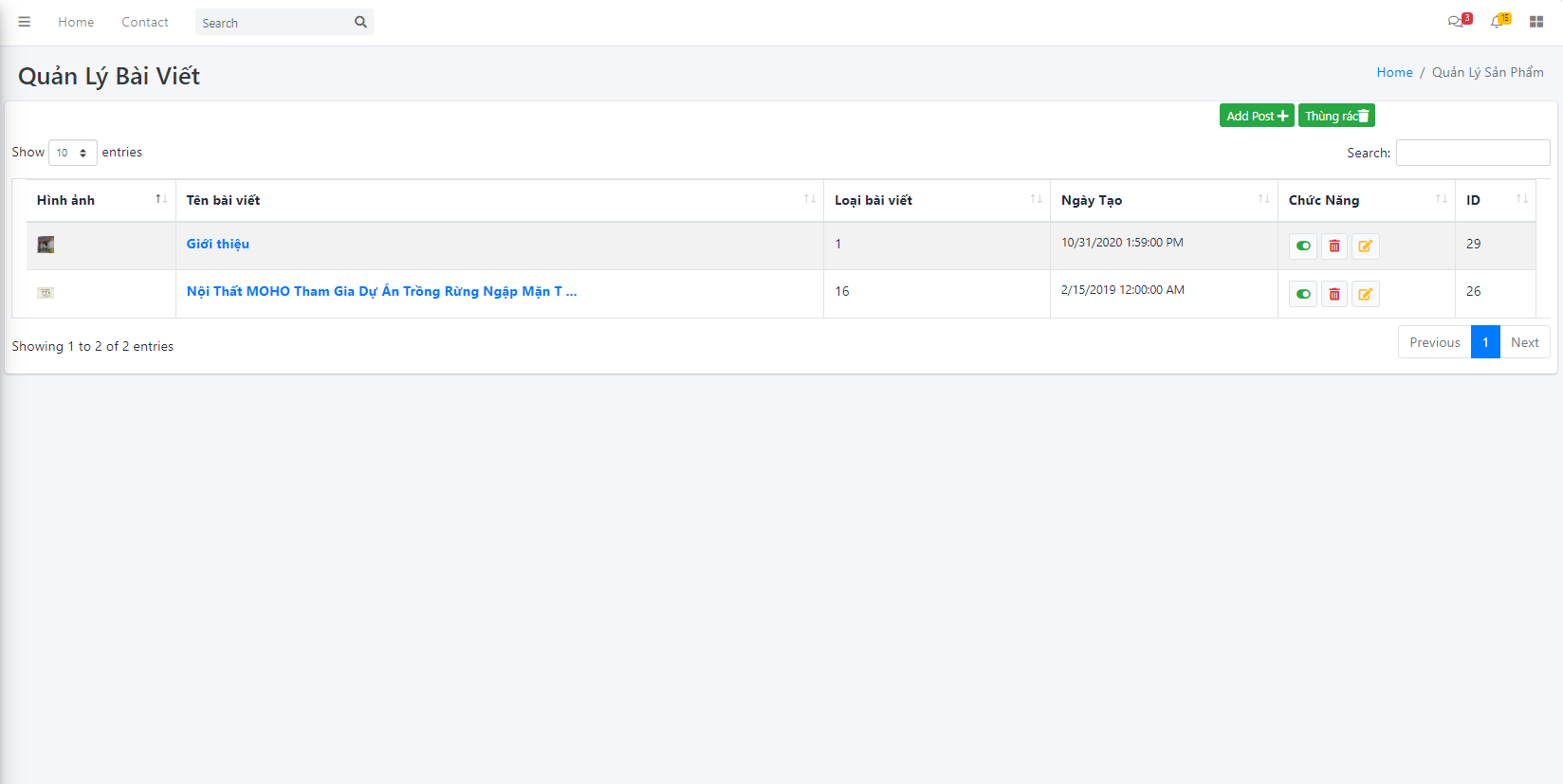
## Giao diện trang quản lý sản phẩm



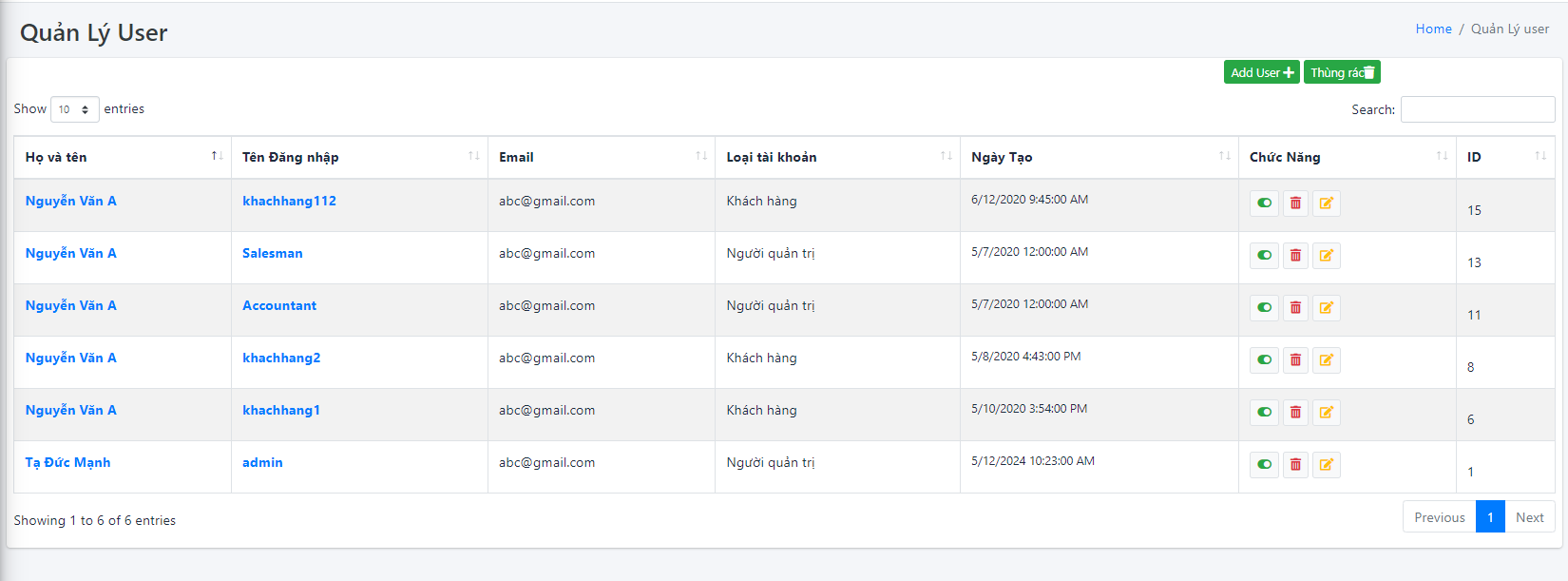
## Giao diện trang thêm sản phẩm



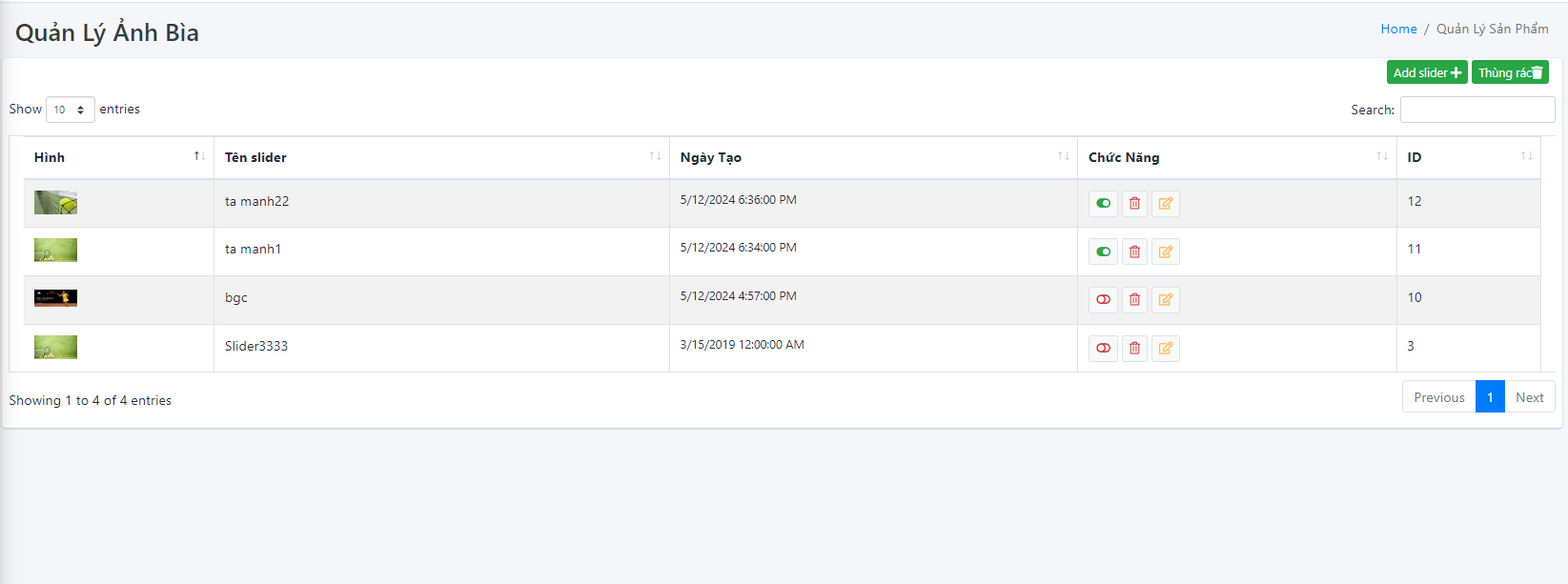
## Giao diện trang quản lý bài viết



## Giao diện trang quản lý tài khoản



## Giao diên cài đặt slide show



Hình 3-36: Giao diện cài đặt slide show

**4.19. Kiểm thử**

**4.19.1. Kiểm thử hộp đen**

Thông tin về website:

* Khi đăng ký tài khoản tại website, người dùng được yêu cầu nhập vào các thông tin: username (tên tài khoản), password (mật khẩu), email.
* Khi đăng nhập tài khoản, người dùng được yêu cầu nhập vào thông tin username, password đã đăng ký.
* Có 2 loại người dùng:
* Người dùng thông thường: có thể chỉnh sửa thông tin cá nhân trong tài khoản của mình
* Người dùng là quản trị viên (admin): có thể xem danh sách người dùng thông thường, xóa, sửa, tìm kiếm danh sách người dùng.

Đồ án sẽ sử dụng kỹ thuật kiểm thử đoán lỗi để kiểm tra các trường hợp:

❖ Ca kiểm thử trường hợp đăng nhập, tạo user thành công cho tài khoản admin như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Test Title | Description | Pre-Condition |
| GUI | Chức năng đăng nhập và sửa user thành công cho account Admin | Kiểm tra chức năng đăng nhập khi người dùng nhập các thông tin hợp lệ, hoàn tất việc đăng nhập và sửa user thành công | Mở sẵn trình duyệt. Truy cập trang đăng ký tài khoản trên trình duyệt |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Steps | Data | Expected Result | Actual Result | Status |
| 1. Truy cập trang Sign In | Truy cập website thành công | Trang đăng nhập mở | Truy cập website thành công. | pass |
| 2. Nhập tên tài khoản hợp lệ | Admin | Hiển thị username | Hiển thị username | pass |
| 3.Nhập tên mật khẩu hợp lệ | 123456 | Nhập thành công | Nhập thành công | pass |
| 4. Click nút Sign In | Button Sign In | Chuyển sang trang Home | Chuyển sang trang Home | pass |
| 5. Click nút Profile | Button Profile | Chuyển sang trang Admin | Chuyển sang trang Admin | pass |

* + 1. **Kiểm thử tương thích**

Kiểm thử tương thích nhằm mục đích đánh giá sự tương thích giữa ứng dụng với các môi trường, nền tảng khác nhau. Kiểm thử tương thích áp dụng cho các hệ điều hành, phần mềm trình duyệt và các nền tảng phần cứng khác nhau: cấu hình, độ phẩn giải cấu hình.

|  |  |  |  |
| --- | --- | --- | --- |
| **Thiết bị** | **Phần cứng** | **Phần mềm** | **Kết quả** |
| PC | CPU: Intel Core i9-11900K  RAM: 32GB  Màn hình: 27” FHD | HĐH: Window 10 Trình duyệt:  • Google Chrome • Microsoft Edge  • Cốc Cốc | Đạt |
| Dell inspiron 15 5515 | CPU: AMD Ryzen 5 5500U  RAM: 8GB  Màn hình: 15.6” HD | HĐH: Window 10 Trình duyệt:  • Google Chrome • Cốc Cốc | Đạt |
| MacBook Air 2021 | CPU: Apple M1 RAM: 8GB  Màn hình: 13.3” | HĐH: macOS Trình duyệt:  • Safari | Đạt |

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**



## Kết luận

Do thời gian thực hiện đề tài có hạn chế, và sử dụng một số công nghệ và dự án nên em vẫn chưa hoàn thành được tất cả các yêu cầu mà giáo viên hướng dẫn đã giao. Em đã xây dựng được website với các chức năng cơ bản sau:

+ Nội dung:

* Trưng bày sản phẩm và mua hàng trực tuyến
* Tìm kiếm sản phẩm
* Quản lý thành viên
* Quản lý danh mục sản phẩm
* Quản lý sản phẩm
* Chức năng giỏ hàng
* Quản lý đơn đặt hàng
* Quản lý liên hệ, góp ý

+ Hình thức:

* Giao diện thân thiện, đẹp mắt, dễ sử dụng
* Các bước đặt hàng đơn giản, không mất nhiều thời gian
* Trưng bày sản phẩm rõ ràng, hình ảnh, thông tin phong phú

+ Hạn chế

* Chưa xuất đơn hàng ra file excel và pdf

**Hướng phát triển**

Hiện tại đối với các chức năng hiện tại của hệ thống vẫn còn một số chức năng cần cải thiện hơn trong tương lai như:

* Xây dựng thêm một số chức năng chưa hoàn thiện: xuất file excel, pdf…
* Cải thiện giao diện ngày càng đẹp mắt, dễ sử dụng, phù hợp với thị hiếu khách hàng.
* Phát triển tính năng bảo trì, khôi phục dữ liệu khi gặp sự cố.
* Cập nhật, áp dụng UI/UX vào xây dựng thiết kế website nâng cao tính tương tác giữa sản phẩm phần mềm với người sử dụng.
* Tích hợp tính năng hỗ trợ khách hàng trực tuyến, chat online.

**TÀI LIỆU THAM KHẢO**

[1] **Ths. Phan Tấn Tài & Ths. Đinh Khắc Quyền**, *Phân tích và thiết kế hệ thống thông tin.*

[2] **Đặng Văn Đức**, *Phân tích thiết kế hướng đối tượng UML*, NXB Khoa học và Kỹ thuật, 2002

[3[]http://www.hanhtranglaptrinh.com/2012/06/ma-hoa-md5-trong-c-aspnet.html?m=1](%5dhttp://www.hanhtranglaptrinh.com/2012/06/ma-hoa-md5-trong-c-aspnet.html?m=1)

[4] <http://msdn.microsoft.com>

[5] ASP.NET tutorials: ([https://docs.microsoft.com/en-us/aspnet/tutorials](https://docs.microsoft.com/en-us/aspnet/tutorials%20) )

[6] W3Schools Online Web Tutorials. (<https://www.w3schools.com>)