**BỘ CÔNG THƯƠNG**

**TRƯỜNG CAO ĐẲNG CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỀ CƯƠNG CHI TIẾT**

**THỰC TẬP TỐT NGHIỆP**

**XÂY DỰNG WEBSITE BÁN ĐỒ ĐIỆN TỬ**

**(SỬ DỤNG HTML,CSS, JAVASCRIPT, ASP.NET CORE MVC, SQL SERVER)**

**Giảng viên hướng dẫn: Hồ Diên Lợi**

**Thực hiện:** Nguyễn Đăng Bảo

MSSV: 2121110369 Lớp: CCQ2111F

TP Thủ Đức, ngày tháng 2024

Mục lục

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 4](#_Toc152103738)

[1.1. Lí do chọn đề tài 4](#_Toc152103739)

[1.2. Mục đích nghiên cứu 4](#_Toc152103740)

[1.3. Phương pháp nghiên cứu 4](#_Toc152103741)

[1.4. Đối tượng nghiên cứu 5](#_Toc152103742)

[1.5. Bố cục đề tài 5](#_Toc152103743)

[CHƯƠNG 2. CƠ SỞ LÍ THUYẾT 8](#_Toc152103744)

[2.1. RESTful Api 8](#_Toc152103745)

[2.1.1. Giới thiệu 8](#_Toc152103746)

[2.1.2. Cách hoạt động 9](#_Toc152103747)

[2.2. ASP.NET Core 9](#_Toc152103748)

[2.2.1. Giới thiệu về ASP.NET Core 9](#_Toc152103749)

[2.2.2. Cấu trúc của ASP.NET Core 10](#_Toc152103750)

[2.3. SQLServer 14](#_Toc152103751)

[2.3.1. Giới thiệu SQLServer 14](#_Toc152103752)

[2.3.2. Các thành phần của SQLServer 14](#_Toc152103753)

[2.4. React JS 15](#_Toc152103754)

[2.4.1. Giới thiệu về React JS 15](#_Toc152103755)

[2.4.2. Các thành phần của React JS 15](#_Toc152103756)

[2.5. TailwindCSS 16](#_Toc152103757)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 18](#_Toc152103758)

[3.1. Thực trạng bài toán 18](#_Toc152103759)

[3.2. Phân tích 18](#_Toc152103760)

[3.2.1. Danh sách các actor 18](#_Toc152103761)

[3.2.2. Danh sách các use case 18](#_Toc152103762)

[3.2.2. Biểu đồ use case 20](#_Toc152103763)

[3.2.3. Biểu đồ hoạt động 22](#_Toc152103764)

[3.2.4. Biểu đồ tuần tự 32](#_Toc152103765)

[3.3. Giao diện 42](#_Toc152103766)

[3.4. Cơ sở dữ liệu 60](#_Toc152103767)

[CHƯƠNG 4. ỨNG DỤNG 64](#_Toc152103768)

[4.1. Trang người dung 64](#_Toc152103769)

[4.1.1. Trang quản trị 64](#_Toc152103770)

[4.1.2. Trang người dùng 64](#_Toc152103771)

[CHƯƠNG 5. TỔNG KẾT 65](#_Toc152103772)

[5.1. Kết quả đạt được 65](#_Toc152103773)

[5.2. Kết quả chưa đạt được 65](#_Toc152103774)

[5.3. Hướng phát triển của đề tài 65](#_Toc152103775)

[TÀI LIỆU THAM KHẢO 65](#_Toc152103776)

# TỔNG QUAN VỀ ĐỀ TÀI

## Lí do chọn đề tài

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang phát triển mạnh mẽ, sự lan tỏa của công nghệ thông tin đã mở ra những cơ hội ứng dụng rộng rãi trong lĩnh vực kinh tế. Việc tích hợp công nghệ hiện đại vào các hoạt động kinh doanh không chỉ giúp tối ưu hóa hiệu suất làm việc mà còn đem lại độ chính xác cao, thúc đẩy sự tăng trưởng và hiệu quả sản xuất.

Trong lĩnh vực bán hàng, sự hòa nhập của công nghệ thông tin đã mở ra một hình thức mới - bán hàng trực tuyến. Việc này không chỉ mở rộng phạm vi tiếp cận khách hàng mà còn giảm thiểu chi phí về mặt không gian trưng bày sản phẩm. Sự thuận tiện của việc mua sắm trực tuyến cho khách hàng không chỉ đến từ việc không cần phải rời khỏi nhà, mà còn từ việc lựa chọn và đặt hàng diễn ra một cách nhanh chóng và dễ dàng.

Đã từ những lợi ích đáng kể này, tôi đã chọn đề tài "Xây dựng website bán điện thoại". Mục tiêu của dự án là cung cấp cho người bán hàng các công cụ hỗ trợ tiên tiến từ công nghệ thông tin trong việc kinh doanh. Đồng thời, nó cũng nhằm tạo ra những trải nghiệm mua sắm trực tuyến thuận lợi và tối ưu nhất cho khách hàng.

## Mục đích nghiên cứu

Mục đích chính của nghiên cứu này là tập trung vào việc xây dựng một website bán điện thoại với mục tiêu cải thiện trải nghiệm mua sắm trực tuyến. Chúng tôi đặt ra mục tiêu tối ưu hóa giao diện người dùng, tạo ra một trải nghiệm mua sắm trực tuyến thuận tiện và trực quan hơn cho khách hàng. Đồng thời, chúng tôi hướng đến việc nâng cao khả năng tương tác giữa người mua và người bán hàng thông qua việc phát triển các tính năng tương tác như hỗ trợ trực tuyến, đánh giá sản phẩm và cung cấp thông tin hữu ích.

Ngoài ra, mục tiêu của nghiên cứu cũng nhằm vào việc tối ưu quy trình bán hàng trực tuyến, từ việc thanh toán an toàn đến quản lý hàng tồn kho thông minh và phân tích dữ liệu khách hàng. Chúng tôi mong muốn rằng việc tối ưu hóa các khía cạnh này sẽ giúp tăng cường hiệu suất kinh doanh và mang lại lợi ích lớn cho cả người bán và người mua hàng.

## Phương pháp nghiên cứu

Đầu tiên, chúng tôi sẽ tiến hành nghiên cứu cặn kẽ về thị trường mua sắm điện thoại và thu thập thông tin chi tiết về nhu cầu của người dùng. Qua đó, chúng tôi sẽ phân tích và xác định yêu cầu cụ thể cho website bao gồm các tính năng, giao diện và trải nghiệm người dùng mong muốn.

Tiếp theo, chúng tôi sẽ bắt đầu quá trình thiết kế bằng việc tạo ra các bản vẽ wireframe và mô phỏng sơ bộ về giao diện người dùng (UI) cũng như trải nghiệm người dùng (UX). Qua đó, chúng tôi sẽ đảm bảo rằng website được thiết kế sao cho dễ sử dụng và hấp dẫn đối với người dùng.

Sau đó, chúng tôi sẽ tiến hành giai đoạn phát triển, sử dụng các công nghệ và framework phù hợp để xây dựng website. Quá trình này sẽ bao gồm tích hợp các tính năng quan trọng như quản lý sản phẩm, giỏ hàng, thanh toán an toàn, hỗ trợ khách hàng và các tính năng khác.

Cuối cùng, trước khi triển khai, chúng tôi sẽ tiến hành kiểm thử kỹ lưỡng để đảm bảo tính ổn định và hiệu suất của website. Qua đó, chúng tôi mong muốn tạo ra một nền tảng mua sắm trực tuyến chất lượng cao, phục vụ tốt nhất cho người bán và người mua hàng.

## Đối tượng nghiên cứu

Người dùng cuối (khách hàng): Đây là nhóm người trực tiếp sử dụng website để thực hiện quá trình mua sắm. Đối tượng này bao gồm các đối tượng khách hàng tiềm năng hoặc hiện tại có nhu cầu mua điện thoại di động. Đa dạng về độ tuổi, sở thích và nhu cầu sử dụng điện thoại của họ sẽ cung cấp thông tin quan trọng về trải nghiệm người dùng cũng như các tính năng và yếu tố quyết định tạo nên trải nghiệm mua sắm trực tuyến.

Người quản lý và phát triển website: Đây là nhóm những người có trách nhiệm thiết kế, phát triển, và duy trì website bán điện thoại. Đối tượng này bao gồm các nhà phát triển web, nhà quản lý dự án, nhà thiết kế UI/UX, chuyên gia marketing và các bộ phận liên quan, nhằm nghiên cứu về quy trình xây dựng, tối ưu hóa website và cải thiện trải nghiệm người dùng.Thông qua việc tìm hiểu và nghiên cứu đối tượng này, sẽ giúp cải thiện và tối ưu hóa website để đáp ứng tốt nhất nhu cầu của khách hàng.

## Bố cục đề tài

1. **Giới thiệu**
   1. **Bối cảnh**

Trong thời đại công nghệ ngày nay, thương mại điện tử đang trở thành một xu hướng không thể phủ nhận. Việc tạo ra một trang web bán hàng điện tử không chỉ là việc kinh doanh mà còn là sự đáp ứng đúng đắn với nhu cầu ngày càng tăng của khách hàng.

* 1. **Mục tiêu**

Dự án được khởi động với mục tiêu xây dựng một nền tảng thương mại điện tử hiện đại, linh hoạt và thuận tiện cho người dùng, đồng thời tối ưu hóa quy trình quản lý để đáp ứng nhu cầu ngày càng đa dạng của thị trường.

1. **Lí do và phương pháp lựa chọn đối tượng nghiên cứu**
   1. **Lý do chọn đề tài**

Lựa chọn xây dựng website bán hàng điện tử bắt nguồn từ sự thấu hiểu về sự chuyển đổi của thị trường và nhu cầu ngày càng cao của người tiêu dùng trong việc mua sắm trực tuyến.

* 1. **Phương pháp nghiên cứu**

Để xác định đối tượng nghiên cứu, chúng tôi đã tiến hành nghiên cứu thị trường, phân tích xu hướng người tiêu dùng và đánh giá khả năng cạnh tranh để chọn lựa một đối tượng có tiềm năng phát triển cao.

1. **Mục đích nghiên cứu**

Mục đích của dự án là xây dựng một trang web bán hàng điện tử chất lượng cao, mang lại trải nghiệm mua sắm trực tuyến tốt nhất cho khách hàng và giúp doanh nghiệp tối ưu hóa quy trình quản lý.

1. **Phương pháp nghiên cứu**

Sử dụng phương pháp nghiên cứu kết hợp giữa phân tích thị trường, nghiên cứu người dùng và thử nghiệm thực tế để đảm bảo tính hiệu quả và linh hoạt của website.

1. **Đối tượng nghiên cứu**

Đối tượng nghiên cứu của chúng tôi là thị trường tiêu dùng mục tiêu, chủ yếu là những người có nhu cầu mua sắm trực tuyến đồ điện tử.

1. **Kết quả dự kiến và ứng dụng**

Chúng tôi kỳ vọng rằng website sẽ mang lại trải nghiệm mua sắm tuyệt vời, tăng cường doanh số bán hàng và giảm chi phí quản lý. Kết quả này sẽ có ứng dụng rộng rãi trong việc phát triển kinh doanh và cải thiện dịch vụ khách hàng.

1. **Đóng góp và ý nghĩa của đề tài**

Dự án không chỉ đóng góp vào sự phát triển của doanh nghiệp mà còn mang lại giá trị cho cộng đồng người tiêu dùng, tạo ra một môi trường thương mại điện tử tích cực và hiệu quả.

1. **Tổng kết**

Trong quá trình xây dựng dự án website bán hàng điện tử, chúng tôi đã đặt ra một chuỗi các bước logic và chiến lược nhằm đảm bảo tính hiệu quả và độ ổn định của sản phẩm. Việc tập trung vào việc phân tích thị trường và lựa chọn đối tượng nghiên cứu một cách cẩn thận đã giúp chúng tôi xác định rõ mục tiêu và hướng phát triển.

Phương pháp nghiên cứu kết hợp với việc sử dụng các công nghệ tiên tiến như HTML, CSS, JavaScript, ASP.NET Core MVC và SQL Server đã tạo nên một nền tảng mạnh mẽ, linh hoạt và dễ quản lý. Điều này không chỉ giúp tối ưu hóa trải nghiệm mua sắm trực tuyến cho người dùng mà còn đảm bảo quản lý sản phẩm, đơn hàng và dữ liệu hiệu quả.

# CƠ SỞ LÍ THUYẾT

## HTML, CSS

Html là gì?

* HTML là từ viết tắt của Hypertext Markup Language, là sự kết hợp của Hypertext và Markup, hay còn được gọi là ngôn ngữ siêu văn bản. HTML có chức năng giúp người dùng xây dựng và cấu trúc các phần trong trang web hoặc ứng dụng, thường được sử dụng trong phân chia các đoạn văn, heading, link, blockquotes …
* Cụ thể HTML là một loại ngôn ngữ đánh dấu siêu văn bản, với mục đích cấu trúc thành các cấu trúc cơ bản của một trang web để website trở thành một hệ thống hoàn chỉnh. HTML dùng ngôn ngữ của mình để đánh dấu siêu văn bản, điều này sẽ giúp các văn bản trên trên website được chia bố cục rõ ràng, chia khung sườn các thành phần của trang web, tạo trang web thành một hệ thống hoàn chỉnh

Css là gì?

* **CSS – Cascading Style Sheet** là một [ngôn ngữ lập trình](https://mona.media/cac-ngon-ngu-lap-trinh-pho-bien-2017/) được thiết kế vô cùng đơn giản, dễ sử dụng. Với CSS hướng tới mục tiêu chính là giúp đơn giản hóa được quá trình tạo ra các [website](https://mona.media/website-la-gi/).
* Nhiệm vụ chính của CSS là thực hiện việc xử lý giao diện của một trang web cụ thể. Đó là những yếu tố như màu sắc văn bản, hay khoảng cách giữa các đoạn, hoặc kiểu [font chữ](https://mona.media/cac-font-chu-dung-cho-website/), hình ảnh, bố cục, màu nền, … đều có thể thay đổi, chỉnh sửa theo ý muốn với hỗ trợ của CSS.

## RESTful Api

### Giới thiệu

#### REST

REST, viết tắt của Representational State Transfer, là một kiến trúc cho việc thiết kế các hệ thống phân phối dữ liệu trên các môi trường mạng. Nó dựa trên các nguyên tắc đơn giản, truyền tải dữ liệu qua giao thức HTTP một cách nhẹ nhàng.

#### RESTful

RESTful đề cập đến việc thiết kế và triển khai các dịch vụ web (Web Services) sử dụng kiến trúc REST. Các ứng dụng RESTful thường được thiết kế để sử dụng các phương thức HTTP một cách rõ ràng và có thể mở rộng dễ dàng.

#### API

API là viết tắt của Application Programming Interface, là một tập hợp các quy tắc và công cụ cho việc xây dựng phần mềm. API cho phép các ứng dụng tương tác với nhau thông qua các giao diện được xác định trước.

Có mấy 3 loại api

**API WEB:** Dùng để tương tác qua mạng thông qua giao thức HTTP.

**API LIBRARY:** Bao gồm các thư viện và hàm để thực hiện các nhiệm vụ cụ thể.

**API OS-LEVEL:** Cho phép ứng dụng tương tác với hệ điều hành.

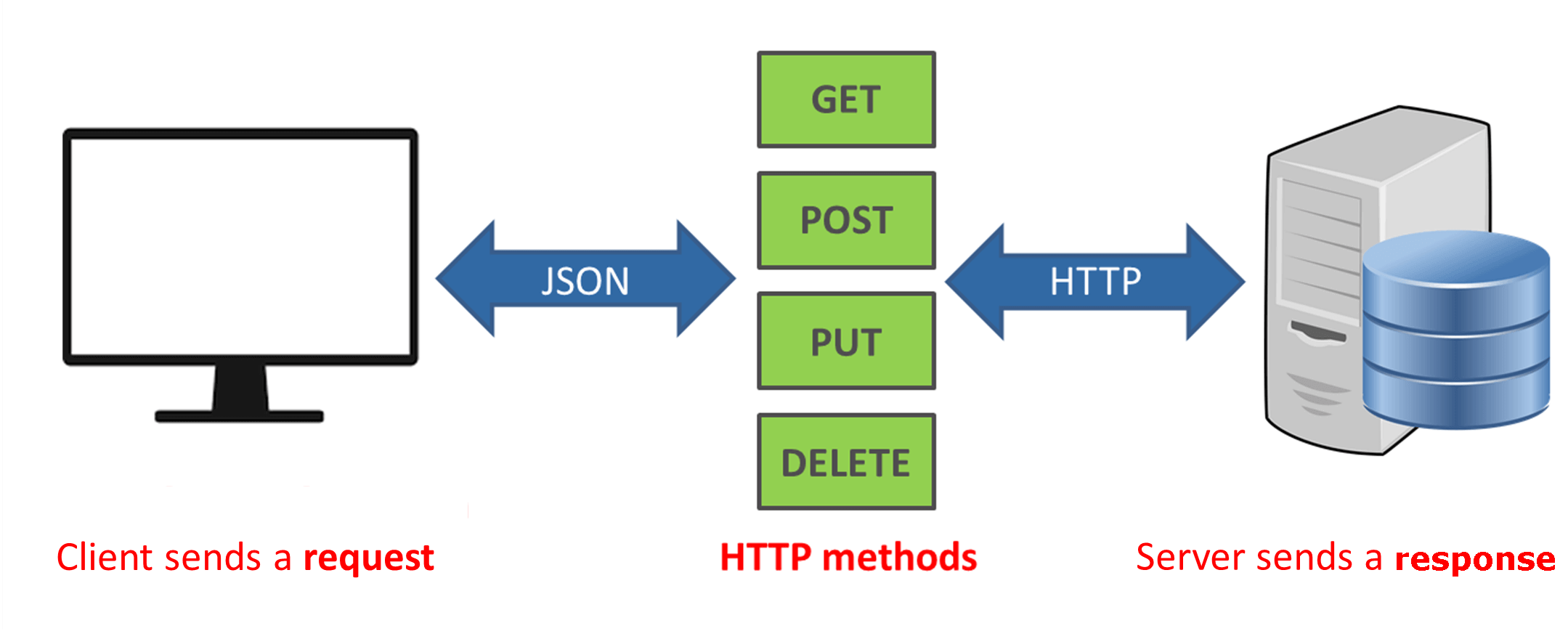
#### RESTful API

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

### Cách hoạt động

* REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.
* GET (SELECT): Trả về một Resource hoặc một danh sách Resource
* POST (CREATE): Tạo mới một Resource
* PUT (UPDATE): Cập nhật thông tin cho Resource
* DELETE (DELETE): Xoá một Resource

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.



Hình Cách hoạt động của RESTful API

## ASP.NET Core

### Giới thiệu về ASP.NET Core

ASP.NET là một nền tảng ứng dụng web (web application framework) được phát triển và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. Lần đầu tiên được đưa ra thị trường vào tháng 1 năm 2002 cùng với phiên bản 1.0 của.NET framework, là công nghệ nối tiếp của Microsoft's Active Server Pages (ASP). ASP.NET được biên dịch dưới dạng Common Language Runtime (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi.NET language.

ASP.NET Core là một nền tảng mã nguồn mở miễn phí kế thừa từ ASP.NET được phát triển bời Microsoft. Lần đầu tiên được được đưa ra thị trường vào tháng 6 năm 2016 cùng với phiên bản 1.0 của .NET Core.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (clound) hoặc chạy on-promise.

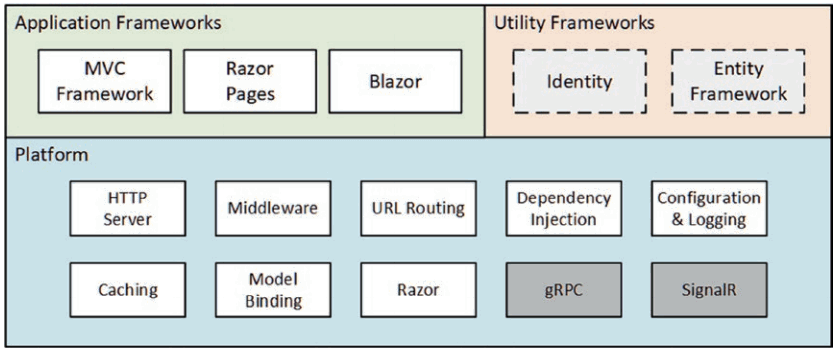
ASP.NET Core được sử dụng để phát triển website và có thể tương với nhiều trình duyệt khác nhau trên các hệ điều hành như Windows, Mac và Linux.

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **ASP.NET Core** | **ASP.NET** |
| **Lịch sử** | Phiên bản đầu tiên được giới thiệu vào tháng 6 năm 2016 cùng với bản .Net Core 1.0 | Ra đời vào tháng 1 năm 2002 cùng với bản .Net Framwork 1.0 |
| **Phát triển** | Microsoft | Microsoft |
| **Hệ điều hành** | Hỗ trợ trên các hệ điều hành Windows, macOS, hoặc Linux | Chỉ hỗ trợ Windows |
| **Công cụ phát triển** | Visual Studio, Visual Studio for Mac, và Visual Studio Code | Visual Studio |
| **Ngôn ngữ lập trình** | C#, F# | C#, VB, F# |
| **Các phiên bản đã phát hành** | 1.0, 1.1, 2.0, 2.1 long-term support, 2.2, 3.0, 3.1 long-term support, 5.0, 6.0 | 1.0, 1.1, 2.0, 3.0, 3.5, 3.5 Service Pack 1, 4.0, 5.0, 4.5.1, 4.6, 4.7, 4.7.1, 4.8 |

Bảng So sánh ASP.NET Core và ASP.NET

### Cấu trúc của ASP.NET Core

ASP.NET Core là một hệ thống phức tạp dành cho phát triển nhiều loại ứng dụng khác nhau hoạt động trên web. Do vậy, ASP.NET Core bao gồm nhiều thành phần với vai trò khác nhau nhưng tương tác trong một khuôn khổ chung.



Hình Cấu trúc của ASP.NET Core

#### Application Frameworks

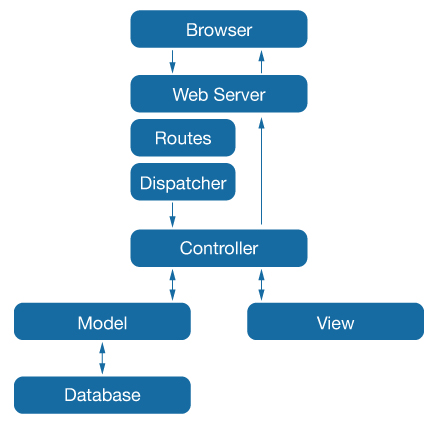
**MVC Framework:** xây dựng theo nguyên lý phân chia nhiệm vụ. MVC phân chia chức năng các thành phần của dứng dụng ra làm các nhóm độc lập, gọi là **M**odel, **V**iew và **C**ontroller. Trong ASP.NET Core, cùng với sự phổ biến của ứng dụng Single-Page, MVC không còn đóng vai trò một framework chính nữa.

MVC gồm 3 thành phần bao gồm:

* M là Model: nơi chứa nghiệp vụ tương tác với dữ liệu, nó bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm, xoá, sửa dữ liệu, …
* V là View: là nơi chứa giao diện, hiển thị dữ liệp giúp người dùng tương tác với hệ thống.
* C là Controller: là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó gồm các class/function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu cần thiết nhờ các nghiệp vụ ở lớp model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.

Sự tương tác giữa các thành phần:

* Controller tương tác với Model.
* Controller tương tác với View.
* Model và View không có sự tương tác với nhau mà nó tương tác thông qua Controller.



Hình Mô hình MVC

**Razor Pages:** một trong các framework dành cho xây dựng ứng dụng web bên trên ASP.NET Core. Razor Pages cho phép trộn HTML và C# (gọi là cú pháp Razor) vào cùng một file (có đuôi cshtml) để dễ dàng tạo ra HTML theo logic của chương trình. Nhờ vậy dữ liệu HTML tạo ra “động” chứ không cố định như ở các trang web “tĩnh” thiết kế sẵn.

**Blazor:** một framework giúp phát triển ứng dụng đơn trang (SPA – Single Page Application) nhưng sử dụng ngôn ngữ C#.

#### Utility Frameworks

Utility Frameworks chứa hai frameworks (không bắt buộc) nhưng lại được sử dụng gần như trong mọi ứng dụng ASP.NET Core.

**Entity Framework Core:** giúp ứng dụng tương tác với cơ sở dữ liệu. Entity Frameworks core giúp ánh xạ (hai chiều) giữa các bảng cơ sở dữ liệu với các domain model.

**Identity:** là framework dành cho xác thực (authentication) và xác minh quyền (authorization) người dùng trong ứng dụng.

#### Platform

Phần Platform chứa nhiều thành phần cần thiết cho việc nhận xử lý truy vấn HTTP, cũng như tạo ra các phản hồi phù hợp.

**HTTP Server:** còn gọi là built-in server với tên gọi Kestrel, có nhiệm vụ tiếp nhận truy vấn HTTP. Kestrel có thể hoạt động độc lập (tích hợp trong một ứng dụng khác) hoặc phối hợp với một web server thông thường (Apache, Nginx, IIS). Kestrel là một HTTP web server mã nguồn mở (open source), đa nền tảng (cross-platform), hướng sự kiện (event-driven) và bất đồng bộ (asynchronous I/O). Nó được phát triển để chạy ứng dụng ASP.NET Core trên bất cứ nền tảng nào. Nó được thêm vào mặc định trong ứng dụng ASP.NET Core.

**Middleware:** Một Middleware là một module code nó nhận yêu cầu gửi đến Request và trả về Response. Cụ thể trong ASP.NET Core, middleware có thể:

* Nhận một HTTP Request gửi đến và phát sinh ra HTTP Response để trả về.
* Nhận một HTTP Request gửi đến, thi hành một số tác vụ (có thể là sửa đổi HTTP Request), sau đó chuyển đến một middleware khác.
* Nhận HTTP Response, sửa nó và chuyển đến một Middleware khác.

**URL Routing:** cơ chế ánh xạ chuỗi truy vấn HTTP sang thực thi một phương thức nào đó. Do vậy, mỗi URL sẽ tương ứng với thực thi một phương thức trên server.

**Dependency Injection:** cơ chế cho phép tự động sinh và chèn object vào một object khác. ASP.NET Core xây dựng sẵn cơ chế này mà không cần đến một thư viện thứ ba.

**Configuration & Logging:** là cơ chế hỗ trợ cấu hình và lưu vết quá trình thực thi ứng dụng.

**Caching:** là cơ chế lưu tạm để tăng hiệu suất cho ứng dụng.

**Model Binding:** Trong ứng dụng ASP.NET Core thì các trang Razor Page hoặc các Controller (MVC) cần các dữ liệu gửi bởi truy vấn, từ các thông tin gửi đến như chuỗi query của url, tham số route, dữ liệu post của Form ... ứng dụng có thể đọc được những dữ liệu này, ta có thể hoàn toàn trích xuất được những dữ liệu này bằng cách sử dụng các phương thức cung cấp bởi đối tượng HttpRequest (thuộc tính Request của Controller hoặc của PageModel).

ASP.NET Core cung cấp một cơ chế giúp bạn tự động quá trình lấy dữ liệu trên gọi là Model Biding, nó hoạt động tự động qua mấy bước chính:

* Lấy các dữ liệu từ các nguồn gửi đến (Route data, Form data, chuỗi Query, File tải lên).
* Tiến hành chuyển và thiết lập dữ liệu đó cho Razor Page hoặc Controller.
* Chuyển đổi chuỗi dữ liệu nhận được thành các kiểu phù hợp với .NET.

**Razor:** cơ chế sinh ra HTML từ dữ liệu và logic của hương trình. Razor được gọi là view engine trong ASP.NET Core. Razor sử dụng loại cú pháp đặc biệt kết hợp giữ C# và HTML.

**gRPC:** là một chuẩn dành cho gọi hàm từ xa (Remote Procedure Call) đa nền tảng qua HTTP do Google (chữ g trong gRPC) phát triển. gRPC có thể hữu ích để các thành phần backend của ứng dụng (chạy trên server) trao đổi dữ liệu.

**SignalR:** làm một framework riêng nhưng lại được sử dụng làm nền tảng cho ứng dụng ASP.NET Core. SignalR cho phép tạo ra kênh truyền thông tốc độ cao (theo thời gian thực) và hai chiều giữa browser và web server. SignalR là nền tảng cho Blazor Server – loại ứng dụng đơn trang mà toàn bộ phần xử lý được đẩy về server và nhận lại kết quả (DOM) theo thời gian thực.

**gRPC** và **SignalR** hiếm khi phải sử dụng đến, trừ trong những tình huống đặc biệt. Chúng được đưa vào đây chủ yếu vì đóng vai trò nền tảng cho một số trường hợp đặc biệt.

## SQLServer

### Giới thiệu SQLServer

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ được phát triển bởi Microsoft. Là một máy chủ cơ sở dữ liệu, nó là một sản phẩm phần mềm có chức năng chính là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm khác. Có thể chạy trên cùng một máy tính hoặc trên một máy tính khác trên mạng (bao gồm cả Internet).

Phần mềm được Microsoft phát triển dựa trên RDBMS. Đây cũng là một hệ quản trị cơ sở dữ liệu quan hệ đối tượng (ORDBMS). SQLServer có thể sử dụng giao diện dòng lệnh và giao diện GUI.

SQLServer hỗ trợ ngôn ngữ SQL (trước đây là SEQUEL – ngôn ngữ truy vấn tiếng anh có cấu trúc) – vốn là sản phẩm của IBM.

### Các thành phần của SQLServer

SQLServer hoạt động theo mô hình khách – chủ, do đó sẽ có 2 thành phần là Workstation và Server.

Workstation được cài trên bất kì thiết bị nào hoặc trên thiết bị của người vận hành máy chủ. Đây là các giao diện phần mềm để tương tác với Server, ví dụ như SSMS, SSCM, Prifiler, BIDS, SQLEM, …

Server được cài trên máy chủ tập trung. Nó chính là các dịch vụ như SQL Server, SQL Server Agent, SSIS, SSAS, SSRS, SQL Browser, SQL Full Text Search, …

## React JS

### Giới thiệu về React JS

React (Hay ReactJS, React.js) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi Jordan Walke, một kỹ sư phần mềm tại Facebook. Người bị ảnh hưởng bởi XHP (Một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, sau đó được triển khai cho Instagram.com năm 2012. Nó được mở mã nguồn (open-sourced) tại JSConf US tháng 5 năm 2013.

React được sử dụng rộng rãi và có hệ sinh thái đa dạng phong phú. UI tất nhiên là quan trọng, nhưng không phải là tất cả. Để phát triển ứng dụng hoàn chỉnh, một mình React Js là không làm được tất cả, ReactJS sẽ cần kết hợp thêm:

* HTML/CSS: nếu làm ứng dụng web.
* Flux/Redux: là một kiến trúc giúp tổ chức code rõ ràng và sạch sẽ.

Nó được sử dụng bởi các công ty lớn, đã thành lập và các công ty mới thành lập như: Netflix, Airbnb, Instagram và New York Times, ... ReactJS mang lại nhiều lợi thế cho các lập trình viên, khiến nó trở thành một lựa chọn tốt hơn so với các Framework khác như Angular.

### Các thành phần của React JS

#### JSX

Trong React, thay vì thường xuyên sử dụng JavaScript để thiết kế bố cục trang web thì sẽ dùng JSX. JSX được đánh giá là sử dụng đơn giản hơn JavaScript và cho phép trích dẫn HTML cũng như việc sử dụng các cú pháp thẻ HTML để render các subcomponent. JSX tối ưu hóa code khi biên soạn, vì vậy nó chạy nhanh hơn so với code JavaScript tương đương.

Các đoạn mã được return trong component trông sẽ giống HTML nhưng thực chất đó là một đoạn jsx, jsx cho phép chúng ta kết nối giữa HTML và JavaScript trong cùng một source.

#### Component

Hoạt động giống như các hàm và trả về các thành phần HTML.

Các component là thành phần độc lập và có thể tái sử dụng lại nhiều lần.

Các component thực hiện công việc giống như các function trong JavaScript nhưng chúng độc lập và nhiệm vụ chính là trả về HTML thông qua hàm render.

Có 2 loại component:

* Stateless Component (functional component): thực chất là 1 function có nhiệm vụ return về một đoạn mã HTML hiển thị ra giao diện. Stateless không thể sử dụng được state và component lifecycle.
* Stateful Component (class component): là một class có chứa phương thức render. Khi component được gọi, hàm render sẽ chạy và trả về nội dung HTML bên trong. Statefull sử dụng được cả state và lifecycle.

#### State

Trong React, State được dùng để biểu diễn trạng thái của Component. Tức là giá trị của State có thể thay đổi được, trái ngược hoàn toàn với Props là 1 giá trị bất biến.

#### Props

Props là thuộc tính mặc định của component để nhận dữ liệu từ component cha truyền từ ngoài vào và nó không thể bị chỉnh sửa bên trong component con.

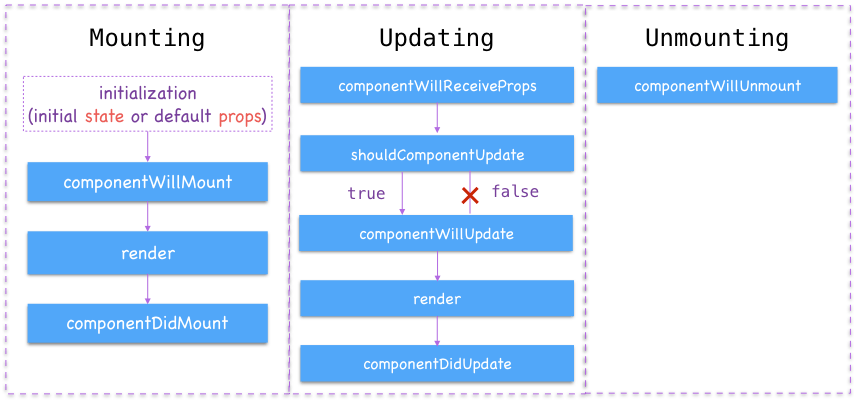
#### React Lifecycle (Vòng đời component)

Các phương thức trong React Component Lifecycle có thể chia ra làm 3 pha chính là: Mounting, Updating và UnMounting.

**Mounting (hay còn gọi là Creation):** bao gồm các phương thức sẽ được khởi chạy lần lượt khi component được khởi tạo và chỉ sử dụng được với stateful.

**Updating:** Updating là giai đoạn khi React Component cần cập nhật giao diện mỗi khi props hoặc state của nó thay đổi.

**UnMounting:** Giai đoạn này sẽ clear tất cả những thứ liên quan đến component này trước khi remove nó.



Hình Vòng đời component

## TailwindCSS

Tailwind css là một utility-first CSS framework nó hỗ trợ phát triển xây dựng nhanh chóng giao diện người dùng, nó cũng có điểm chung giống như Bootstrap & điểm làm nó nổi bật hơn cả đó là chúng ta có thể tùy biến phát triển css theo cách mà chúng ta định nghĩa ra.

Tailwind CSS đã dựng sẵn các class CSS và cả các class responsive và nhiều tiện ích khác mà chúng ta có thể sử dụng dễ dàng vào HTML và không cần viết một dòng CSS nào. Ngược lại, cách làm này lại không có chúng ta những class dựng sẵn theo kiểu của Bootstrap (đã dựng sẵn các thành phần), mà mọi thứ chúng ta bắt buộc phải tự dựng hoàn toàn bằng các class tiện ích này.

**Ưu điểm:**

* Viết CSS custom cực kì ít.
* Không cần viết 1 dòng css nào mà chỉ cần thêm class để tạo giao diện, không cần phải đau đầu suy nghĩ tên class để đặt cho các div.
* Các tên class của tailwind rất khoa học, dễ hiểu và dễ sử dụng, mỗi class là một thuộc tính.
* TailwindCSS giúp cải thiện hiệu suất vì giảm thiểu các việc trùng lặp thuộc tính.
* Có nhiều plugin hỗ trợ loại bỏ các class thừa không sử dụng, giảm thiểu việc có mặt trong việc khai báo ở class HTML.

**Nhược điểm:**

* Số lượng class cho mỗi tag có thể rất nhiều.
* Phải vững kiến thức về CSS.

Không đủ hết 100% thuộc tính CSS.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Thực trạng bài toán

Trong phần này, chúng ta sẽ phân tích cụ thể về tình hình hiện tại của bài toán trong quá trình thiết kế hệ thống web bán điện thoại. Điều này bao gồm nhiều yếu tố quan trọng như nhu cầu thị trường, xu hướng mua sắm trực tuyến và các vấn đề kỹ thuật.

#### Nhu cầu thị trường và xu hướng mua sắm điện thoại trực tuyến

- Nhu cầu người dung Tình hình thị trường hiện tại cho thấy nhu cầu mua sắm điện thoại trực tuyến ngày càng tăng, đặc biệt là trong việc tìm kiếm sự thuận tiện và lựa chọn đa dạng.

- Xu hướng mua sắm Người dùng ngày càng chú trọng đến trải nghiệm mua sắm, bao gồm giao diện thân thiện, đánh giá sản phẩm, và dịch vụ hỗ trợ sau bán hàng.

#### Thách thức kỹ thuật và quản lý thông tin

- Bảo mật thông tin: Các vấn đề về bảo mật dữ liệu và thông tin cá nhân của người dùng là một trong những thách thức quan trọng nhất.

- Quản lý dữ liệu: Việc quản lý hàng tồn kho, thông tin sản phẩm và quản lý đơn hàng một cách hiệu quả là một thách thức kỹ thuật cần được giải quyết.

#### Cơ hội và tiềm năng

- Tăng cường trải nghiệm người dùng: Phát triển một giao diện người dùng tối ưu và cung cấp trải nghiệm mua sắm trực tuyến tốt hơn có thể là cơ hội để thu hút và giữ chân khách hàng.

- Mở rộng thị trường: Khả năng tiếp cận và mở rộng thị trường từ mua sắm trực tuyến có thể tạo ra cơ hội lớn để tăng doanh số bán hàng.

Phần này cung cấp một cái nhìn tổng quan về tình hình hiện tại của bài toán, nhấn mạnh vào nhu cầu thị trường, thách thức kỹ thuật và cơ hội mà dự án xây dựng website bán điện thoại đang đối diện.

## Phân tích

### Danh sách các actor

Admin: là người quản trị web, admin đăng nhập tài khoản có toàn quyền sử dụng hệ thống.

Customer: là khách hàng thành viên đăng ký đã có tài khoản và mật khẩu để giao dịch.

Guest: là những người chỉ xem sản phẩm trong danh mục nhưng chưa mua, có thể xem thông tin sản phẩm và tin tức. Free User muốn mua hàng phải đăng ký thành viên.

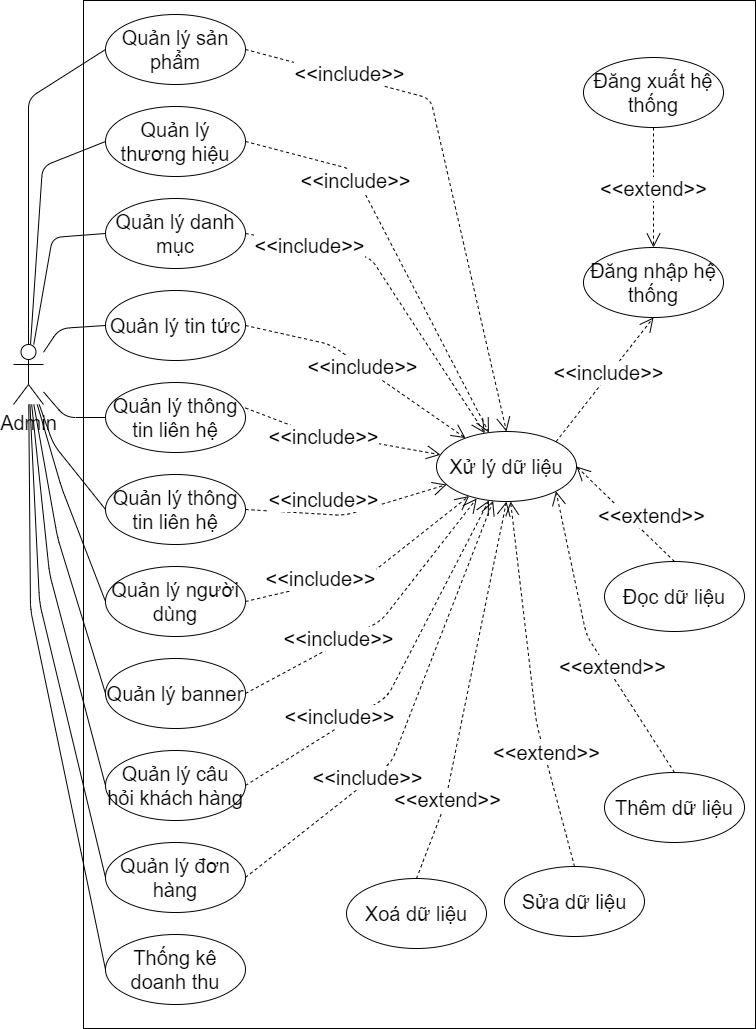
## 3.2.2. Danh sách các use case

|  |  |  |
| --- | --- | --- |
| ID | Tên Use Case | Mô tả |
| UC1 | Đăng nhập | Khách hàng đăng nhập vào website. |
| UC2 | Đăng ký | Tác nhân Guest muốn đăng ký tài khoản để đặt hàng. |
| UC3 | Đăng xuất | Khách hàng đăng xuất khỏi hệ thống |
| UC4 | Đặt hàng | Khách hàng đặt hàng |
| UC5 | Thêm sản phẩm vào giỏ hàng | Khách hàng thêm sản phẩm vào giỏ hàng |
| UC6 | Cập nhật giỏ hàng | Khách hàng cập nhật giỏ hàng |
| UC7 | Xoá giỏ hàng | Khách hàng xoá sản phẩm khỏi giỏ hàng |
| UC8 | Xem sản phẩm | Khách hàng xem sản phẩm |
| UC9 | Tìm kiếm sản phẩm | Khách hàng tìm kiếm sản phẩm theo chủ đề |
| UC10 | Quản lý sản phẩm | Quản trị quản lý sản phẩm |
| UC11 | Quản lý thương hiệu | Quản trị quản lý thương hiệu |
| UC12 | Quản lý danh mục | Quản trị quản lý danh mục |
| UC13 | Quản lý banner | Quản trị quản lý banner |
| UC14 | Quản lý đơn hàng | Quản trị quản lý đơn đặt hàng |
| UC15 | Quản lý người dùng | Quản trị quản lý người dùng |

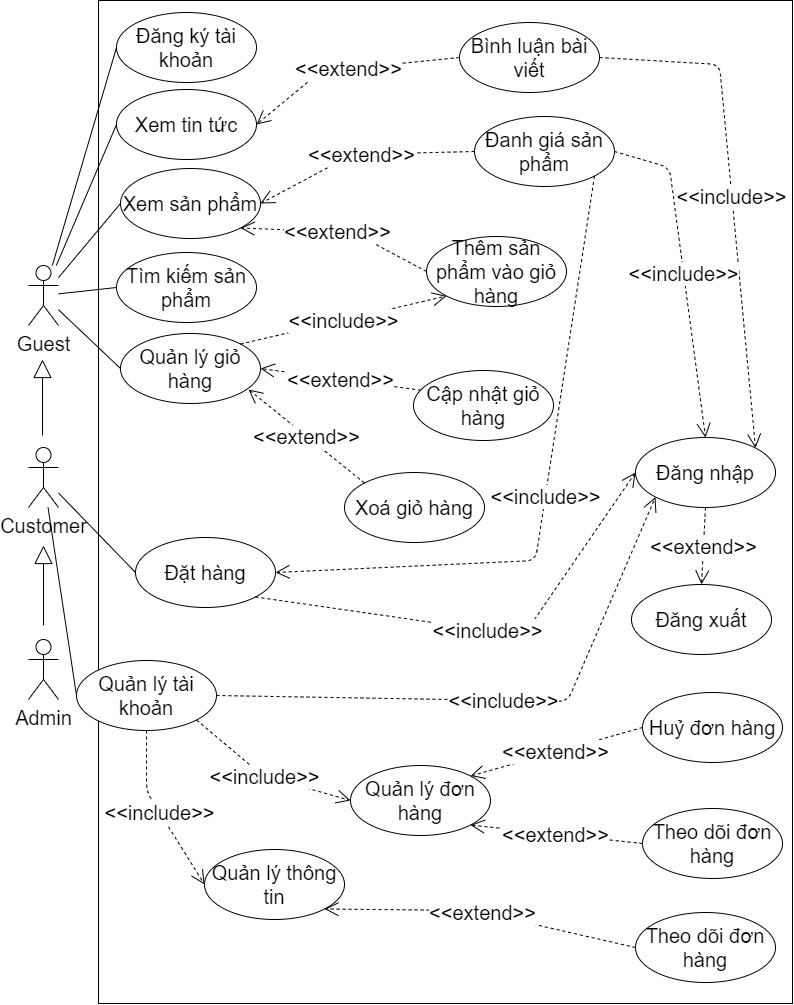
Bảng Danh sách use case

### Biểu đồ use case

* **Biểu đồ use case sử dụng cho quản trị**

Hình Biểu đồ use case sử dụng cho quản trị

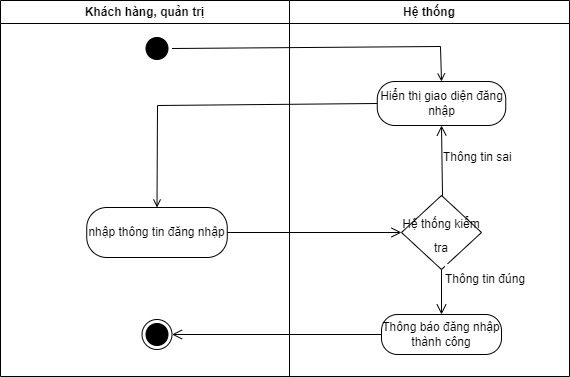
* **Vẽ biểu đồ use cas cho trang khách hang**

Hình Biểu đồ use case sử dụng cho khách hàng 

### Biểu đồ hoạt động

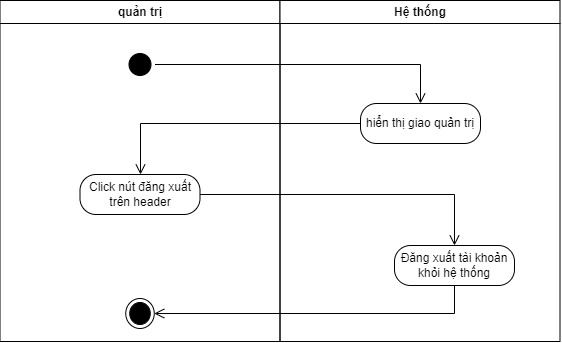
* **Vẽ biểu đồ use cas cho trang quản trị**

#### Đăng nhập



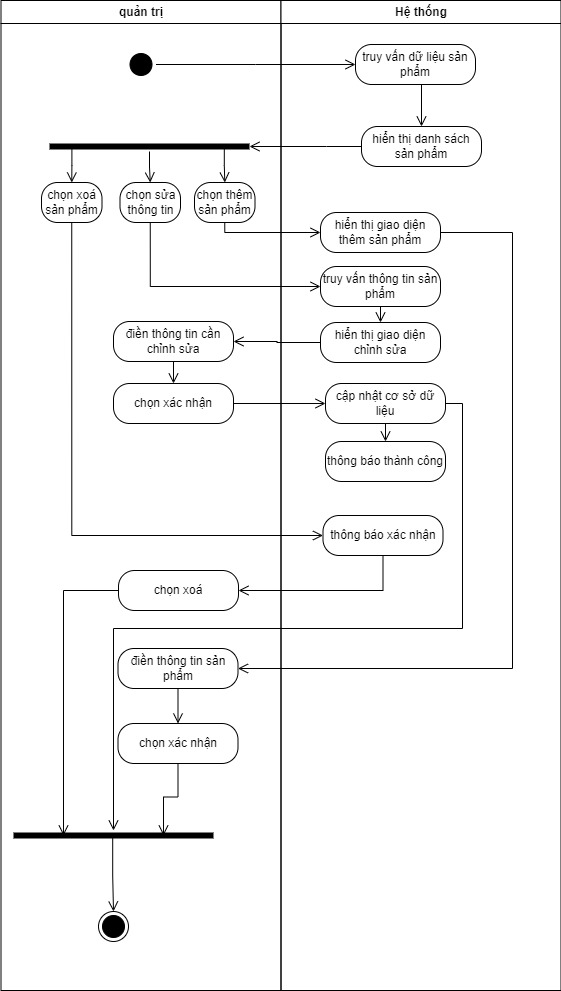
Hình 7 Biểu đồ đăng nhập của quản trị

#### Đăng xuất



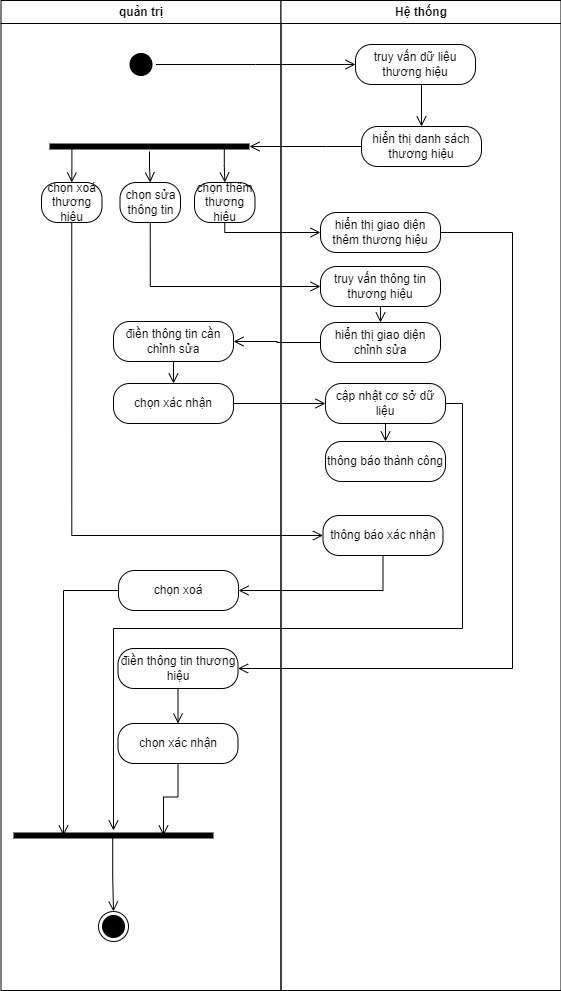
Hình Biểu đồ đăng xuất của quản trị

#### Biểu đồ hoạt động quản lý sản phẩm



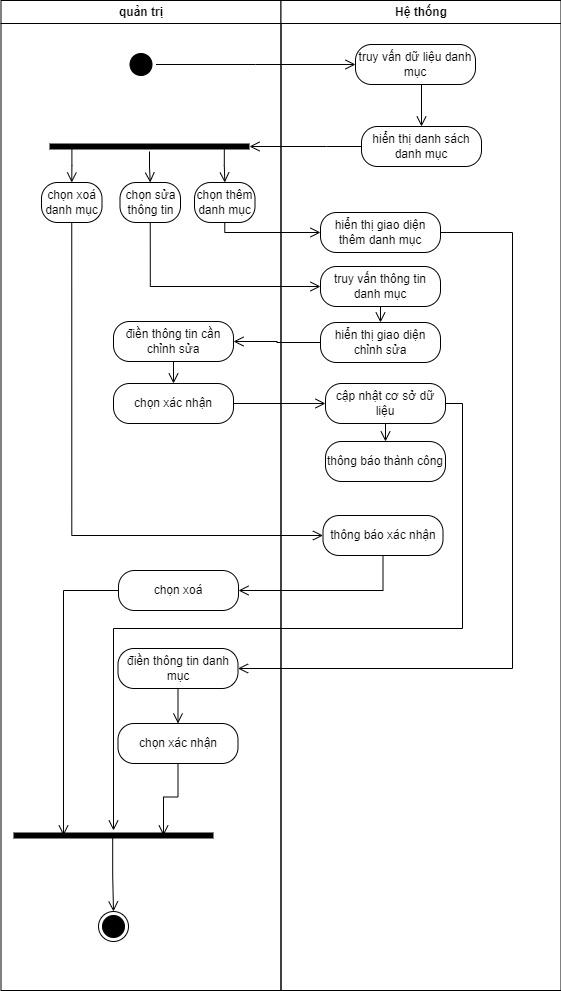
Hình Biểu đồ hoạt động quản lý sản phẩm

#### Biểu đồ hoạt động quản lý thương hiệu



Hình Biểu đồ hoạt động quản lý thương hiệu

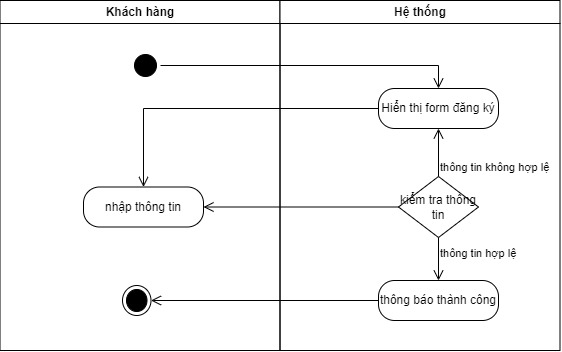
#### Biểu đồ hoạt động quản lý danh mục



Hình Biểu đồ hoạt động quản lý danh mục

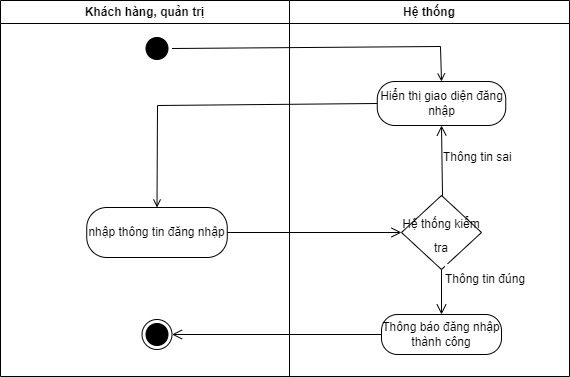
* **Vẽ biểu đồ use cas cho trang khách hang**

#### Đăng ký tài khoản



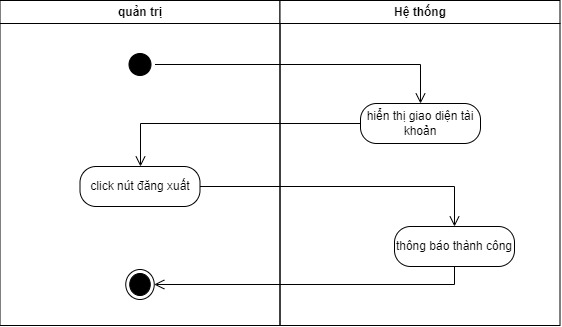
Hình 12 Biểu đồ hoạt động đăng ký tài khoản

#### Đăng nhập



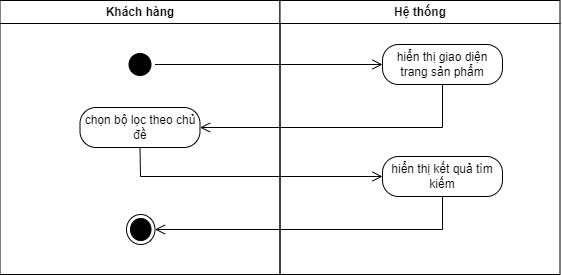
Hình 13 Biểu đồ hoạt động đăng nhập của khách hàng

#### Đăng xuất



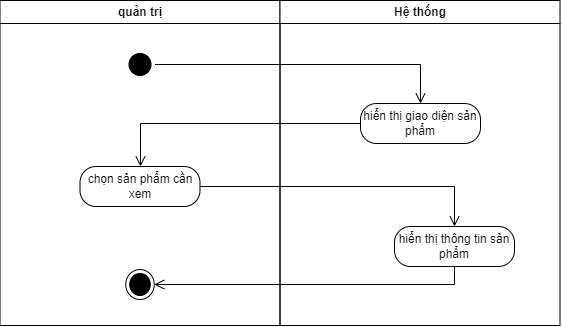
Hình 14 Biểu đồ hoạt động đăng xuất tài khoản của khách hàng

#### Tìm kiếm sản phẩm



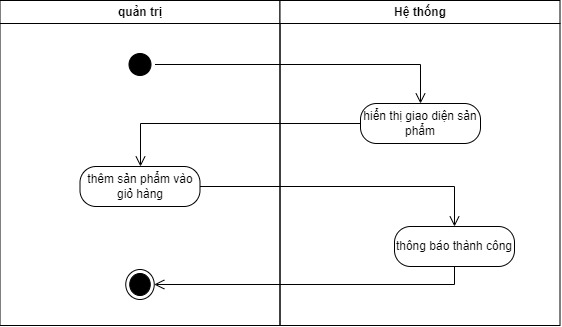
Hình Biểu đồ hoạt động tìm kiếm sản phẩm

#### Xem thông tin sản phẩm



Hình Biểu đồ hoạt động xem thông tin sản phẩm

#### Thêm sản phẩm vào giỏ hàng



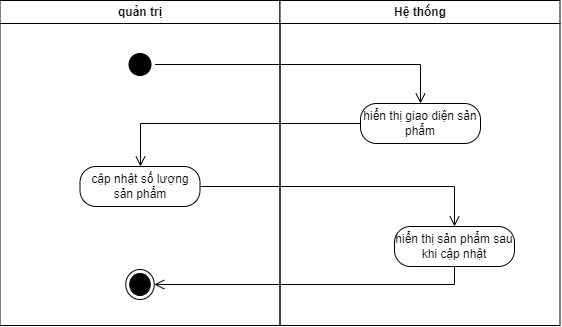
Hình Biểu đồ hoạt động thêm sản phẩm vào giỏ hàng

#### Xoá sản phẩm khỏi giỏ hàng



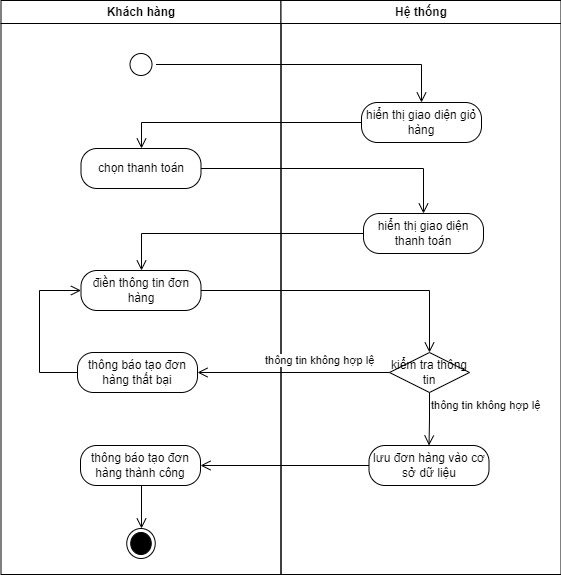
Hình Biểu đồ hoạt động xoá sản phẩm vào giỏ hàng

#### Cập nhật giỏ hàng



Hình Biểu đồ hoạt động cập nhật giỏ hàng

#### Đặt hàng

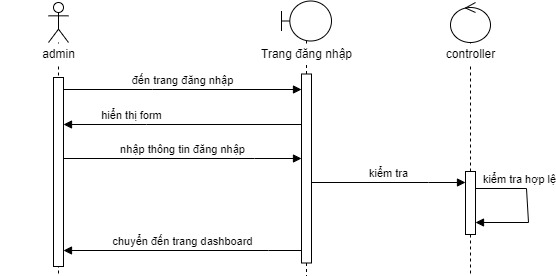


Hình 20 Biểu đồ hoạt động đặt hàng

### Biểu đồ tuần tự

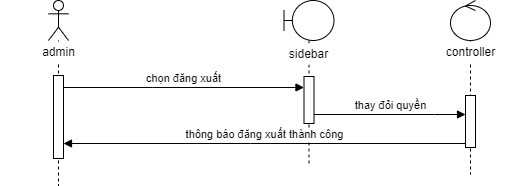
* Vẽ biểu đồ use cas cho trang quản trị

#### Đăng nhập



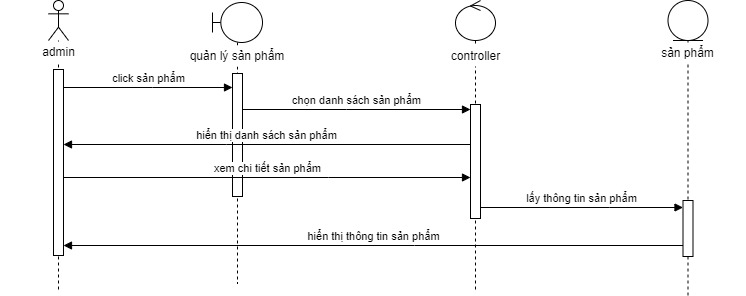
Hình 21 Biểu đồ tuần tự đăng nhập của quản trị

#### Đăng xuất



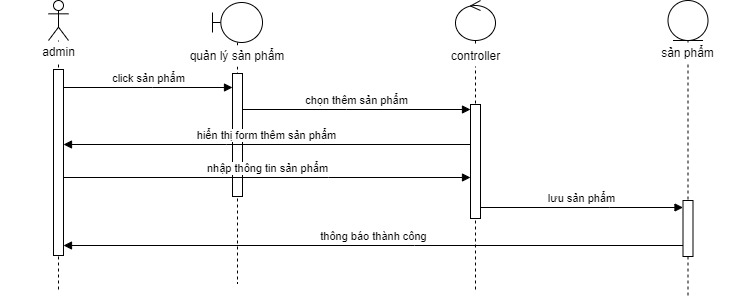
Hình 22 Biểu đồ tuần tự đăng xuất của quản trị

#### Xem thông tin sản phẩm



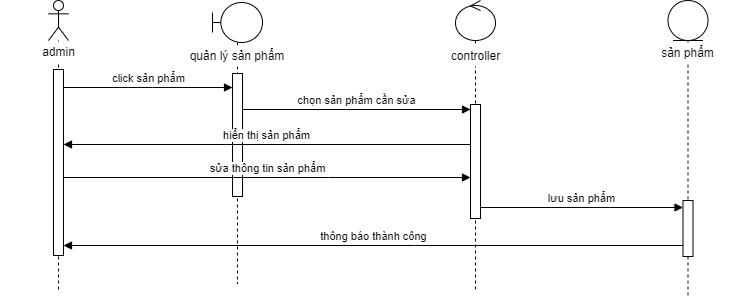
Hình 23 Biểu đồ tuần tự xem thông tin sản phẩm của quản trị

#### Thêm mới sản phẩm



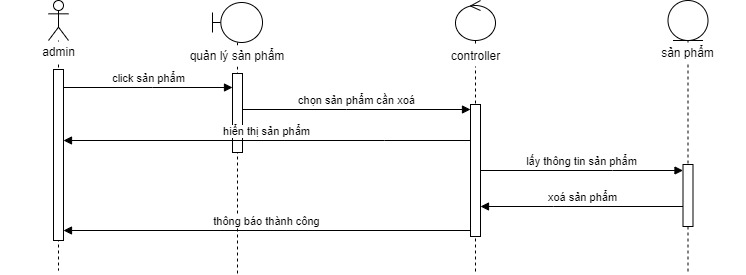
Hình 24Biểu đồ tuần tự thêm sản phẩm

#### Cập nhật sản phẩm



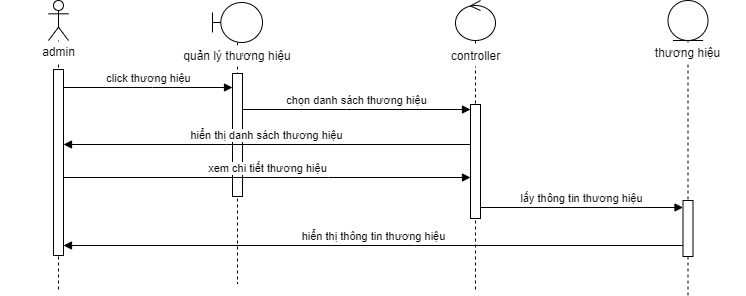
Hình 25 Biểu đồ tuần tự cập nhật sản phẩm

#### Xoá sản phẩm



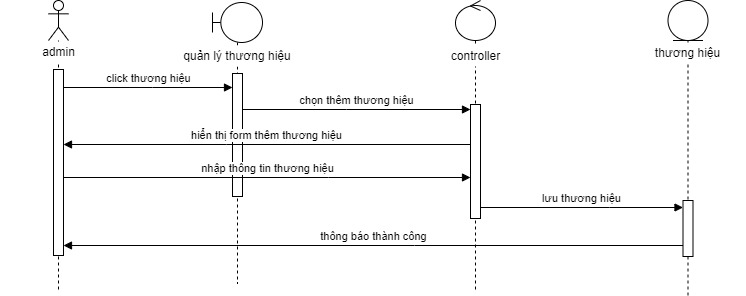
Hình 26 Biểu đồ tuần tự xoá sản phẩm

#### Xem thông tin thương hiệu



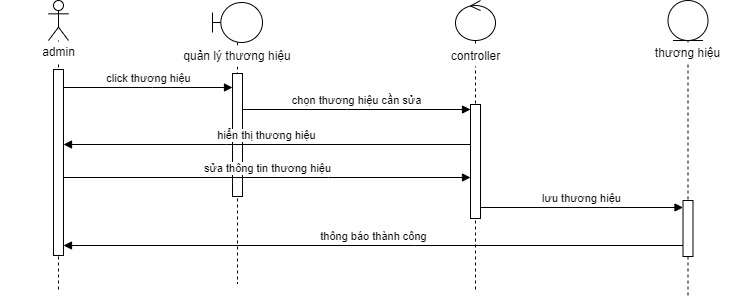
Hình 27 Biểu đồ tuần tự xem thông tin thương hiệu

#### Thêm mới thương hiệu



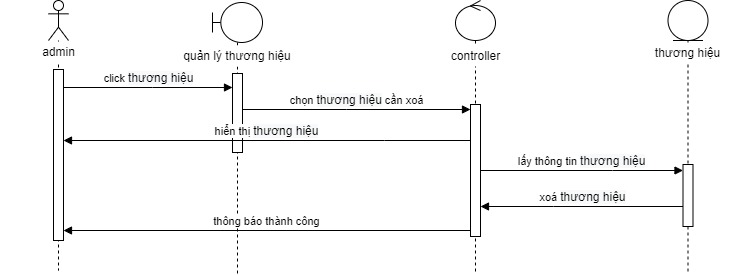
Hình 28 Biểu đồ tuần tự thêm thương hiệu

#### Cập nhật thương hiệu



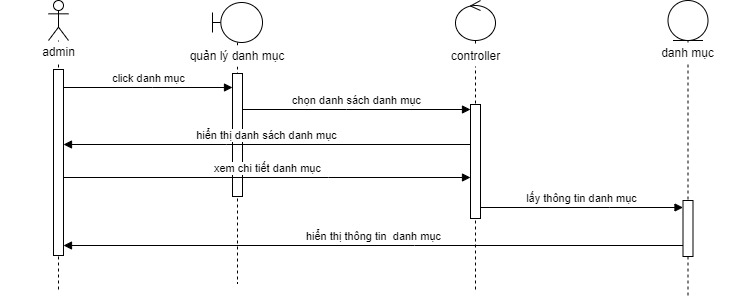
Hình 29 Biểu đồ tuần tự cập nhật thương hiệu

#### Xoá thương hiệu



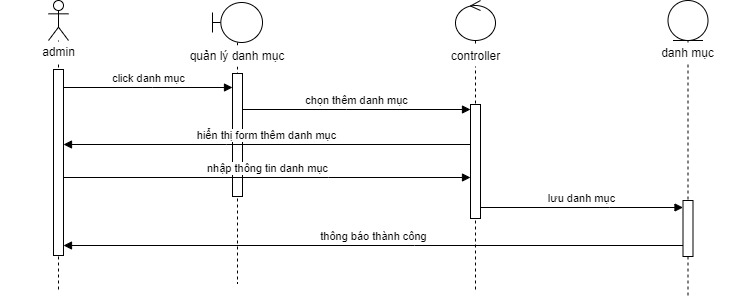
Hình 30 Biểu đồ tuần tự xoá thương hiệu

#### Xem thông tin danh mục



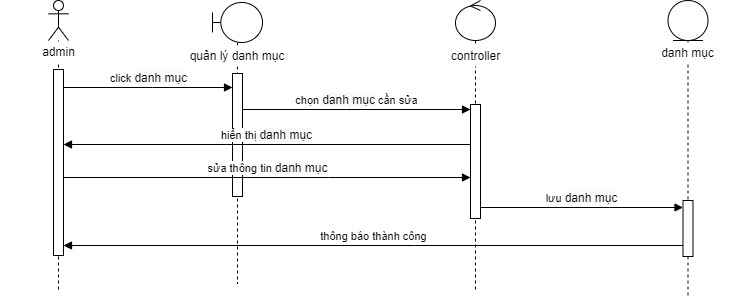
Hình 31 Biểu đồ tuần tự xem thông tin danh mục

#### Thêm danh mục



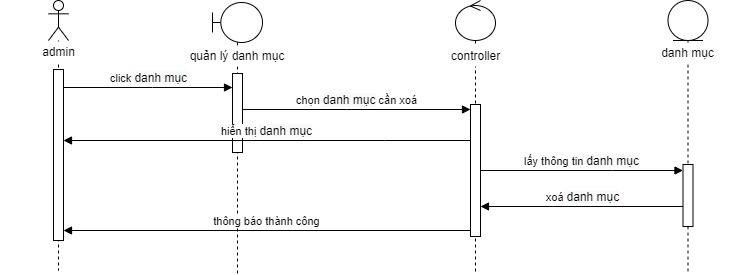
Hình 32 Biểu đồ tuần tự thêm danh mục

#### Sửa danh mục



Hình 33 Biểu đồ tuần tự sửa danh mục

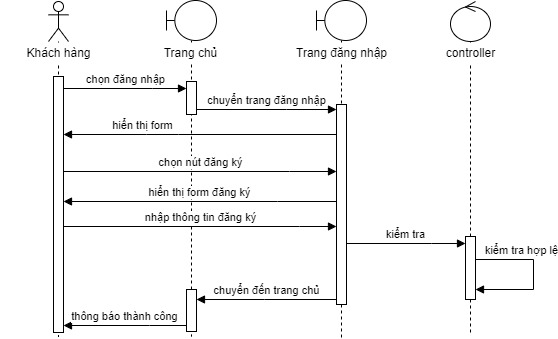
#### Xoá danh mục



Hình 34 Biểu đồ tuần tự xoá danh mục

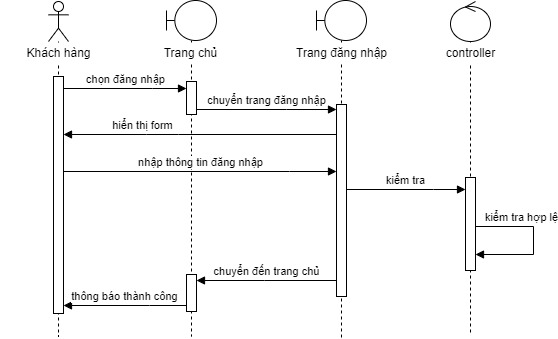
* **Vẽ biểu đồ use cas cho trang khách hang**

#### Đăng ký tài khoản



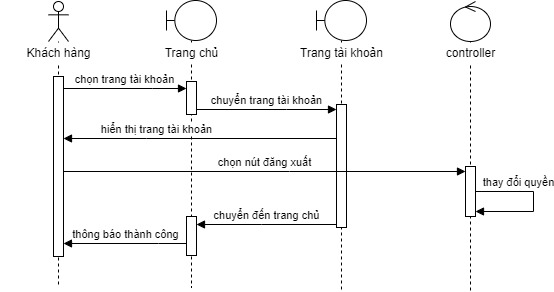
Hình 35 Biểu đồ tuần tự đăng ký tài khoản

#### Đăng nhập



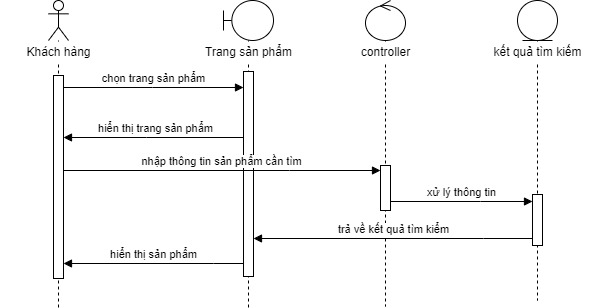
Hình 36 Biểu đồ tuần tự đăng nhập của khách hàng

#### Đăng xuất



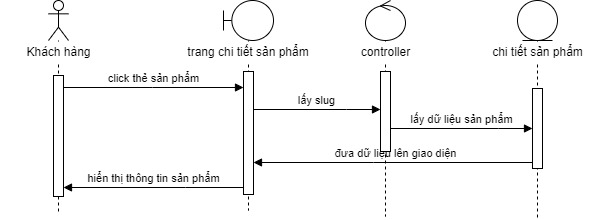
Hình 37 Biểu đồ tuần tự đăng xuất của khách hàng

#### Tìm kiếm sản phẩm



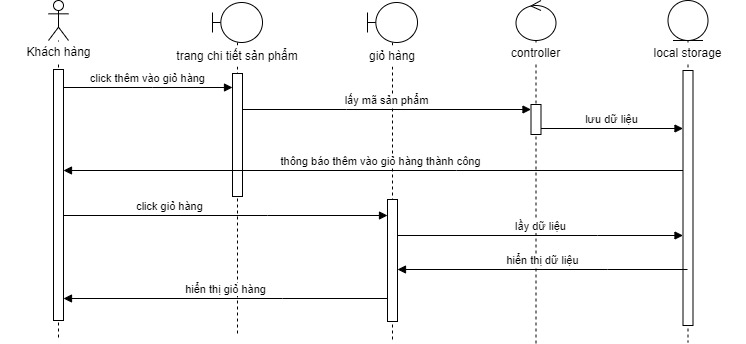
Hình 38 Biểu đồ tuần tự tìm kiếm sản phẩm

#### Xem thông tin sản phẩm



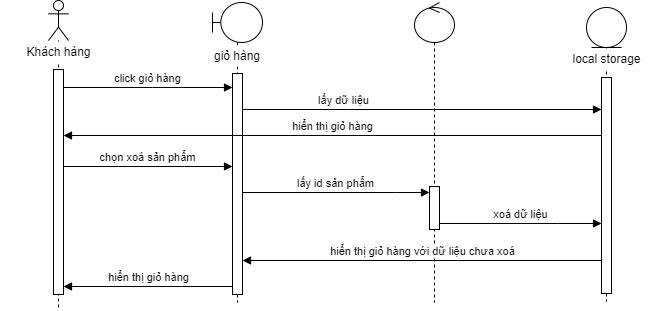
Hình 39 Biểu đồ tuần tự xem thông tin sản phẩm

#### Thêm sản phẩm vào giỏ hàng



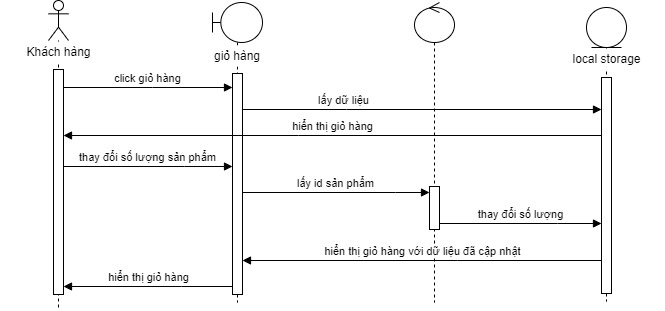
Hình 40 Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng

#### Xoá sản phẩm khỏi giỏ hàng



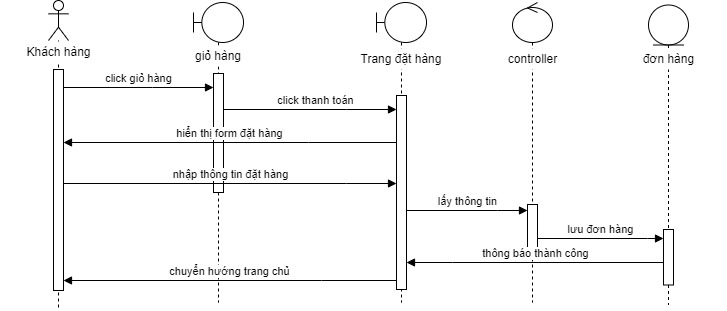
Hình 41 Biểu đồ tuần tự xoá sản phẩm khỏi giỏ hàng

#### Cập nhật giỏ hàng



Hình 42 Biểu đồ tuần tự cập nhật giỏ hàng

#### Đặt hàng

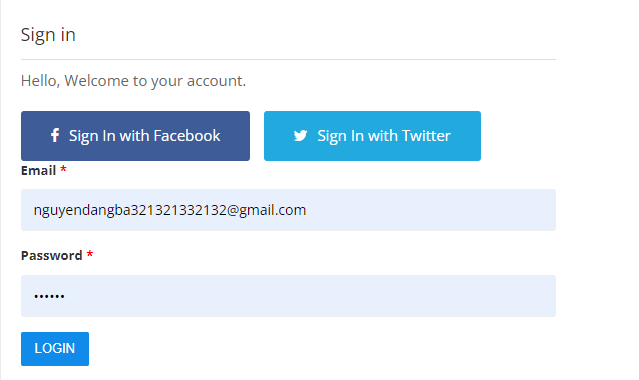


Hình 43 Biểu đồ tuần tự đặt hàng

## Giao diện

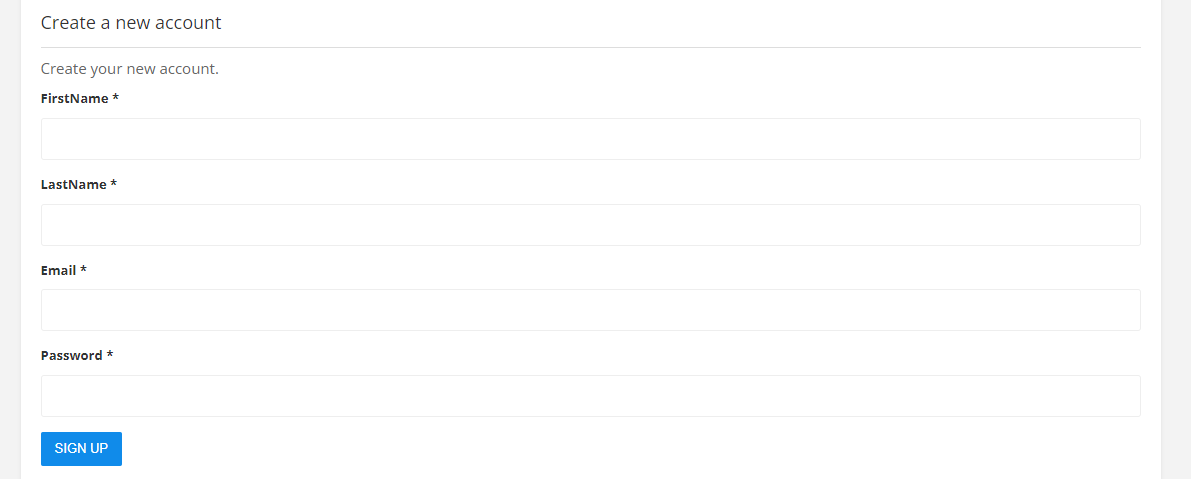
### Phía khách hàng

#### Giao diện đăng nhập



Hình 44 Giao diện đăng nhập của khách hàng

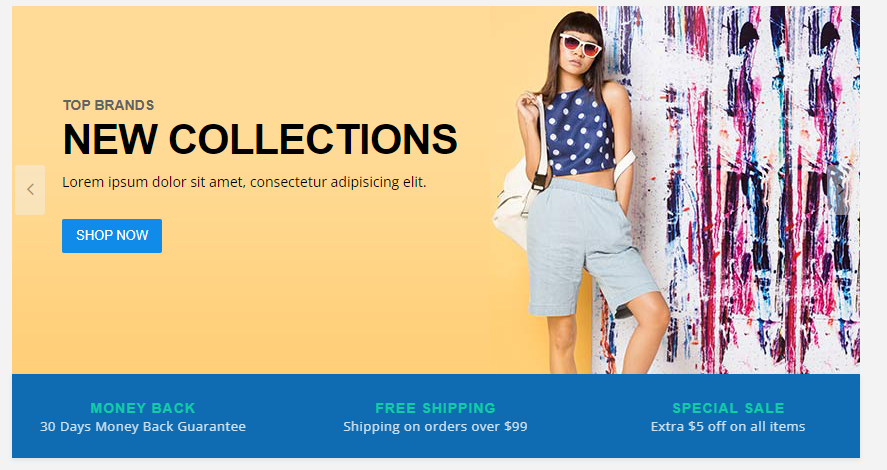
#### Giao diện đăng ký



Hình 45 Giao diện đăng ký

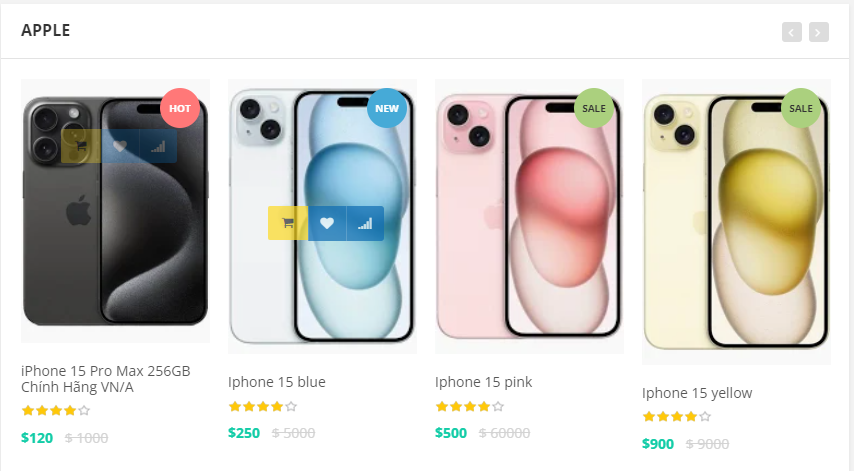
#### Giao diện trang chủ

**Phần Hero Banner:**



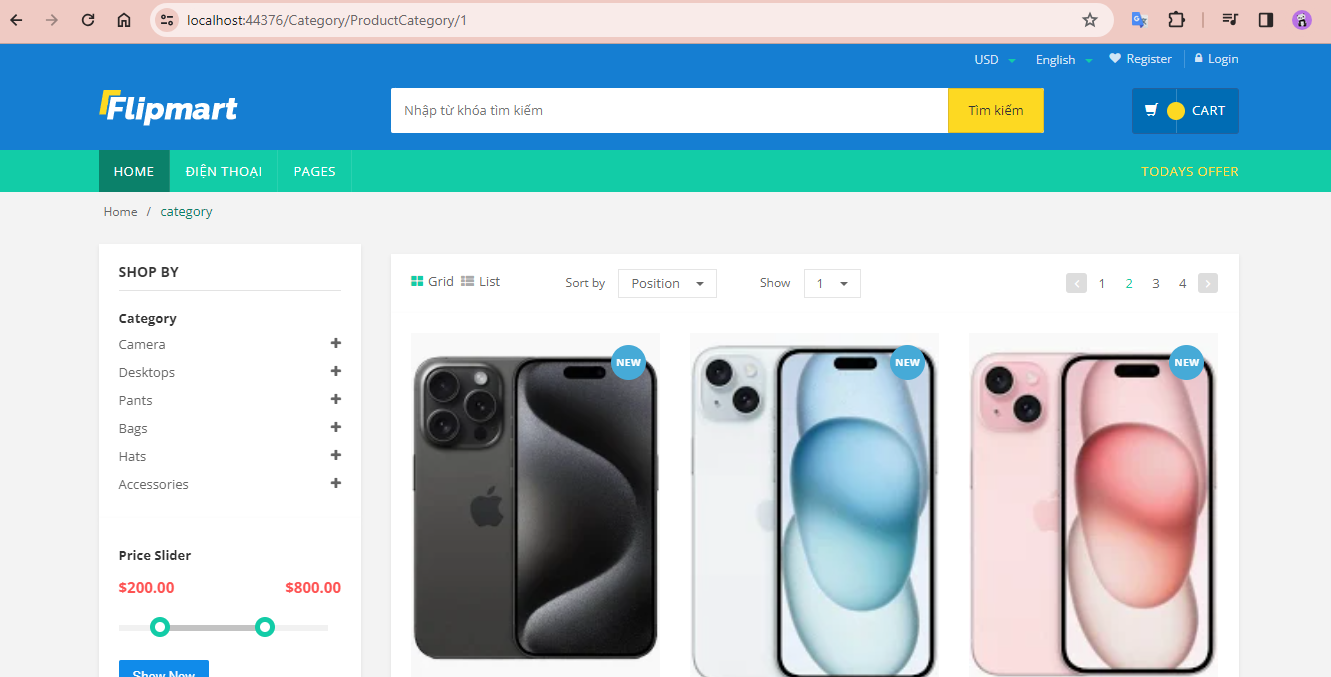
Hình 46 Phần hero banner

**Sản phẩm theo thương hiệu**



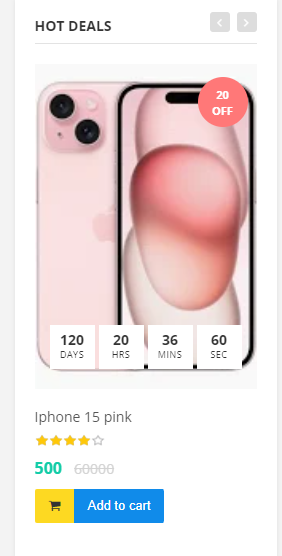
Hình 47 Sản phẩm theo thương hiệu

**Sản phẩm theo danh mục**



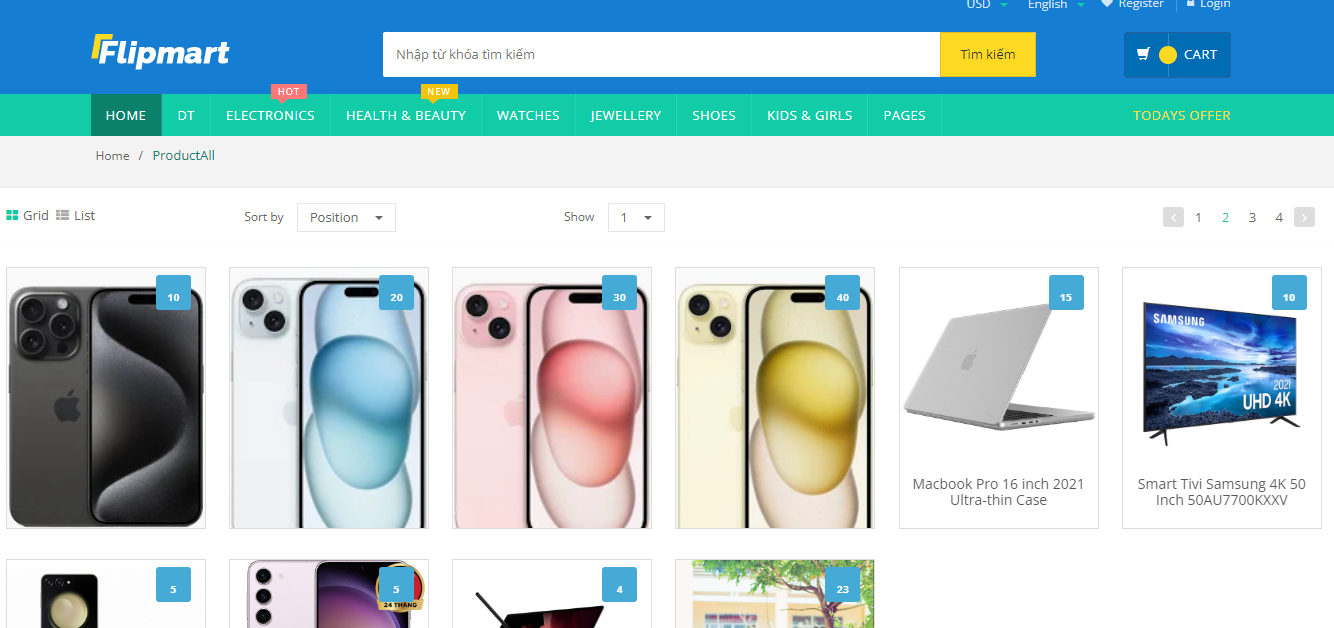
Hình 48 Sản phẩm theo danh mục

**Sản phẩm đang giảm giá**



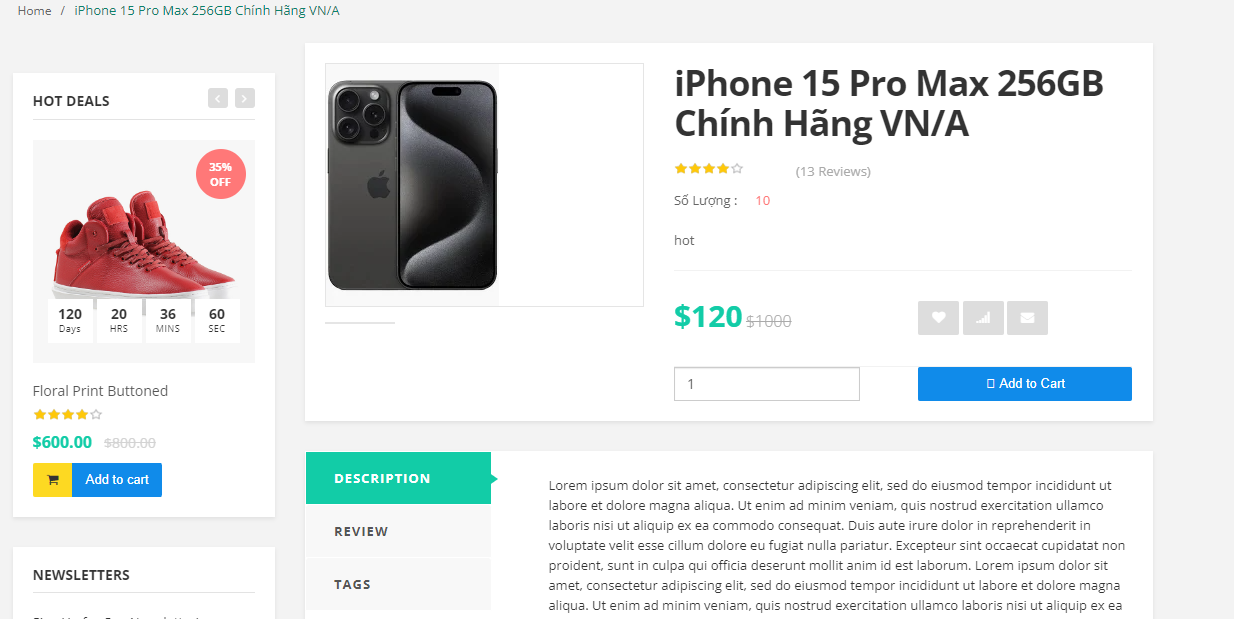
Hình 49 Sản phẩm đang giảm giá

Giao diện trang sản phẩm



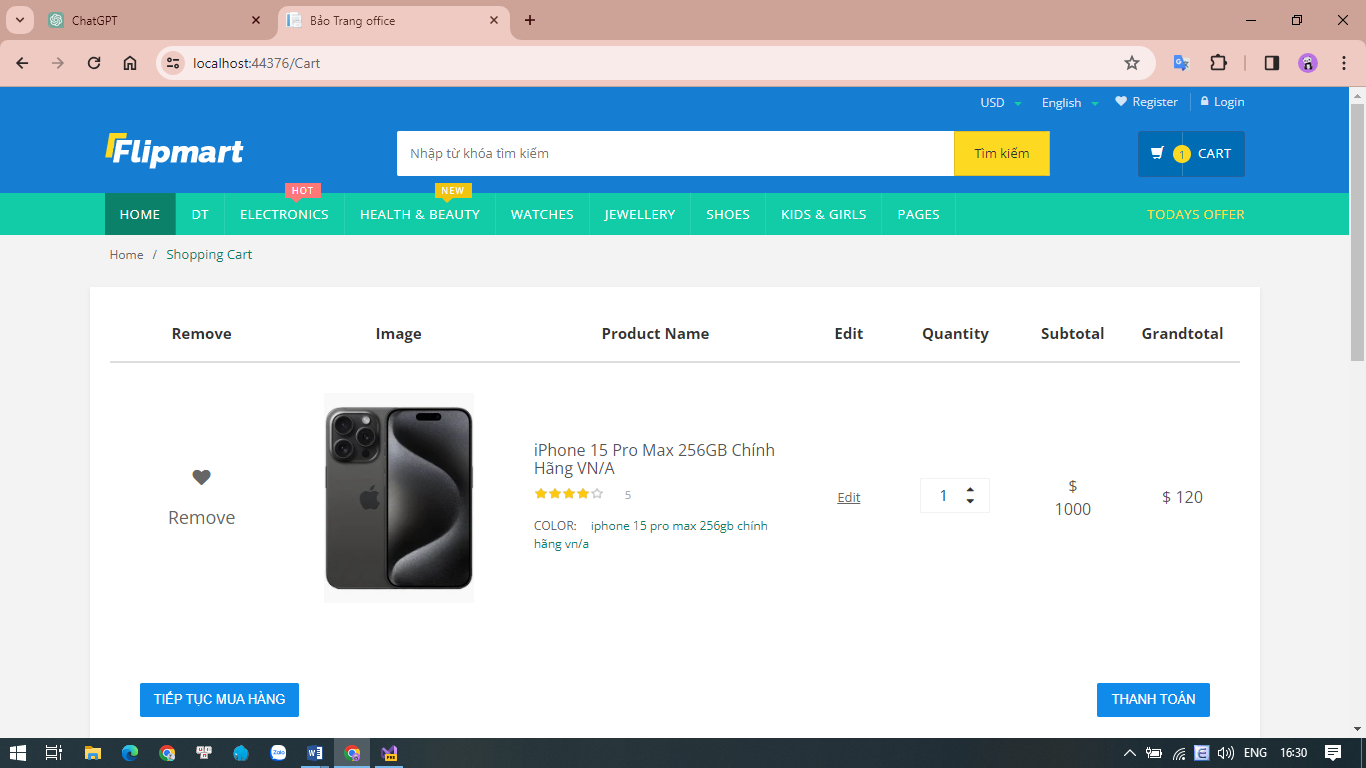
Hình 50 Giao diện trang sản phẩm

#### Giao diện trang thông tin sản phẩm



Hình 51 Giao diện trang thông tin sản phẩm

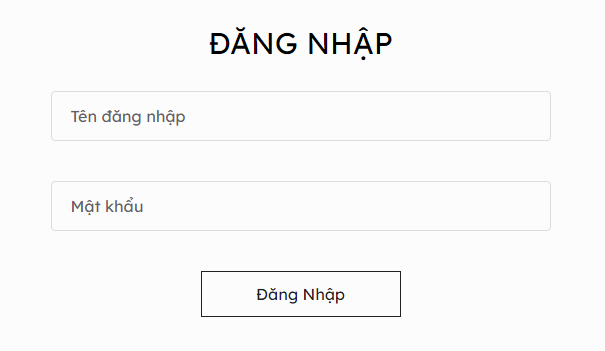
#### Giao diện giỏ hàng



Hình 52 Giao diện giỏ hàng

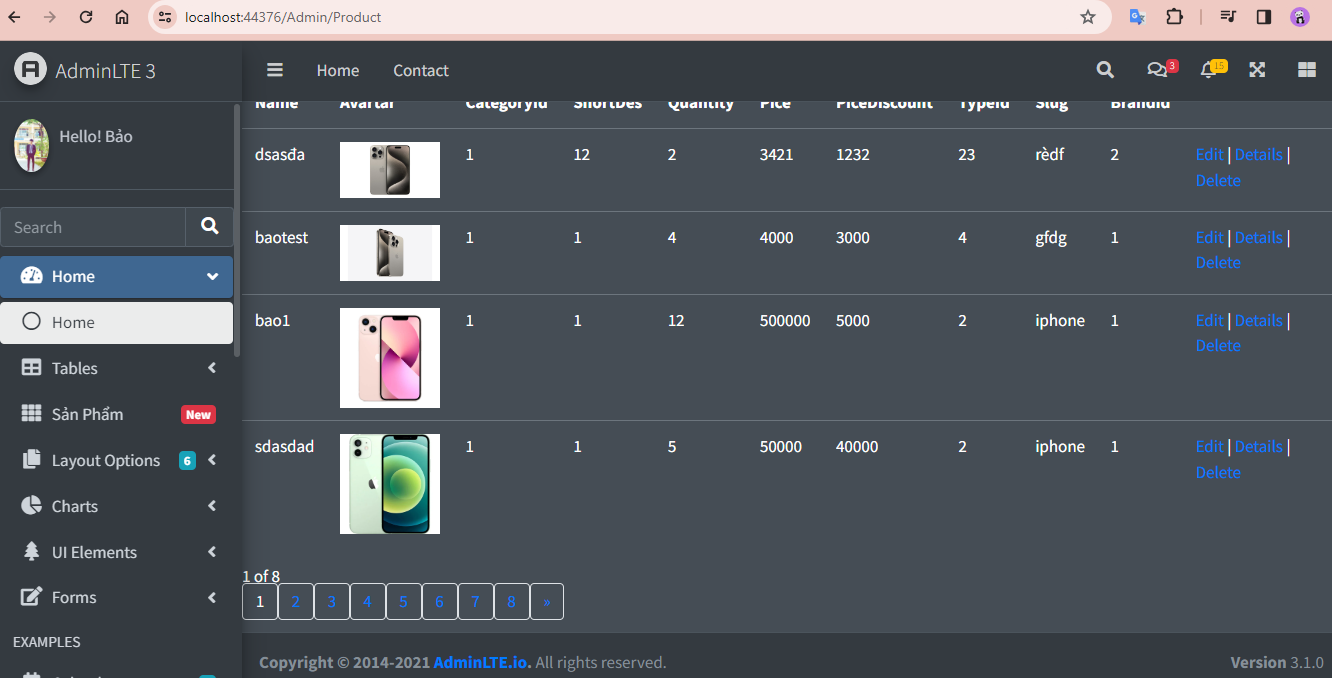
### Phía quản trị

#### Giao diện đăng nhập



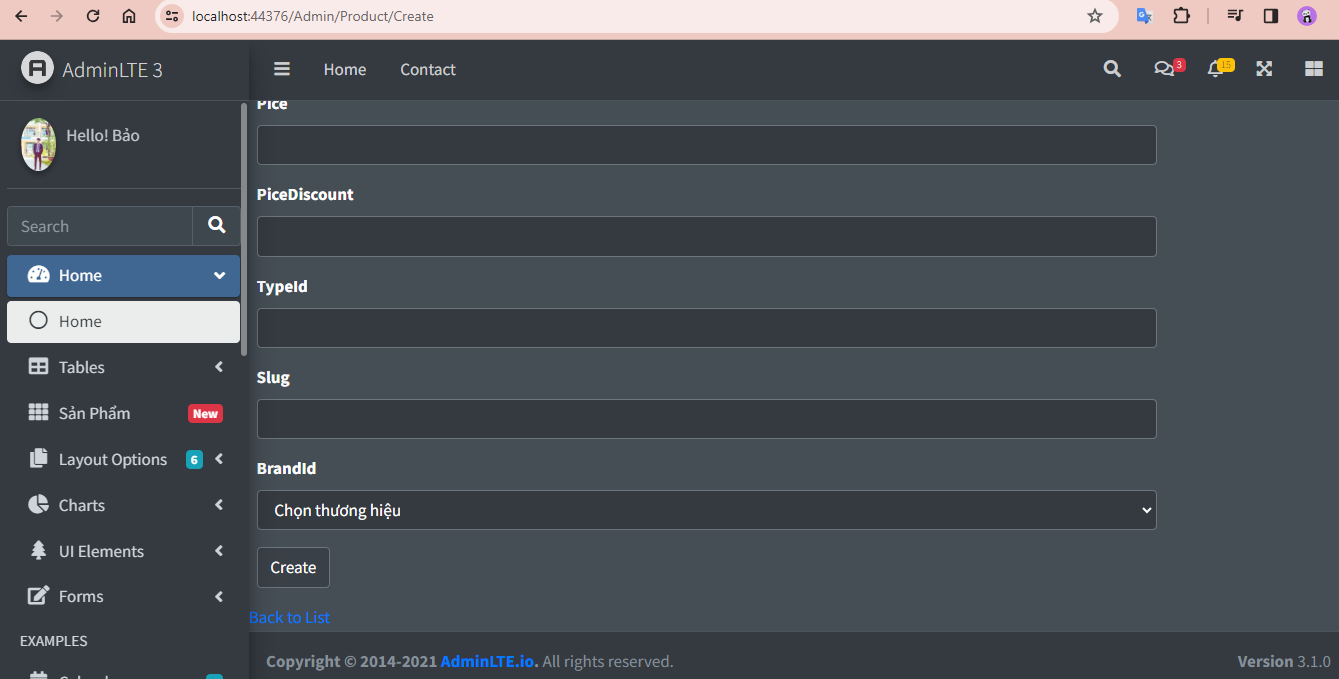
Hình 53 Giao diện đăng nhập của quản trị

#### Danh sách sản phẩm



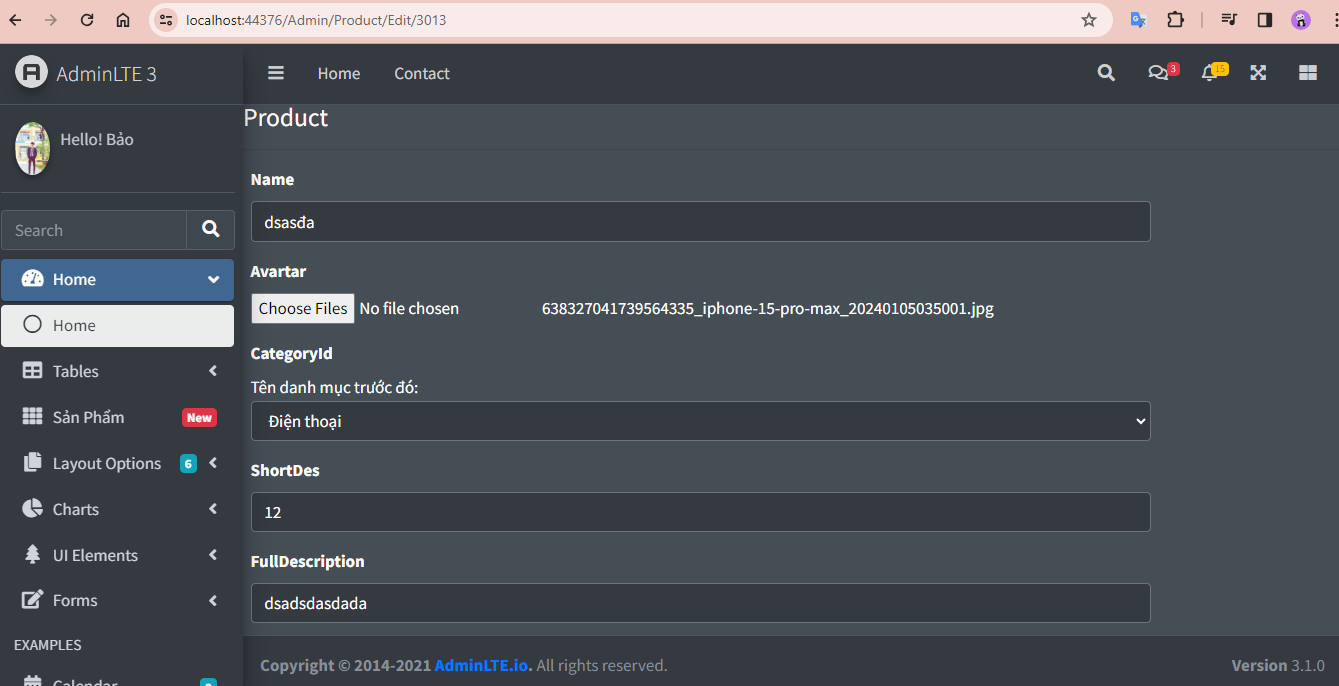
Hình 54 Danh sách sản phẩm

#### Thêm sản phẩm



Hình 55 Thêm sản phẩm

#### Cập nhật sản phẩm



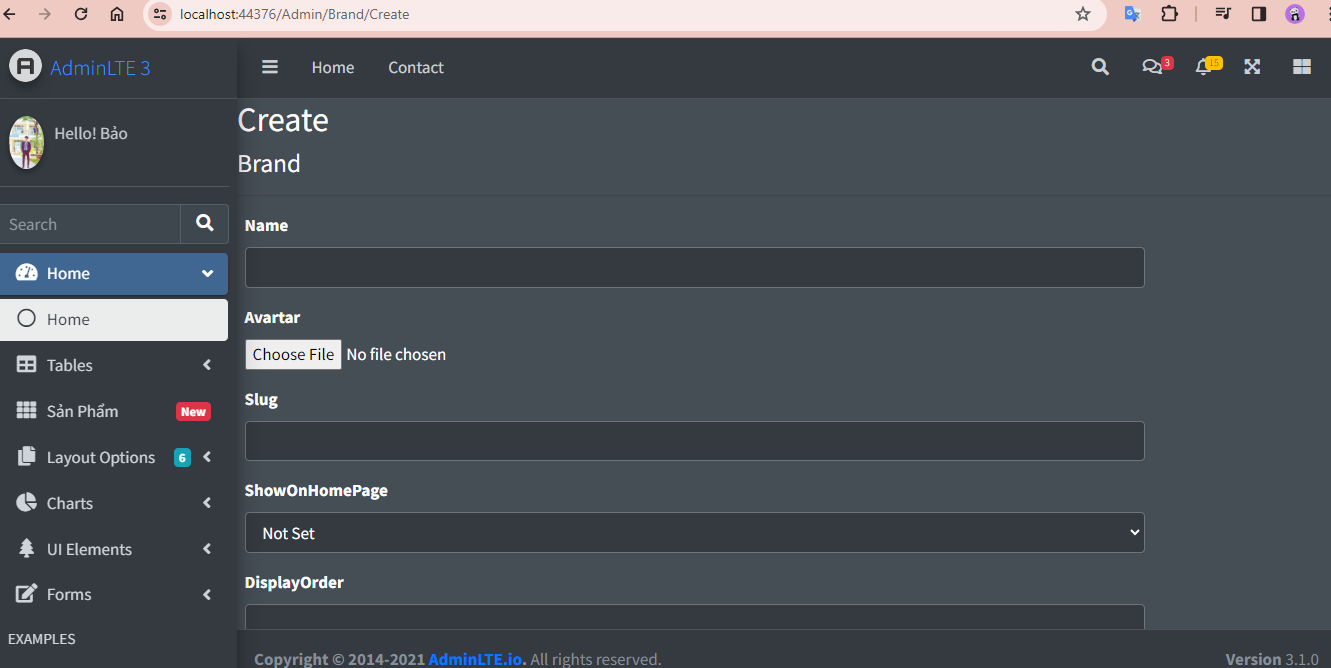
Hình 56 Cập nhật sản phẩm

#### Danh sách thương hiệu



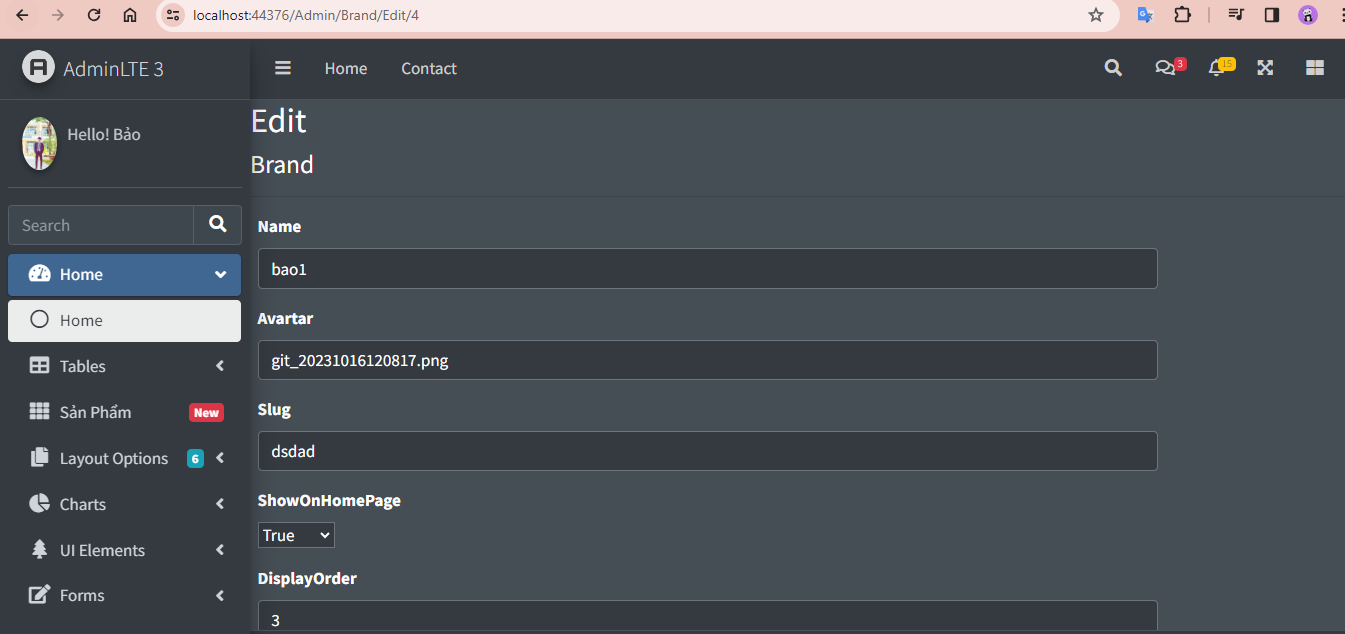
Hình 57 Danh sách thương hiệu

#### Thêm thương hiệu



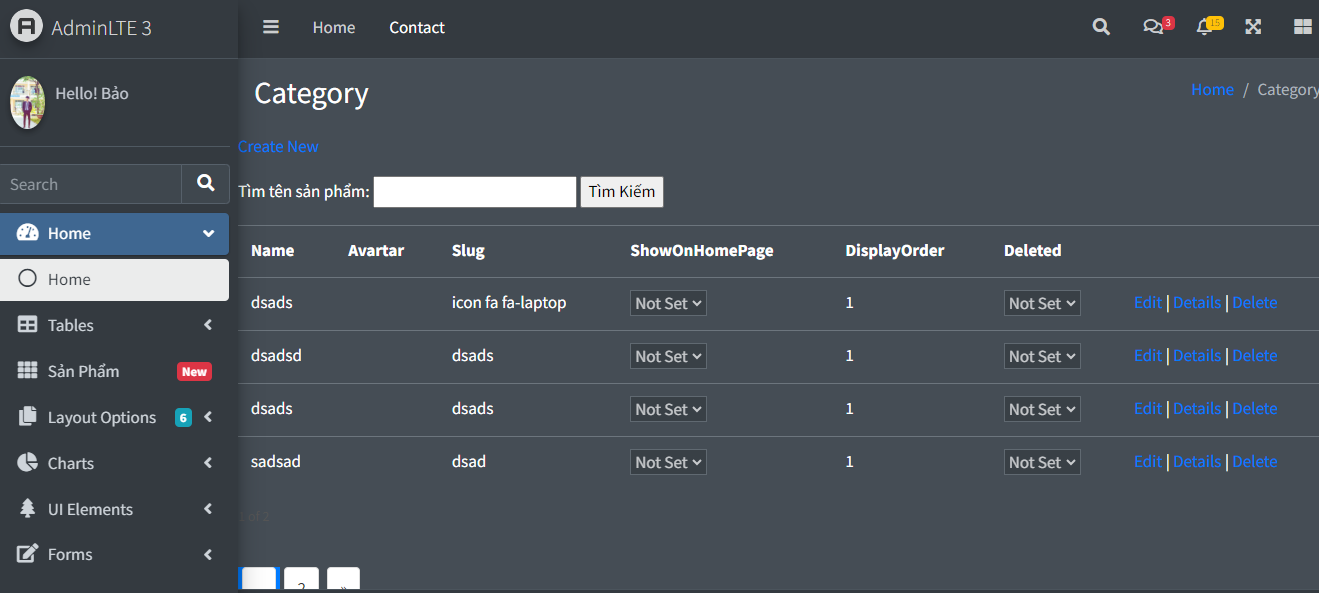
Hình Thêm thương hiệu

#### Cập nhật thương hiệu



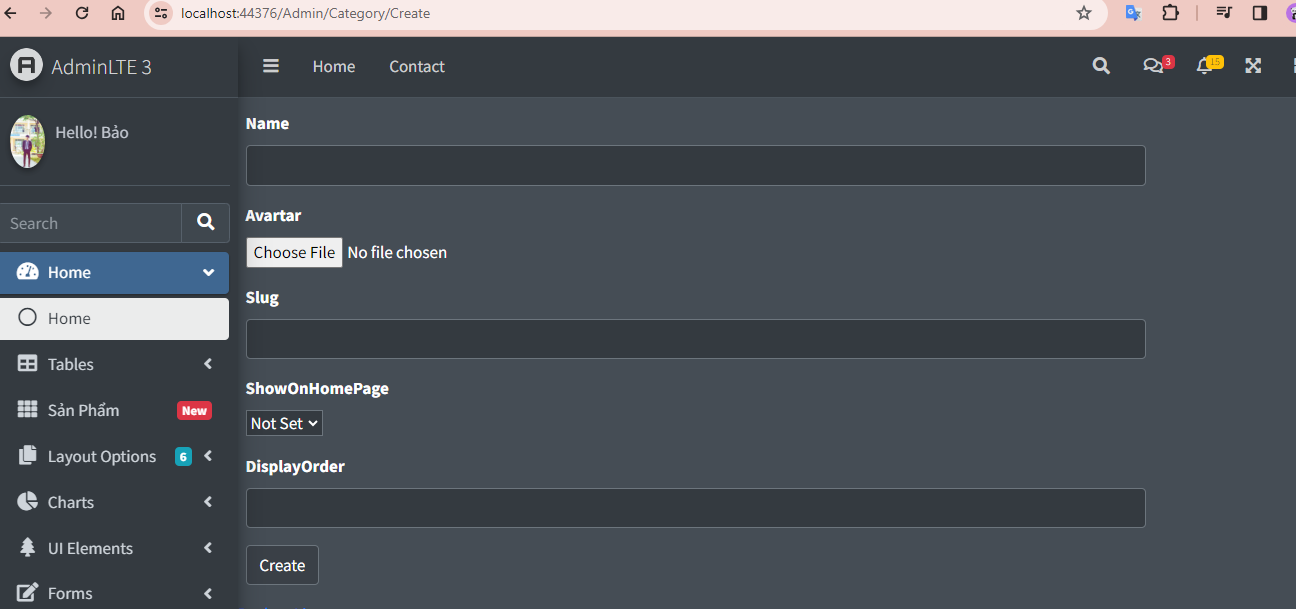
Hình 59 Cập nhật thương hiệu

#### Danh sách danh mục



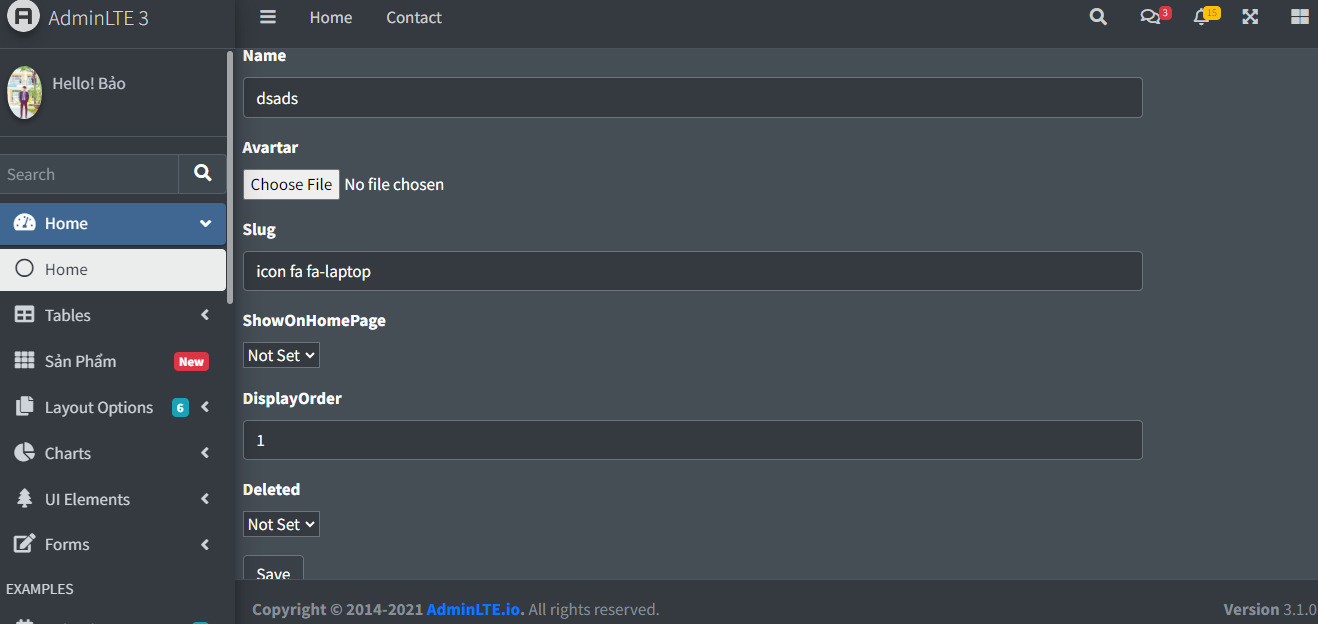
Hình 60 Danh sách danh mục

#### Thêm danh mục



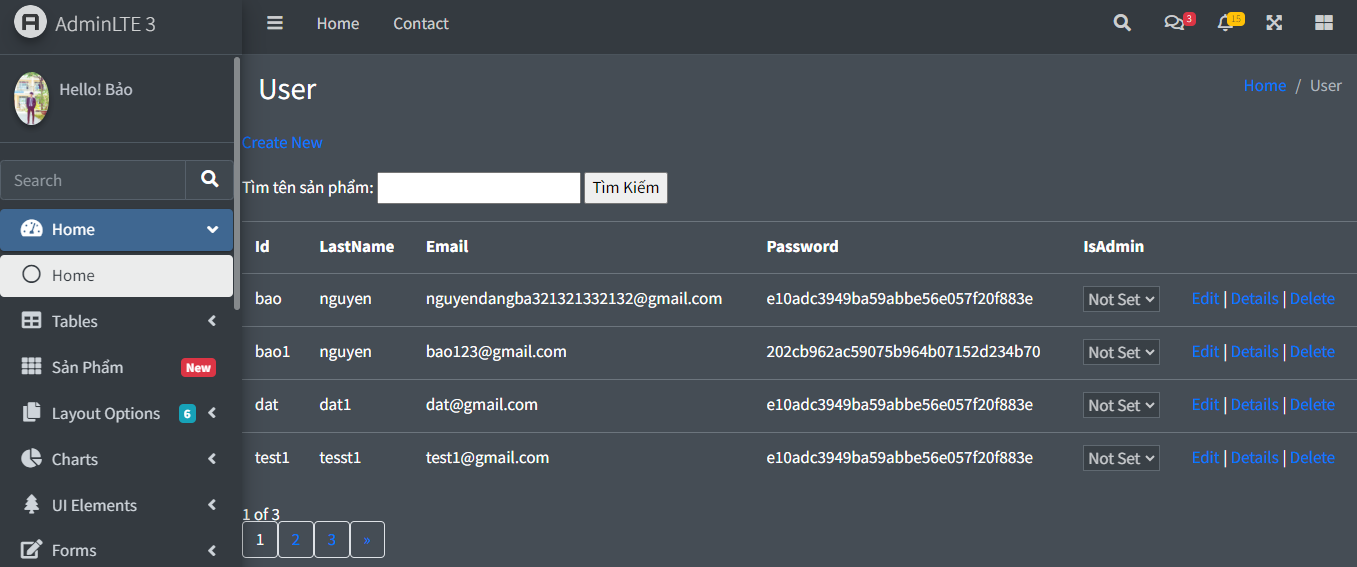
Hình 61 Thêm danh mục

#### Cập nhật danh mục



Hình 62 Cập nhật danh mục

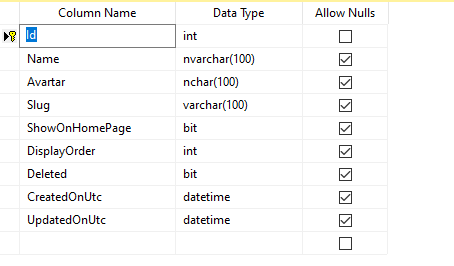
#### Danh sách người dùng



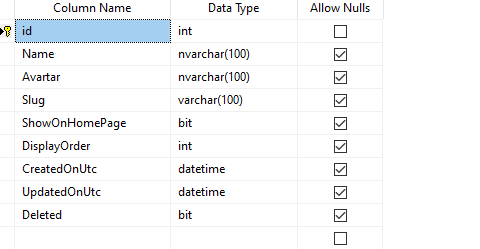
Hình 63 Danh sách người dùng

## Cơ sở dữ liệu

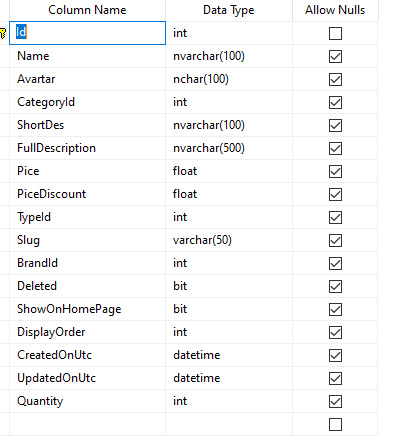
* Bảng category



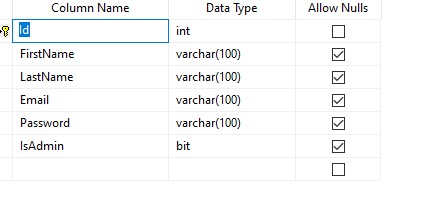
* Bảng Brand



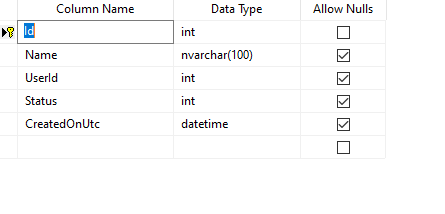
* Bảng Product



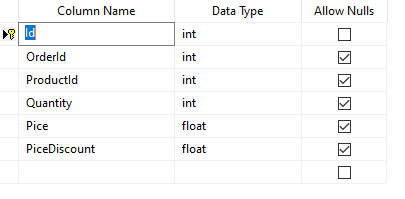
* Bảng user



* Bảng Order



* Bảng OrderDetail



# ỨNG DỤNG

## Trang người dung

### Trang quản trị

Trang quản trị được thiết kế để cung cấp cho người quản trị các công cụ mạnh mẽ để giám sát và quản lý hệ thống. Các tính năng chính bao gồm:

#### Quản lý sản phẩm

**Thêm sản phẩm:** Cho phép thêm sản phẩm mới vào hệ thống với các thông tin chi tiết.

**Sửa/Xóa Sản Phẩm:** Cập nhật hoặc xoá thông tin của sản phẩm đang tồn tại.

#### Quản lý người dung

**Xem Danh Sách Người Dùng:** Hiển thị danh sách tất cả người dùng đăng ký.

**Phân Quyền Người Dùng:** Cho phép cấp quyền truy cập tùy chỉnh cho từng người dùng.

### Trang người dung

Trang người dùng được tối ưu hóa để đảm bảo trải nghiệm mua sắm trực tuyến thuận tiện và thú vị. Các tính năng chính bao gồm:

#### Duyệt Sản Phẩm

Hiển Thị Danh Sách Sản Phẩm: Đưa ra danh sách sản phẩm theo các danh mục và tiêu chí tìm kiếm.

Chi Tiết Sản Phẩm: Xem thông tin chi tiết và đánh giá của sản phẩm.

#### Quản Lý Giỏ Hàng

Thêm/Xoá Sản Phẩm: Thêm sản phẩm vào giỏ hàng hoặc loại bỏ sản phẩm đã chọn.

Thanh Toán: Tiến hành quy trình thanh toán một cách thuận tiện.

#### Tài Khoản Người Dùng

Đăng Ký/Đăng Nhập: Người dùng có thể đăng ký tài khoản mới hoặc đăng nhập vào tài khoản đã có.

Xem Lịch Sử Đơn Hàng: Kiểm tra lịch sử các đơn hàng đã đặt.

# TỔNG KẾT

## Kết quả đạt được

Dưới đây là những kết quả mà đề tài đã đạt được:

#### Hoàn Thiện Website Bán Điện Tử

Xây dựng trang web bán điện tử với giao diện thân thiện, linh hoạt và dễ sử dụng.

Thực hiện quản lý sản phẩm, người dùng và đơn hàng một cách hiệu quả.

#### Triển Khai RESTful API

Thiết kế và triển khai RESTful API cho việc tương tác linh hoạt giữa các thành phần của hệ thống.

#### Tích Hợp Công Nghệ

Sử dụng HTML, CSS, JavaScript, ASP.NET Core MVC, và SQL Server một cách hiệu quả và mạch lạc.

#### Trải Nghiệm Mua Sắm Tốt Hơn

Cải thiện trải nghiệm mua sắm trực tuyến cho người dùng thông qua giao diện thân thiện và tính năng linh hoạt.

## Kết quả chưa đạt được

#### Tối Ưu Hiệu Suất

Cần thêm nỗ lực vào tối ưu hóa hiệu suất để đảm bảo website hoạt động mượt mà với số lượng lớn người dùng cùng một lúc.

Giao diện chưa bắt mắt kém thu hút người xem.

Hệ thống website chưa được bảo mật.

Một vài chức năng còn lỗi vặt.

#### Bảo Mật

Cần kiểm tra và cải thiện hệ thống bảo mật để đảm bảo an toàn cho thông tin người dùng và dữ liệu của hệ thống.

## Hướng phát triển của đề tài

Hướng phát triển của Website là tiến đến một website thương mại điện tử với đầy đủ các chức năng thanh toán tiền qua mạng.

Giao diện chưa thu hút được người dùng. Trong tương lai cần thiết kế giao diện đẹp, thân thiện với người dùng, có nhiều chức năng.

Trong thời gian sau này cần bổ sung các chức năng về kiểm kê: Thống kê số lần truy cập database, thống kê giá xuất nhập, tồn kho, thống kê doanh thu, xử lý hóa đơn tự động.

Mặc dù đã cố gắng hoàn chỉnh các yêu cầu nhưng bài báo cáo còn rất nhiều thiếu sót mong nhận được sự chỉ bảo hướng dẫn của các thầy cô khác giúp đỡ xem xét, đề xuất thêm các ý kiến cũng như bổ sung các vấn đề phục vụ cho việc xây dựng Website để em có thể hoàn chỉnh hơn. Em xin cảm ơn các quý thầy cô.

# TÀI LIỆU THAM KHẢO

[1] React Tutorial. [Trực tuyến]. Địa chỉ: https://reactjs.org. [Truy cập ngày 12/03/2022]

[2] ASP.NET Core Tutorial. [Trực tuyến]. Địa chỉ: https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0&tabs=visual-studio. [Truy cập ngày 12/03/2022]

[3] TailwindCSS Documention. [Trực tuyến]. Địa chỉ: https://tailwindcss.com. [Truy cập ngày 20/03/2022]

[4] Entity Framework Core Tutorial. [Trực tuyến]. Địa chỉ: https://www.entityframeworktutorial.net. [Truy cập ngày 25/03/2022]

[5] HTML CSS Tutorial. [Trực tuyến]. Địa chỉ: https://www.w3schools.com. [Truy cập ngày 12/03/2022]

[6] RESTful API. [Trực tuyến]. Địa chỉ: https://topdev.vn/blog/restful-api-la-gi/. [Truy cập ngày 12/03/2022]

[7] Three Layer [Trực tuyến]. Địa chỉ: https://topdev.vn/blog/mo-hinh-3-lop-la-gi/. [Truy cập ngày 12/03/2022]

[8] f8 Học Lập Trình Để Đi Làm [Trực tuyến]. Địa chỉ: https://fullstack.edu.vn/. [Truy cập ngày 12/03/2022]

[9] Learn ASP.NET Web API [Trực tuyến]. Địa chỉ: https://www.tutorialsteacher.com/webapi. [Truy cập ngày 20/03/2022]