**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**---------------------------------------**

**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI: XÂY DỰNG WEBSITE BÁN ĐỒ UỐNG**

**BẰNG ASP.NET**

**CBHD :Th.S An Văn Minh**

**Sinh viên : Lê Quang Linh**

**Mã Sinh Viên : 2020601821**

**Hà Nội – Năm 2024**

**LỜI CẢM ƠN**

Được sự phân công của Khoa Công nghệ thông tin Trường Đại Học Công Nghiệp Hà Nội, và dưới sự hướng dẫn của thầy ThS.An Văn Minh, em đã hoàn thành đề tài “Xây dựng website bán đồ uống bằng ASP.NET”.

Để hoàn thành đề tài này, em xin chân thành cảm ơn tới các thầy cô giáo đã tận tình hướng dẫn, giảng dạy trong suốt quá trình học tập, nghiên cứu và rèn luyện ở Trường Đại Học Công Nghiệp Hà Nội. Đặc biệt xin gửi lời cảm ơn chân thành tới thầy ThS. An Văn Minh đã tận tình, chu đáo hướng dẫn em thực hiện đề tài này.

Em cũng xin bày tỏ lòng biết ơn sâu sắc đến gia đình, người thân, bạn bè và đồng nghiệp đã giúp đỡ động viên em hoàn thiện đồ án tốt nghiệp này.

Mặc dù đã có nhiều cố gắng để thực hiện đề tài một cách hoàn chỉnh nhất. Song do thời gian có hạn, chưa có nhiều kinh nghiệm trong các dự án cho nên trong đồ án không thể tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của các thầy cô và bạn bè để em có thể hoàn thiện đề tài này tốt hơn.

*Em xin chân thành cảm ơn!*

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc167137953)

[1.1. Giới thiệu đề tài 1](#_Toc167137954)

[1.2. Mục đích của đề tài 1](#_Toc167137955)

[1.3. Lý do chọn đề tài 1](#_Toc167137956)

[1.4. Cơ sở lý thuyết của đề tài 2](#_Toc167137957)

[1.5. Cơ sở thực tiễn của đề tài 5](#_Toc167137958)

[1.6. Phương pháp nghiên cứu để xây dựng và phát triển đề tài 5](#_Toc167137959)

[CHƯƠNG 2: KIẾN THỨC TỔNG QUAN 8](#_Toc167137960)

[2.1. Cơ sở lý thuyết 8](#_Toc167137961)

[2.1.1. Tổng quan về ASP.NET Core 8](#_Toc167137962)

[2.1.2. Tổng quan về mô hình MVC 10](#_Toc167137963)

[2.1.3. Tổng quan về Entity Framework 13](#_Toc167137965)

[2.1.4. Tổng quan về hệ quản trị cơ sở dữ liệu SQL Server 16](#_Toc167137966)

[3.1 Mô hình hóa chức năng 19](#_Toc167137967)

[3.1.1 Xác định các tác nhân 19](#_Toc167137968)

[3.1.2. Yêu cầu phi chức năng 20](#_Toc167137969)

[3.2.2. Mô tả chi tiết quan hệ trong các use case 22](#_Toc167137970)

[3.3 Đặc tả chi tiết các use case 24](#_Toc167137971)

[3.3.1 Use case đăng nhập 24](#_Toc167137972)

[3.3.2 Use case đăng ký 26](#_Toc167137973)

[3.3.3 Use case tìm kiếm sản phẩm theo tên 28](#_Toc167137974)

[3.3.4 Use case xem chi tiết sản phẩm 30](#_Toc167137974)

[3.3.5. Use case hiển thị sản phẩm theo danh mục 30](#_Toc167137975)

[3.3.6. Use case lấy lại mật khẩu 32](#_Toc167137976)

[3.3.7. Use case thêm sản phẩm vào giỏ hàng 34](#_Toc167137977)

[3.3.8. Use case đặt hàng 35](#_Toc167137978)

[3.3.9. Use case Xem tin tức 37](#_Toc167137979)

[3.3.10. Use case quản lý sản phẩm 38](#_Toc167137980)

[3.3.11. Use case quản lý danh mục 41](#_Toc167137981)

[3.3.12. Use case quản lý tin tức 43](#_Toc167137982)

[3.3.13. Use case quản lý đơn hàng 46](#_Toc167137983)

[3.3.14. Use case quản lý khách hàng 48](#_Toc167137984)

[3.4 Phân tích use case 51](#_Toc167137985)

[3.4.1. Biểu đồ trình tự chức năng đăng nhập 51](#_Toc167137986)

[3.4.2 Biểu đồ trình tự chức năng đăng ký 51](#_Toc167137987)

[3.4.3 Biểu đồ trình tự chức năng tìm kiếm sản phẩm theo tên 52](#_Toc167137988)

[3.4.4 Biểu đồ trình tự chức năng xem chi tiết sản phẩm 52](#_Toc167137989)

[3.4.5 Biểu đồ trình tự chức năng thêm sản phẩm vào giỏ hàng 53](#_Toc167137990)

[3.4.6 Biểu đồ trình tự chức năng đặt hàng 53](#_Toc167137991)

[3.4.7 Biểu đồ trình tự chức năng lấy lại mật khẩu 54](#_Toc167137992)

[3.4.8 Biểu đồ trình tự chức năng xem tin tức 55](#_Toc167137993)

[3.4.9 Biểu đồ trình tự chức năng quản lý sản phẩm 56](#_Toc167137994)

[3.4.10 Biểu đồ trình tự chức năng quản lý danh mục 58](#_Toc167137995)

[3.4.11 Biểu đồ trình tự chức năng quản lý tin tức 60](#_Toc167137996)

[3.4.12 Biểu đồ trình tự use case quản lý khách hàng 62](#_Toc167137997)

[3.4.13 Biểu đồ trình tự use case hiển thị sản phẩm theo danh mục 63](#_Toc167137998)

[3.5 Mô hình cơ sở dữ liệu 63](#_Toc167137999)

[3.5.1 Thiết kế bảng 63](#_Toc167138000)

[3.5.2 Biểu đồ quan hệ thực thể 66](#_Toc167138001)

[CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG 67](#_Toc167138002)

[4.1 Công cụ sử dụng 67](#_Toc167138003)

[4.1.1 Microsoft Visual Studio 67](#_Toc167138004)

[4.1.2 Rational Rose 67](#_Toc167138005)

[4.2 Giao diện các chức năng 69](#_Toc167138006)

[TÀI LIỆU THAM KHẢO 79](#_Toc167138007)

**DANH MỤC HÌNH ẢNH**

Hình 2.1. Mô tả luồng xử lý trong mô hình MVC 12

Hình 2.2. Cấu trúc của SQL Server 17

Hình 3.1. Biểu đồ use case tổng quát 21

Hình 3.2. Quan hệ trong use case thêm sản phẩm vào giỏ hàng 22

Hình 3.3. Quan hệ trong use case quản lý sản phẩm 22

Hình 3.4. Quan hệ trong use case quản lý khách hàng 23

Hình 3.5. Quan hệ trong use case quản lý đơn hàng 23

Hình 3.6. Quan hệ trong use case quản lý tin tức 24

Hình 3.7. Quan hệ trong use case quản lý danh mục sản phẩm 24

Hình 3.8. Biểu đồ use case đăng nhập 25

Hình 3.9. Biểu đồ use case đăng ký 26

Hình 3.10. Biểu đồ use case tìm kiếm sản phẩm theo tên 28

Hình 3.11. Biểu đồ use case xem chi tiết sản phẩm 29

Hình 3.12. Biểu đồ use case hiển thị sản phẩm theo danh mục 30

Hình 3.13. Biểu đồ use case lấy lại mật khẩu 32

Hình 3.14. Biểu đồ use case thêm sản phẩm vào giỏ hàng 34

Hình 3.15. Biểu đồ use case đặt hàng 35

Hình 3.16. Biểu đồ use case xem tin tức 37

Hình 3.17. Biểu đồ use case quản lý sản phẩm 38

Hình 3.18. Biểu đồ use case quản lý danh mục 41

Hình 3.19. Biểu đồ use case quản lý tin tức 43

Hình 3.20. Biểu đồ use case quản lý đơn hàng 46

Hình 3.21. Biểu đồ use case quản lý khách hàng 48

Hình 3.22. Biểu đồ trình tự chức năng đăng nhập 51

Hình 3.27. Biểu đồ trình tự chức năng Đặt hàng 53

Hình 3.28. Biểu đồ trình tự chức năng lấy lại mật khẩu 54

Hình 3.29. Biểu đồ trình tự chức năng xem tin tức 55

Hình 3.30. Biểu đồ trình tự chức năng quản lý sản phẩm 58

Hình 3.31. Biểu đồ trình tự chức năng quản lý danh mục 60

Hình 3.32. Biểu đồ trình tự chức năng quản lý tin tức 61

Hình 3.33. Biểu đồ trình tự chức năng quản lý khách hàng 62

Hình 3.34. Biểu đồ trình tự chức năng hiển thị sản phẩm theo danh mục 63

Hình 3.35. Bảng account 63

Hình 3.36. Bảng chi tiết đơn hàng 63

Hình 3.37. Bảng danh mục 64

Hình 3.38. Bảng đơn hàng 64

Hình 3.39. Bảng khách hàng 64

Hình 3.40. Bảng quyền truy cập 65

Hình 3.41. Bảng sản phẩm 65

Hình 3.42. Bảng tin tức 65

Hình 3.44. Biểu đồ quan hệ thực thể 66

Hình 4.1. Giao diện trang chủ 69

Hình 4.2. Giao diện trang đăng nhập 70

Hình 4.3. Giao diện trang danh sách sản phẩm 70

Hình 4.5. Giao diện giỏ hàng 72

Hình 4.6. Giao diện đặt hàng 72

Hình 4.7. Giao diện trang tin tức 73

Hình 4.8. Giao diện trang chủ admin 74

Hình 4.9. Giao diện quản lý sản phẩm 74

Hình 4.10. Giao diện quản lý danh mục 75

Hình 4.11. Giao diện quản lý tin tức 76

**DANH MỤC THUẬT NGỮ, CÁC CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| **Ký hiệu** | **Giải thích** |
| HTML | Ngôn ngữ đánh dấu (Hyper Text Markup Language) |
| CSS | Ngôn ngữ tạo phong cách cho trang web (Cascading Style Sheets) |
| UC | Use case |
| CSDL | Cơ sở dữ liệu |
| SQL | Ngôn ngữ truy vấn có cấu trúc |
| Admin | Quản trị hệ thống |

# 

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

## 1.1. Giới thiệu đề tài

Tên đề tài: “ Xây dựng Website bán đồ uống bằng ASP.NET”.

## 1.2. Mục đích của đề tài

1. **Kiểm chứng kiến thức:** Đồ án tốt nghiệp là cơ hội để áp dụng tất cả kiến thức đã học trong quá trình đào tạo. Xây dựng một trang web bán hàng sẽ cần phải áp dụng nhiều khía cạnh của công nghệ ASP.NET Core.

2. **Phát triển kỹ năng:** Bằng việc hoàn thành đồ án này, em có thể phát triển khả năng phân tích, thiết kế và phát triển phần mềm web của mình trước các giảng viên, cán bộ khoa và đồng nghiệp.

3. **Thực hành quy trình phát triển phần mềm**: Quá trình phát triển đồ án sẽ giúp em hiểu rõ hơn về quy trình thiết kế, phát triển, kiểm thử và triển khai phần mềm trong một dự án thực tế.

4. **Tìm hiểu công nghệ mới:** ASP.NET Core là một công nghệ phát triển web tiên tiến, việc sử dụng nó trong đồ án tốt nghiệp sẽ giúp ta tiếp cận và làm quen với những công nghệ mới.

5. **Tiếp thu năng giải quyết vấn đề:** Xây dựng một trang web bán hàng yêu cầu em phải suy nghĩ logic, sắp xếp dữ liệu và giải quyết các vấn đề thực tế, từ đó rèn luyện khả năng tư duy logic và phân tích vấn đề.

## 1.3. Lý do chọn đề tài

Mục đích của đề tài là xây dựng một website bán đồ uống bằng ASP.NET, với mục tiêu chính là cung cấp cho bản thân em một cơ hội thực hành và áp dụng kiến thức đã học trong quá trình đào tạo. Cụ thể, các mục tiêu của đề tài bao gồm:

1. Thực hành và áp dụng kiến thức về lập trình web và phát triển ứng dụng ASP.NET.
2. Nâng cao kỹ năng kỹ thuật liên quan đến phát triển website, bao gồm cả phát triển giao diện người dùng và phát triển cơ sở dữ liệu.
3. Hiểu biết thực tiễn về quy trình phát triển phần mềm từ việc xác định yêu cầu, thiết kế, triển khai và kiểm thử.
4. Củng cố và phát triển kỹ năng phân tích thiết kế hệ thống, bao gồm việc lập mô hình và xây dựng kiến trúc hệ thống một cách khoa học và hiệu quả.
5. Cải thiện kiến thức và kỹ năng về xây dựng cơ sở dữ liệu, bao gồm thiết kế cơ sở dữ liệu quan hệ, tối ưu hóa truy vấn SQL, và đảm bảo tính toàn vẹn dữ liệu.
6. Tạo cơ hội nghề nghiệp cho bản thân trong tương lai, bằng cách xây dựng một dự án có thể trưng dụng khi tìm kiếm việc làm hoặc khi ứng dụng vào môi trường làm việc thực tế.

Ngoài ra đề tài này còn là cơ hội quan trọng để em thể hiện và phát triển kỹ năng của mình trong lĩnh vực phát triển website

## 1.4. Cơ sở lý thuyết của đề tài

**Quy trình phát triển phần mềm:**

1. **Phân tích yêu cầu:**

* Xác định các tính năng cần thiết như đăng nhập, đăng ký, xem sản phẩm, quản lý sản phẩm, giỏ hàng, quản lý đơn hàng, quản lý người dùng…
* Tạo tài liệu SRS (Software Requirements Specification) để mô tả chi tiết các yêu cầu chức năng và phi chức năng của hệ thống.
* Sử dụng các kỹ thuật mô hình hóa như Use Case Diagrams, Sequence Diagrams để mô tả các yêu cầu.

1. **Thiết kế hệ thống**

* Thiết kế kiến trúc: Xác định kiến trúc 3 tầng: Presentation Layer (HTML/CSS, Javascript, Bootstap), Business Logic Layer (ASP.NET Core), và Data Access Layer (Entity Framework Core).
* Thiết kế cơ sở dữ liệu: Thiết kế lược đồ cơ sở dữ liệu sử dụng Entity Framework Core. Xác định các bảng cần thiết: KhachHang, SanPham, DonHang, DanhMuc, v.v.

1. **Thiết kế giao diện người dùng**

* Phát triển phần giao diện, tương tác người dùng với HTML/CSS, Javascript, Bootstap, JQuery.
* Tạo các wireframe và prototype cho các trang chính: Home, DonHang, DanhSachSanPham, ChiTietSanPham, GioHang, …
* Đảm bảo giao diện người dùng thân thiện và tương thích với các thiết bị di động.

1. **Phát triển hệ thống**

* Xây dựng các component, service, và module cho các tính năng khác nhau.
* Tích hợp thêm các thư viện như MailKit, MimeKit, MoreLinq… để tối ưu hóa thời gian thực hiện dự án.
* Sử dụng Entity Framework Core để kết nối và tương tác với cơ sở dữ liệu.
* Triển khai hệ thống quản lý phiên bản: Sử dụng Git để quản lý mã nguồn, bao gồm commit, branch, merge, và pull request.

**Kiến thức về phát triển web:**

* **ASP.NET Core:** Là một framework phát triển ứng dụng web mạnh mẽ của Microsoft, dựa trên nền tảng .NET Core. Nó cung cấp một mô hình lập trình theo mô hình MVC (Model-View-Controller) để xây dựng các ứng dụng web linh hoạt và dễ bảo trì. So với ASP.NET truyền thống, ASP.NET Core có hiệu suất cao hơn, khả năng đa nền tảng và tích hợp sâu với các công nghệ mới như Docker và Kubernetes.
* **Ngôn ngữ lập trình C#:** Là ngôn ngữ chính được sử dụng trong việc phát triển các ứng dụng trên nền tảng ASP.NET Core. C# là một ngôn ngữ lập trình hiệu quả, dễ học và mạnh mẽ, với cú pháp tương tự như C++ và hỗ trợ nhiều tính năng hiện đại như generics, lập trình hướng đối tượng và async/await.
* **HTML, CSS và JavaScript:** Là các công cụ lập trình cơ bản và không thể thiếu trong phát triển giao diện người dùng của các trang web trên nền tảng ASP.NET Core. HTML định nghĩa cấu trúc của trang web, CSS điều chỉnh kiểu dáng và trang trí, và JavaScript thêm tính năng tương tác động cho trang web.
* **Entity Framework Core (EF Core):** Là một ORM cho .NET, giúp phát triển các ứng dụng ASP.NET Core dễ dàng làm việc với cơ sở dữ liệu mà không cần viết nhiều mã SQL. EF Core hỗ trợ lập bản đồ đối tượng, truy vấn LINQ, và các phương pháp quản lý cơ sở dữ liệu như code-first và database-first.
* **Hệ quản trị cơ sở dữ liệu SQL:** SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ của Microsoft, thường được sử dụng cùng với ASP.NET Core. SQL Server hỗ trợ lưu trữ, quản lý và truy vấn dữ liệu hiệu quả. Nó cung cấp các tính năng mạnh mẽ như các thủ tục lưu trữ, khóa ngoại, giao dịch và tích hợp với Entity Framework Core để quản lý dữ liệu trong các ứng dụng ASP.NET Core.

**Bảo mật và xác thực:**

* Đảm bảo an toàn và bảo mật cho thông tin người dùng và dữ liệu bằng cách sử dụng các biện pháp bảo mật như xác thực và ủy quyền.
* Mã hóa dữ liệu quan trọng trước khi lưu trữ vào cơ sở dữ liệu để đảm bảo tính bảo mật của nó.

## 1.5. Cơ sở thực tiễn của đề tài

* Người dùng có thể dễ dàng tìm kiếm và đặt hàng sản phẩm mà mình muốn mua trực tuyến một cách nhanh chóng và thuận tiện.
* Người dùng có thể chọn sản phẩm theo nhu cầu của mình bao gồm chọn sản phẩm theo tên, theo danh mục, theo giá bán…
* Người dùng mua hàng có thể theo dõi đơn hàng đã đặt hoặc có thể hủy đơn hàng đã dặt một cách nhanh chóng.
* Người quản lý có thể quản lý người dùng, sản phẩm, đơn hàng và thống kê doanh thu.
* Thông tin khách hàng được bảo vệ bằng các công nghệ bảo mật hiện đại, đảm bảo tính riêng tư và an toàn.
* Các thông tin về sản phẩm sẽ được quản trị viên liên tục cập nhật và kiểm tra định kỳ để đảm bảo tính chính xác và độ tin cậy cho khách hàng.

## 1.6. Phương pháp nghiên cứu để xây dựng và phát triển đề tài

**Nghiên cứu thị trường và phân tích yêu cầu:**

* Tiến hành khảo sát thị trường để hiểu về nhu cầu và xu hướng mua sắm hàng trực tuyến.
* Phân tích yêu cầu chức năng và phi chức năng của trang web, xác định các tính năng quan trọng như quản lý sản phẩm, đơn hàng, khách hàng và tích hợp thanh toán.

**Nghiên cứu công nghệ mới:**

* Nghiên cứu về mô hình MVC và ASP.NET Core, bao gồm các tính năng, ưu điểm, hạn chế và cách thức triển khai.
* So sánh và lựa chọn công nghệ phù hợp cho dự án, dựa trên yêu cầu chức năng và mục tiêu đề ra.

**Tìm hiểu về bài toán:**

* Xây dựng bài toán: Bao gồm giao diện tương tác người dùng, giao diện quản lý và phần xử lý các yêu cầu từ phía người dùng, trả về kết quả.
* Thiết kế cấu trúc dữ liệu và quan hệ giữa các thành phần của hệ thống.

**Triển khai và phát triển phần mềm:**

* Phát triển giao diện người dùng sử dụng HTML/CSS và JavaScript, bao gồm các trang chính như trang chủ, trang sản phẩm, giỏ hàng và trang thanh toán.
* ASP.NET Core để cung cấp dữ liệu cho giao diện người dùng, bao gồm các endpoint cho việc quản lý sản phẩm, đơn hàng, khách hàng và tích hợp thanh toán.

**Tích hợp và kiểm thử:**

* Tích hợp các thành phần của hệ thống và kiểm tra tính tương thích và ổn định.
* Tiến hành kiểm thử chức năng và hiệu suất cho cả phần giao diện người dùng và phần xử lý yêu cầu từ phía người dùng của trang web.

**Triển khai và đánh giá:**

* Triển khai trang web trên môi trường sản phẩm thực tế và đảm bảo sự ổn định và khả dụng của hệ thống.
* Đánh giá hiệu suất của hệ thống dựa trên các tiêu chí như tốc độ tải trang, thời gian phản hồi và độ tin cậy của các tính năng.

**Tổng kết và đánh giá kết quả:**

* Tổng kết quá trình phát triển dự án và đánh giá mức độ đạt được các mục tiêu đã đề ra.
* Đề xuất các cải tiến và hướng phát triển tương lai cho hệ thống.

# **CHƯƠNG 2: KIẾN THỨC TỔNG QUAN**

## 2.1. Cơ sở lý thuyết

### **2.1.1. Tổng quan về ASP.NET Core**

**ASP.NET Core là gì?**

ASP.NET Core là một open-source mới và framework đa nền tảng cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và back-end cho mobile. 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 hoặc chạy on-promise.

ASP.NET Core bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

Đồng thời nó đã trở thành một mã nguồn mở. Đây là một thay đổi rất lớn và theo mình là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.

**ASP.NET Core ra đời trong hoàn cảnh nào?**

ASP.NET Core là một phiên bản mới của ASP.NET, được phát hành bởi Microsoft và là một mã nguồn mở trên GitHub. ASP.NET Core được sử dụng để phát triển khuôn khổ website và có thể thích ứng với nhiều trình duyệt khác nhau như Windows, Mac hoặc Linux kể cả trên nền tảng MVC. Ban đầu, phiên bản này có tên là ASP.NET 5 nhưng sau đó được đổi tên thành ASP.NET Core.

ASP.NET Core được thiết kế để tối ưu development framework cho những ứng dụng cái mà được chạy on-promise hay được triển khai trên đám mây. ASP.NET Core bao gồm các thành phần theo hướng module với mục đích tối thiểu tài nguyên và tiết kiệm chi phí khi phát triển. Đồng thời, ASP.NET Core cũng là một mã nguồn mở, một xu thế mà các ngôn ngữ lập trình hiện nay hướng đến.

Khi sử dụng ASP.NET Core chúng ta có thể có được nên tảng sau:

* Tích hợp những client-side frameworks hiện đại và những luồng phát triển.
* Hệ thống cấu hình dựa trên môi trường đám mây thật sự.
* Dependency injection được xây dựng sẵn.
* HTTP request được tối ưu nhẹ hơn.
* Có thể host trên IIS hoặc self-host trong process của riêng chúng ta.
* Được xây dựng trên .NET Core, hỗ trợ thực sự app versioning.
* Chuyển các thực thể, thành phần, module như những NuGet packages.
* Những công cụ mới để đơn giản hóa quá trình phát triển web hiện đại.
* Xây dựng và chạy đa nền tảng (Windows, Mac và Linux).
* Mã nguồn mở và tập trung vào cộng đồng.

**Những lợi ích khi sử dụng ASP.NET Core**

Xây dựng web UI và web API sửng dụng ASP.NET Core:

* Chúng ta có thể tạo ra các ứng dụng web có thể kiểm thử theo mô hình MVC.
* Chúng ta có thể xây dựng HTTP services cái mà hỗ trợ nhiều định dạng và có đầy đủ hỗ trợ cho nội dung của dữ liệu trả về.
* Razor cung cấp một ngôn ngữ hiệu quả để tạo Views.
* Tag Helper cho phép code server side tham gia vào việc tạo và render phần tử HTML.
* Model Binding tự động ánh xạ dữ liệu từ HTTP request tới tham số của method action.
* Model Validation tự động thực hiện validate client và server.

Phát triển client-side:

* ASP.NET Core được thiết kế để tích hợp một cách liên tục với nhiều client-side frameworks, bao gồm AngularJS, KnockoutJS và Bootstrap.

Ngoài ra, ASP.NET Core còn được đánh giá cao khi mang trong mình nhiều lợi ích khác như:

* Là khung mẫu và gọn gàng hơn kiến trúc nhiều tầng.
* Là một khung công tác với mã nguồn mở.
* Dễ dàng tạo ứng dụng ASP.NET đa nền tảng trên Windows, Mac và Linux.
* Cấu hình là môi trường sẵn sàng cho đám mây.
* Khả năng lưu trữ trên: IIS, HTTP.sys, Kestrel, Nginx, Apache và Docker.

### **2.1.2. Tổng quan về mô hình MVC**

**Mô hình MVC là gì?**

**MVC** là viết tắt của cụm từ “Model-View-Controller“. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

* **Model (dữ liệu)**: Quản lí xử lí các dữ liệu.
* **View (giao diện)**: Nới hiển thị dữ liệu cho người dùng.
* **Controller (bộ điều khiển)**: Điều khiển sự tương tác của hai thành phần Model và View.

Mô hình MVC thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

**Các thành phần trong mô hình MVC**

Mô hình MVC gồm 3 loại chính là thành phần bên trong không thể thiếu khi áp dụng mô hình này:

* **Model:** Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản.
* **View:** Đây là phần giao diện dành cho người sử dụng. View là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được.
* **Controller:**Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View. Nó nhận input và thực hiện các update tương ứng.

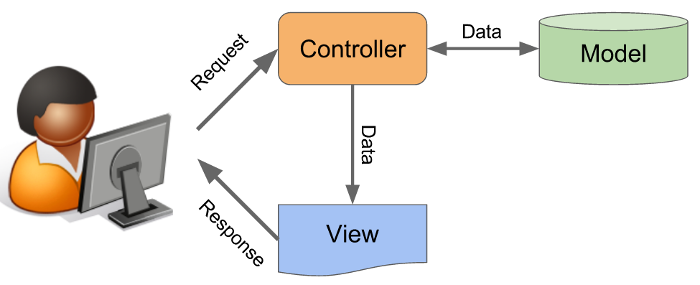
**Lịch sử của MVC**

MVC được tiến sĩ Trygve Reenskaug đưa vào ngôn ngữ lập trình Smalltalk-76 khi ông đến trung tâm Nghiên cứu Xerox Palo Alto (PARC) vào giữa năm 1970. Sau đó, việc triển khai trở nên phổ biến trong các phiên bản khác của Small- Talk. Năm 1988, các bài báo “The Journal of Object Technology” – JOT mang lại bước tranh toàn cảnh về MVC mang liệu sự hiệu quả tốt nhất**.**

**Luồng xử lý trong MVC**

Luồng xử lý trong của mô hình MVC, bạn có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:

* Khi một yêu cầu của từ máy khách (Client) gửi đến Server, thì nó sẽ được bộ định tuyến (Routing) của ASP.NET Core MVC xử lý trước. Bộ định tuyến sẽ xác định xem yêu cầu đó tương ứng với Controller và Action nào, sau đó chuyển yêu cầu đến Controller để xử lý.
* Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
* Model chuẩn bị data và gửi lại cho Controller.
* Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

****

Hình 2.1. Mô tả luồng xử lý trong mô hình MVC (nguồn: https://vietnix.vn)

**Lợi ích của mô hình MVC trong ASP.NET Core**

Mô hình MVC trong ASP.NET Core mang lại nhiều lợi ích quan trọng cho việc phát triển ứng dụng web:

* **Tính tổ chức và dễ bảo trì:** Mô hình MVC phân chia ứng dụng thành ba phần chính: Models, Views, và Controllers. Điều này giúp tổ chức mã nguồn của bạn một cách rõ ràng và dễ dàng bảo trì. Mỗi phần có một trách nhiệm cụ thể, giúp giảm thiểu sự phụ thuộc giữa các phần khác nhau và tạo điều kiện cho việc tìm và sửa lỗi dễ dàng hơn.
* **Tính linh hoạt và tái sử dụng:** Mô hình MVC cho phép tái sử dụng mã nguồn một cách hiệu quả. Models chứa logic kinh doanh và dữ liệu, Controllers xử lý luồng điều khiển, và Views hiển thị dữ liệu cho người dùng. Bạn có thể tái sử dụng các phần của ứng dụng một cách dễ dàng, giúp tăng tốc quá trình phát triển và giảm thiểu việc lặp lại công việc.
* **Phát triển đồng thời:** Mô hình MVC cho phép nhiều nhóm phát triển làm việc đồng thời trên các phần khác nhau của ứng dụng mà không ảnh hưởng đến nhau. Models, Views, và Controllers là các phần độc lập, giúp tăng cường tính hiệu quả của quá trình phát triển và giảm thiểu xung đột mã nguồn.
* **Kiểm thử dễ dàng:** Mô hình MVC tách biệt các phần của ứng dụng, tạo điều kiện cho việc kiểm thử dễ dàng hơn. Bạn có thể kiểm thử các phần của ứng dụng một cách độc lập, từ các phần logic kinh doanh trong Models và Controllers đến các giao diện người dùng trong Views.
* **Tích hợp dễ dàng với các công nghệ khác:** Mô hình MVC là một cấu trúc linh hoạt, cho phép tích hợp với nhiều công nghệ khác nhau. Ví dụ, bạn có thể tích hợp MVC với Angular hoặc React để tạo ra các ứng dụng web SPA (Single Page Application), hoặc tích hợp với Web API để xây dựng ứng dụng web service.

### **2.1.3. Tổng quan về Entity Framework**

**Entity Framework là gì?**

Entity Framework được phát triển từ năm 2008 bởi Microsoft. Hiểu một cách đơn giản, Entity Framework là một thư viện ORM hay một Framework làm việc cùng cơ sở dữ liệu do Microsoft khuyến nghị giúp phản xạ quan hệ đối tượng qua lại giữa các object của một chương trình, bảng cơ sở dữ liệu, bản ghi,...

Bằng cách cung cấp cơ chế tự động hoá các hoạt động liên quan đến CSDL cho các doanh nghiệp mà việc lưu trữ, truy cập dữ liệu và xây dựng ứng dụng đơn giản và dễ dàng.

**Đặc điểm cơ bản của Entity Framework**

Để có thể hiểu rõ hơn về Entity framework và ứng dụng được Entity framework linh hoạt trong công việc thì việc tìm hiểu thêm về các đặc điểm cơ bản của nó là không thể nào thiếu được.

Dưới đây là ba đặc điểm cơ bản và quan trọng mà một lập trình viên nên biết đến và hiểu rõ khi sử dụng Entity framework:

* Bằng một lớp con của DBContent mà tất cả các cơ sở dữ liệu đều được hiển thị. Bạn hoàn toàn có thể tự động thực hiện hay can thiệp vào những thao tác như thể hiện object mà bạn xây dựng trên mỗi hàng hay thể hiện thuộc tính của object ở mỗi cột.
* Với Entity framework, bạn có thể truy vấn dữ liệu sử dụng LINQ mà không cần đến SQL. Điều này giúp dễ dàng hơn khi thực hiện các truy vấn CRUD từ code C# với các class.
* Khả năng thực hiện các thao tác với cấu trúc dữ liệu nhanh chóng và đơn giản như tạo bảng, thay đổi cấu trúc bảng, tạo cơ sở dữ liệu,... mà không bị mất dữ liệu nhờ công cụ Migration.

**Các thành phần của Entity Framework**

Việc tìm hiểu thêm về các thành phần của Entity framework cũng giúp bạn hiểu rõ hơn về Entity framework. Các thành phần cơ bản này bao gồm:

* **Entity Data Model:** là thành phần dùng để lưu trữ các thông tin ánh xạ, nối các lớp model với cơ sở dữ liệu, mang lại sự tách biệt giữa chương trình và CSDL với 3 thành phần chính là Conceptual Model, Mapping và Storage Model.
* **LINQ to Entities và Entity SQL:** Là hai ngôn ngữ truy vấn được sử dụng với mục đích giúp viết các truy vấn tới object model và trả về các thực thể được định ngĩa trong Conceptual Model.
* **Object Service:** Cụ thể hoá quá trình chuyển đổi và truy xuất dữ liệu. Ngoài ra, nó còn đảm nhận nhiệm vụ quản lý và theo dõi trạng thái thay đổi của Object.
* **Entity Client Data Provider:** Thành phần này chịu trách nhiệm tương tác với Data provider và chuyển đổi truy vấn LINQ to Entities và Entity SQL để truy vấn SQL.

**Lợi ích của việc sử dụng Entity Framework là gì?**

Entity framework là công cụ phổ biến được sử dụng bởi hầu hết các doanh nghiệp. Lý do mà Entity framework được yêu thích như vậy chính là những lợi ích to lớn mà nó mang lại.

* **Tăng năng suất:** Khả năng giảm lượng code và các công việc dư thừa của Entity framework giúp tăng năng suất cho người sử dụng, tiết kiệm thời gian, chất xám và công sức của họ trong quá trình làm việc.
* **Đơn giản hoá bảo trì phần mềm:**Với lượng code ít khi sử dụng Entity framework, việc bảo trì và kiểm tra phần mềm được đơn giản hơn nhiều đặc biệt là khi bạn phải thường xuyên bảo trì.
* **Không cần viết code SQL:** Bạn có thể sử dụng LINQ to Entities khi thực hiện các truy vấn dữ liệu để thao tác với các Object được ra đời từ Entity framework.
* **Tiết kiệm thời gian viết code:** Bạn sẽ tiết kiệm được thời gian viết code khá lớn để thao tác với database bởi Entity framework sẽ tạo ra các classes cho việc truy xuất CSDL.
* Entity framework tự động tạo ra các commands, classes tương ứng cho việc insert, select, update, delete, … dữ liệu từ CSDL quan hệ.
* **Làm việc đơn giản với EF:** Giúp người dùng có thể làm việc với EF một cacgs đơn giản và thuận tiện hơn bởi Entity framework được đặt trong .NET Framework và tích hợp trong Visual studio.

Có thể dễ dàng thấy được Entity framework là một thuật ngữ khá dễ hiểu, mang lại cho các nhà lập trình viên những giá trị lợi ích lớn. Hy vọng, với những thông tin được cung cấp trong bài viết, bạn đã có cái nhìn tổng quan hơn về Entity framework là gì và những thành phần cơ bản của nó.

### **2.1.4. Tổng quan về hệ quản trị cơ sở dữ liệu SQL Server**

**SQL Server là gì?**

SQL Server (Structured Query Language) là một loại phần mềm được phát triển bởi Microsoft, dùng để lưu trữ và truy xuất dữ liệu dựa theo tiêu chuẩn RDBMS (Relational Database Management System). Một RDBMS bao gồm: cơ sở dữ liệu, công cụ cơ sở dữ liệu, các ứng dụng quản lý dữ liệu và các bộ phận khác.

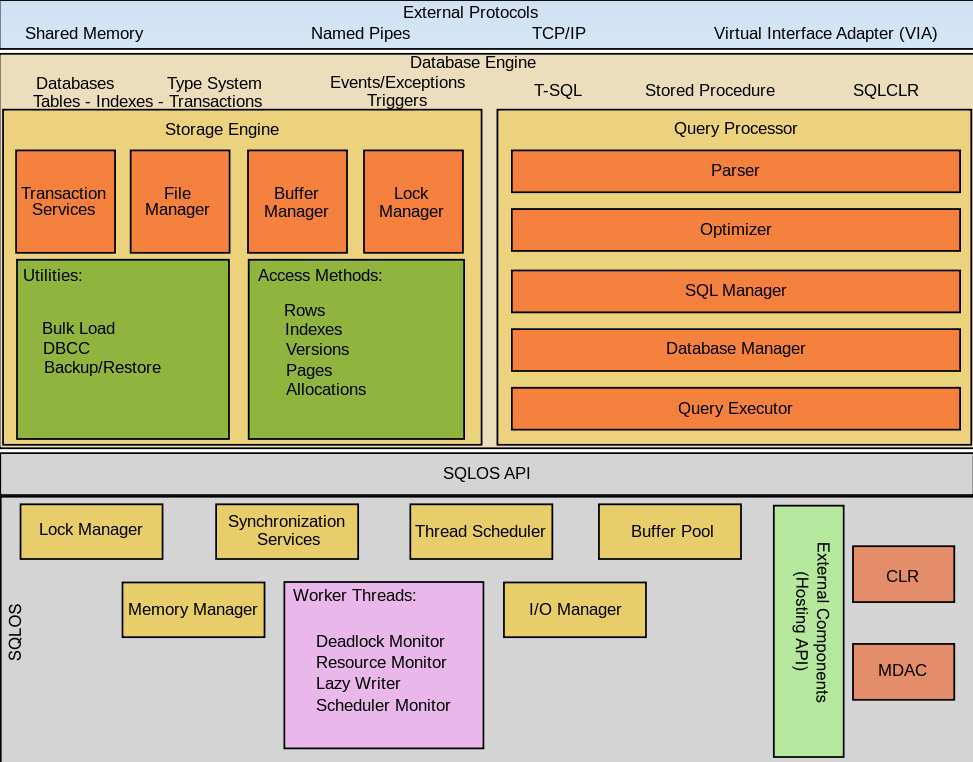
Phiên bản đầu tiên của Microsoft SQL Server ra đời vào năm 1989 cho các hệ điều hành chạy 16 bit với SQL Server phiên bản 1.0 và tiếp tục phát triển cho tới ngày nay.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn lên đến Tera-Byte và có khả năng phục vụ hàng ngàn người dùng cùng lúc.

**Các thành phần của SQL Server**

SQL Server cơ bản được thiết lập dựa trên một cấu trúc bảng biểu, bao gồm các dòng kết nối các phần tử dữ liệu có liên quan trong các bảng khác nhau. Do đó loại bỏ nhu cầu lưu trữ dữ liệu ở nhiều vị trí trong cơ sở dữ liệu.Thành phần chính của SQL Server là Database Engine và SQLOS.

* **Database Engine:** Đây là thành phần cốt lõi của SQL Server, có chức năng quản lý việc lưu trữ, xử lý và bảo mật dữ liệu. Database Engine bao gồm rất nhiều các công cụ khác nhau như công cụ lưu trữ quản lý các tệp, bảng, trang, chỉ mục, bộ đệm dữ liệu và giao dịch cơ sở dữ liệu
* **SQLOS:** Là tầng cuối cùng trong kiến trúc tổng thể của SQL Server. SQLOS sẽ chịu trách nhiệm xử lý các nhiệm vụ như quản lý bộ nhớ, lên lịch nhiệm vụ, khoá dữ liệu nhằm tránh các xung đột ngoài ý muốn có thể xảy ra mỗi khi thực hiện các thao tác cập nhật.



Hình 2.2. Cấu trúc của SQL Server

**SQL Server có những tính năng nào?**

Microsoft cung cấp tính năng quản lý dữ liệu cùng SQL Server với các dịch vụ tích hợp lập trình SQL Server, SQL Server Data Quality và SQL Server Master. Ngoài ra, hai bộ công cụ dành riêng cho quản trị viên cơ sở dữ liệu và lập trình viên:

* SQL Server Data Tools: Được sử dụng trong việc phát triển cơ sở dữ liệu.
* SQL Server Management Studio được ứng dụng để triển khai, giám sát và quản lý cơ sở dữ liệu.

SQL Server còn được trang bị tính năng kinh doanh giúp người dùng có thể thực hiện phân tích dữ liệu thông qua:

* SQL Server Analysis Services: sử dụng để phân tích các dữ liệu.
* SQL Server Reporting Services: để tạo ra báo cáo dễ dàng hơn.

**Tại sao nên sử dụng SQL Server trong thiết kế website ?**

SQL Server không phải là một hệ quản trị cơ sở dữ liệu độc lập. Nó là thành phần với vai trò ngôn ngữ làm công cụ giao tiếp của cơ sở dữ liệu và người dùng. Bởi vậy, nó được sử dụng trong các dịch vụ thiết kế web được tích hợp tính năng giao tiếp với người dùng với các vai trò sau:

* SQL là một ngôn ngữ đòi hỏi sự tương tác cao. Người dùng có thể dễ dàng trao đổi với tiện ích bằng câu lệnh SQL đến cơ sở dữ liệu và nhận lại kết quả từ đó.
* SQL là một ngôn ngữ lập trình cơ sở dữ liệu. Bằng cách nhúng các câu lệnh SQL trong ngôn ngữ lập trình, các lập trình viên có thể xây dựng được các chương trình ứng dụng giao tiếp với cơ sở dữ liệu.

**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

Trong chương 3 tác giả sẽ trình bày chi tiết về hệ thống website, các nội dung chương này sẽ được trình bày cụ thể về: các yêu cầu hệ thống, biểu đồ use case, đặc tả và phân tích các use case, thiết kế cơ sở dữ liệu.

## 3.1 Mô hình hóa chức năng

### **3.1.1 Xác định các tác nhân**

Dựa vào yêu cầu, ta có các tác nhân sau: Quản trị viên, khách hàng

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| Quản trị viên | Người có quyền cao nhất trong hệ thống, được cung cấp các chức năng: Đăng nhập, đăng ký, quên mật khẩu, tạo tài khoản, quản lý quyền truy cập, quản lý sản phẩm, quản lý tin tức, quản lý đơn hàng, quản lý tài khoản, quản lý danh mục, xem thống kê doanh thu (trong ngày, trong tháng, tổng doanh thu…) |
| Khách hàng | Là người mua hàng, được cung cấp một số chức năng như: Đăng nhập, đăng ký, lấy lại mật khẩu, xem sản phẩm, tìm kiếm sản phẩm theo tên, xem tin tức, thêm sản phẩm vào giỏ hàng, đặt hàng, xem chi tiết đơn hàng,… |

*Bảng 3.1. Danh sách tác nhân của hệ thống và các yêu cầu chức năng*

### **3.1.2. Yêu cầu phi chức năng**

- Giao diện trang web được thiết kế dễ sử dụng và có tính thẩm mỹ cao.

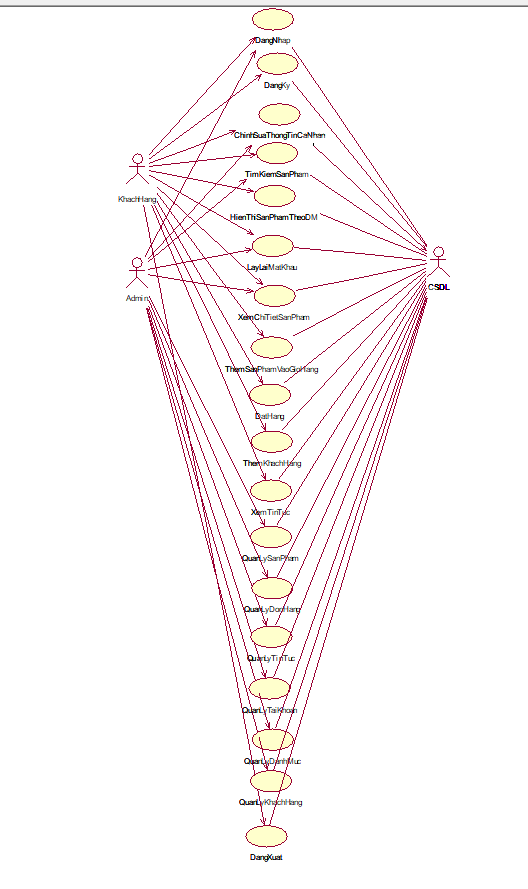
- Được viết trên nền tảng web.

- Hiệu năng: tốc độ, khả năng và độ tin cậy của hệ thống.

- Website bảo mật phân quyền để thực hiện các chức năng của hệ thống.

**3.2. Biểu đồ use case**

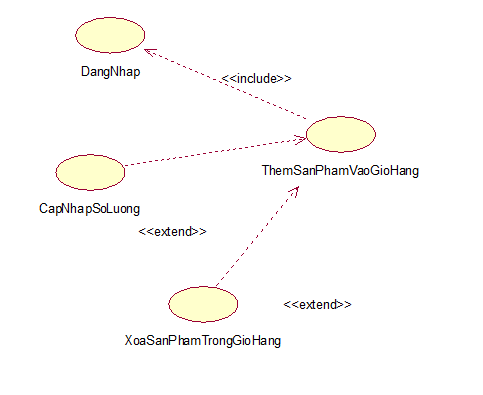
**3.2.1. Biểu đồ use case tổng quát**

****

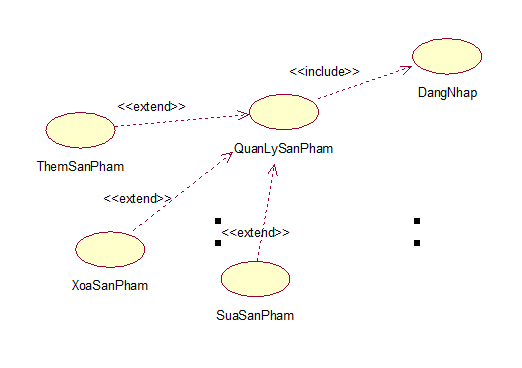
Hình 3.1. Biểu đồ use case tổng quát

### **3.2.2. Mô tả chi tiết quan hệ trong các use case**

**Thêm sản phẩm vào giỏ hàng**

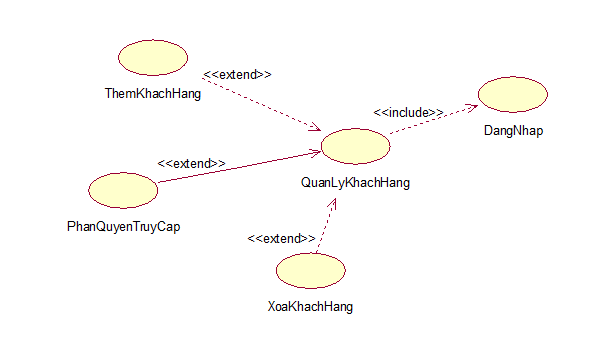
****

Hình 3.2. Quan hệ trong use case thêm sản phẩm vào giỏ hàng

**Quản lý sản phẩm**

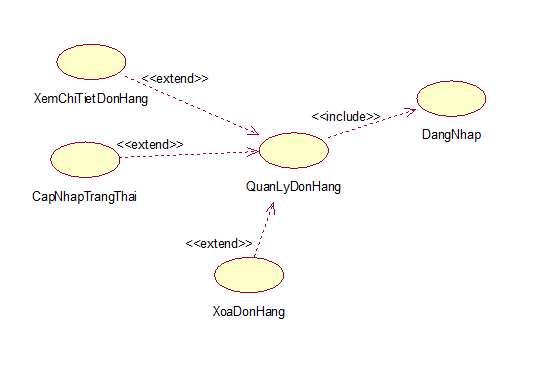
Hình 3.3. Quan hệ trong use case quản lý sản phẩm

**Quản lý khách hàng**

****

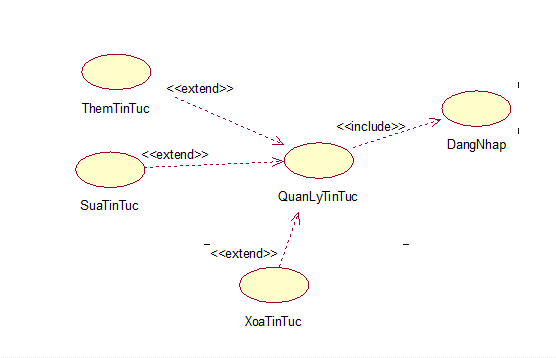
Hình 3.4. Quan hệ trong use case quản lý khách hàng

**Quản lý đơn hàng**

****

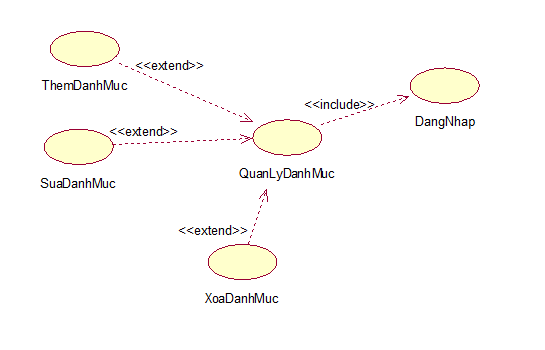
Hình 3.5. Quan hệ trong use case quản lý đơn hàng

**Quản lý tin tức**

****

Hình 3.6. Quan hệ trong use case quản lý tin tức

**Quản lý danh mục**

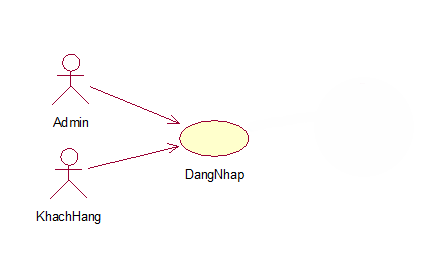
****

Hình 3.7. Quan hệ trong use case quản lý danh mục sản phẩm

## 3.3 Đặc tả chi tiết các use case

### **3.3.1 Use case đăng nhập**

a) Biểu đồ use case



Hình 3.8. Biểu đồ use case đăng nhập

b) Mô tả use case

Use case này cho phép người dùng (hoặc người quản lý) đăng nhập vào hệ thống.

* **Luồng sự kiện:**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng muốn đăng nhập vào hệ thống. Hệ thống sẽ hiển thị màn hình yêu cầu nhập tên đăng nhập và mật khẩu.
2. Người dùng nhập email và mật khẩu sau đó click vào nút “Đăng nhập”. Hệ thống kiểm tra email, mật khẩu trong bảng “KhachHang” và hiển thị màn hình hệ thống với các menu tương ứng với quyền. Use case kết thúc.
   * **Luồng rẽ nhánh:**
3. Tại bước 2 trong luồng cơ bản nếu người dùng không nhập email hoặc không nhập mật khẩu hệ thống sẽ hiển thị thông báo “Thông tin không được để trống”. Người dùng cần nhập đủ thông tin để tiếp tục.
4. Tại bước 2 trong luồng cơ bản nếu người dùng nhập sai tên hoặc mật khẩu thì hệ thống sẽ hiển thị thông báo lỗi “Thông tin tài khoản hoặc mật khẩu không chính xác!” và use case kết thúc.
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo “Lỗi kết nối!” và use case kết thúc.

* **Các yêu cầu đặc biệt:**

Người dùng cần có một tài khoản trước khi có thể đăng nhập.

* **Tiền điều kiện:**

Người dùng cần có một tài khoản trước khi có thể đăng nhập.

* **Hậu điều kiện:**

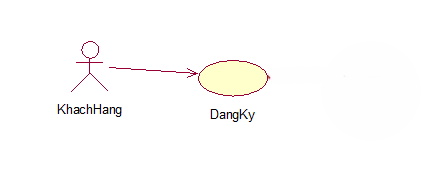
Không có.

* **Điểm mở rộng:**

Không có.

### **3.3.2 Use case đăng ký**

a) Biểu đồ use case



Hình 3.9. Biểu đồ use case đăng ký

b) Mô tả use case

Use case này cho phép người dùng đăng kí tạo tài khoản để sử dụng các chức năng khác của website.

* **Luồng sự kiện**

O ***Luồng cơ bản***

1. Use này bắt đầu khi người dùng kích vào “Đăng kí’ ở trong giao diện đăng nhập. Hệ thống sẽ hiển thi ra màn hình một form đăng kí bao gồm: email, số điện thoại, tên đăng nhập, mật khẩu, đia chỉ, họ và tên.
2. Người dùng điền thông tin đăng kí lên form đăng kí và nhất nút đăng kí, hệ thông sẽ kiểm tra thông tin mà người dùng nhập vào trong bảng “KhachHang”. Use case kết thúc.

o ***Luồng rẽ nhánh***

1. Tại bước 2 của luồng cơ bản nếu người dùng nhập không đầy đủ thông tin hoặc sai định dạng thì hệ thống sẽ yêu cầu nhâp lại.
2. Tại bước 2 của luồng cơ bản nếu người dùng nhập vào email hoặc số điện thoại đã tồn tại trong CSDL thì hệ thống sẽ đưa ra thông báo “Email đăng ký đã tồn tại” hoặc “Số điện thoại đã tồn tại”. Use case sẽ kết thúc.
3. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có

* **Tiền điều kiện:**

Người dùng chưa có tài khoản truy cập hệ thống

* **Hậu điều kiện:**

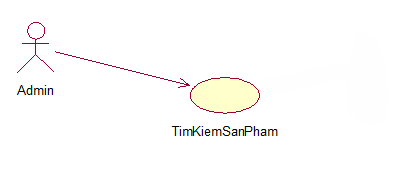
Nếu đăng kí thành công hệ thống sẽ hiển thị màn hình đăng nhập

* **Điểm mở rộng:**

Không có

### **3.3.3 Use case tìm kiếm sản phẩm theo tên**

a) Biểu đồ use case



Hình 3.10. Biểu đồ use case tìm kiếm sản phẩm theo tên

b) Mô tả use case

Use case này cho phép người dùng tìm kiếm sản phẩm theo tên mà người dùng nhập vào.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use này bắt đầu khi người dùng nhập tên sản phẩm cần tìm vào form tìm kiếm và kích vào nút “Tìm kiếm”.
  2. Hệ thống sẽ kiểm tra thông tin tên sản phẩm mà khách hàng đã nhập vào form từ bảng “SanPham” và hiển thị các sản phẩm, số lượng sản phẩm liên quan đến thông tin mà khách hàng nhập vào. Use case kết thúc.

o ***Luồng rẽ nhánh***

* 1. Tại bước 2 của luồng cơ bản nếu người dùng nhập không đầy đủ thông tin thì hệ thống sẽ yêu cầu nhâp lại. Người dùng nhập đầy đủ thông tin để tiếp tục
  2. Tại bước 2 của luồng cơ bản nếu không tìm thấy sản phẩm nào liên quan đến thông tin khách hàng nhập vào form thì hệ thống sẽ hiển thị thông báo “Không có sản phẩm nào phù hợp”, use case kết thúc.
  3. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Các yêu cầu đặc biệt:**

Không có

* **Tiền điều kiện:**

Không có

* **Hậu điều kiện:**

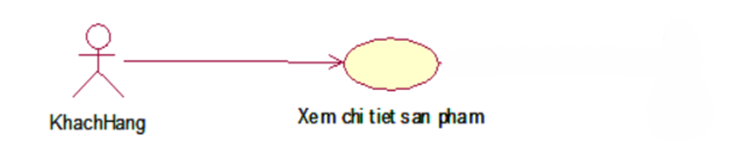
Không có

* **Điểm mở rộng:**

Không có

### **3.3.4 Use case xem chi tiết sản phẩm**

a) Biểu đồ use case



Hình 3.11. Biểu đồ use case xem chi tiết sản phẩm

b) Mô tả use case

Use case này cho phép khách hàng xem chi tiết sản phẩm mà khách hàng quan tâm.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use này bắt đầu khi khách hàng click vào nút “Thông Tin” khi kích chuột vào ảnh sản phẩm hoặc tên sản phẩm mà khách hàng quan tâm.
  2. Hệ thống sẽ lấy từ bảng “SanPham” ra các thông tin của sản phẩm bao gồm tên sản phẩm, giá bán sản phẩm, danh mục sản phẩm, mô tả sản phẩm, use case kết thúc.

o ***Luồng rẽ nhánh***

* 1. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Các yêu cầu đặc biệt:**

Không có

* **Tiền điều kiện:**

Không có

* **Hậu điều kiện:**

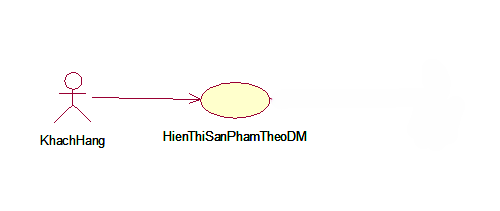
Không có

* **Điểm mở rộng:**

Không có

### **3.3.5. Use case hiển thị sản phẩm theo danh mục**

a) Biểu đồ use case



Hình 3.12. Biểu đồ use case hiển thị sản phẩm theo danh mục

b) Mô tả use case

Use case này cho phép người dùng hiển thị ra những sản phẩm thuộc danh mục mà khách hàng quan tâm.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use này bắt đầu khi khách hàng click vào tên của danh mục trên thanh Menu danh mục. Hệ thống sẽ lấy từ bảng “SanPham” các thông tin của sản phẩm như ảnh sản phẩm, tên sản phẩm, giá bán và hiển thị lên trên màn hình. Use case kết thúc.

o ***Luồng rẽ nhánh***

* 1. Tại bước 1 trong luồng cơ bản, nếu như danh mục mà khách hàng lựa chọn không có sản phẩm nào, hệ thống sẽ không hiển thị thông tin và use case kết thúc.
  2. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Các yêu cầu đặc biệt:**

Không có

* **Tiền điều kiện:**

Không có

* **Hậu điều kiện:**

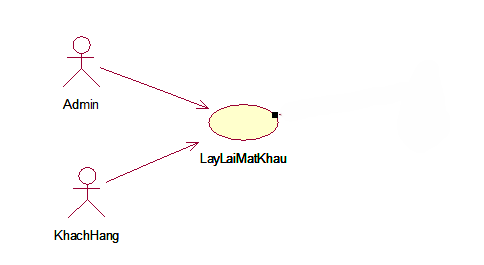
Không có

* **Điểm mở rộng:**

Không có

### **3.3.6. Use case lấy lại mật khẩu**

a) Biểu đồ use case



Hình 3.13. Biểu đồ use case lấy lại mật khẩu

b) Mô tả use case

Use case này cho phép người dùng (hoặc người quản lý) có thể lấy lại được mật khẩu thông qua email đã đăng ký.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use này bắt đầu khi khách hàng click vào Quên mật khẩu trên trang Login. Hệ thống sẽ hiển thị ra một form yêu cầu người dùng nhập email mà khách hàng dùng để đăng ký tài khoản.
  2. Khách hàng nhập thông tin email đăng ký vào form, sau đó click vào nút Send Reset Email. Hệ thống sẽ kiểm tra email mà khách hàng vừa nhập vào trong bảng “KhachHang” và gửi đến email đó một đường dẫn đến với trang đổi mật khẩu. Khách hàng kiểm tra email đăng ký của mình để lấy đường dẫn đến trang Reset Password.
  3. Khách hàng click vào đường dẫn, nhập thông tin mật khẩu mới vào form sau đó click nút Reset Password để đổi mật khẩu. Hệ thống sẽ cập nhập lại mật khẩu mới của khách hàng trong CSDL và hiển thị thông báo cập nhập thành công. Use case kết thúc.

o ***Luồng rẽ nhánh***

* 1. Tại bước 2 trong luồng cơ bản, nếu như khách hàng không nhập thông tin email vào form hoặc email của khách hàng không có trong CSDL thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu khách hàng nhập lại.
  2. Nếu khách hàng không vào đúng đường link mà hệ thông gửi đến email thì màn hình sẽ hiển thị ra thông báo lỗi. Use case kết thúc.
  3. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Các yêu cầu đặc biệt:**

Không có

* **Tiền điều kiện:**

Khách hàng phải nhập vào đúng email đã đăng ký tài khoản.

* **Hậu điều kiện:**

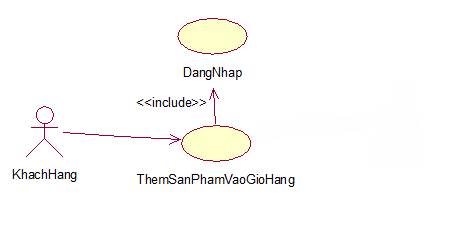
Không có

* **Điểm mở rộng:**

Không có

### **3.3.7. Use case thêm sản phẩm vào giỏ hàng**

a) Biểu đồ use case:



Hình 3.14. Biểu đồ use case thêm sản phẩm vào giỏ hàng

b) Đặc tả use case:

Use case này cho phép thêm sản phẩm muốn mua vào giỏ hàng.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use này bắt đầu khi khách kích vào “Thêm vào giỏ hàng” trên mỗi sản phẩm. Hệ thống hiển thị danh sách các sản phẩm trong giỏ hàng bao gồm tên sản phẩm, số lượng, đơn giá, tổng tiền và xóa sản phẩm (nếu muốn xóa). Use case kết thúc.

o ***Luồng rẽ nhánh***

* 1. Tại bước 1 luồng cơ bản, nếu khách hàng muốn tăng giảm số lượng sản phẩm có thể kích nút “+” và “-” trên cột số lượng của mỗi sản phẩm. Hệ thống sẽ hiển thị số lượng tương ứng sau khi thay đổi và cập nhập lại tổng tiền lên màn hình.
  2. Tại bước 1 luồng cơ bản, nếu khách hàng muốn xóa sản phẩm ra khỏi giỏ hàng, có thể kích nút “Xóa” trên cột xóa của mỗi sản phẩm. Hệ thống cập nhật và hiển thị danh sách sản phẩm trong giỏ hàng sau khi xóa.
  3. Nếu khách hàng muốn xóa các sản phẩm ra khỏi giỏ hàng có thể kích “Xóa tất cả”. Hệ thống sẽ hiển thị thông báo “Bạn không có sản phẩm nào trong giỏ hàng”.
  4. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Các yêu cầu đặc biệt:**

Không có

* **Tiền điều kiện:**

Khách hàng phải đăng nhập vào hệ thống

* **Hậu điều kiện:**

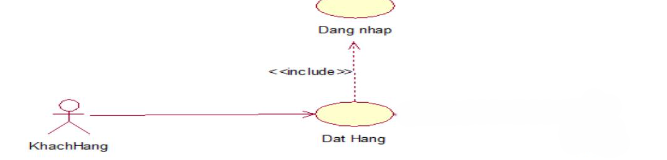
Không có

* **Điểm mở rộng:**

Không có

### **3.3.8. Use case đặt hàng**

a) Biểu đồ use case:



Hình 3.15. Biểu đồ use case đặt hàng

b) Đặc tả use case:

Use case này cho phép khách hàng đặt hàng.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use case này bắt đầu khi người dùng kích vào biểu tượng giỏ hàng trên thanh menu. Hệ thống hiển thị hiển thị danh sách các sản phẩm trong giỏ hàng bao gồm tên sản phẩm, số lượng, đơn giá, tổng tiền và xóa sản phẩm (nếu muốn xóa) và nút “Đặt Hàng”
  2. Người dùng kích vào “Đặt hàng”. Hệ thống hiển thị một biểu mẫu thông tin người nhận bao gồm họ và tên, email, địa chỉ nhận hàng, số điện thoại, một bảng danh sách sản phẩm muốn mua bao gồm tên sản phẩm, số lượng, đơn giá, tổng tiền lên màn hình.
  3. Người dùng có thể thay đổi thông tin trên biểu mẫu và kích vào “Mua Hàng”. Hệ thống sẽ hiển thị lên thông báo “Đặt hàng thành công” và đưa khách hàng đến trang Đơn hàng. Use case kết thúc.

o ***Luồng rẽ nhánh***

* 1. Tại bước 1 trong luồng cơ bản nếu như trong giỏ hàng của khách hàng không có sản phẩm nào thì khách hàng không thể đặt được hàng. Use case kết thúc.
  2. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Yêu cầu đặc biệt:**

Giỏ hàng phải có sản phẩm

* **Tiền điều kiện:**

Khách hàng phải đăng nhập vào hệ thống

* **Hậu điều kiện:**

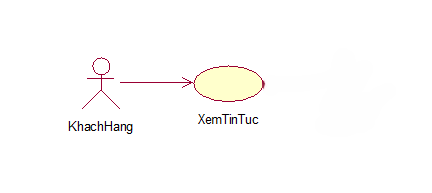
Không có

* **Điểm mở rộng:**

Không có

### **3.3.9. Use case xem tin tức**

a) Biểu đồ use case:



Hình 3.16. Biểu đồ use case xem tin tức

b) Đặc tả use case:

Use case này cho phép khách hàng xem những tin tức về cửa hàng.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use case này bắt đầu khi người dùng kích vào Tin Tức trên thanh menu. Hệ thống hiển thị hiển thị ra một danh sách các tin tức có trong bảng TinTuc và các thông tin liên quan như ảnh minh họa, tiêu đề, tác giả, ngày đăng lên trên màn hình.
  2. Người dùng kích vào tiêu đề của tin tức cần đọc. Hệ thống hiển thị chi tiết hơn các thông tin như nội dung, tiêu tiêu đề, tác giả, ngày đăng. Use case kết thúc.

o ***Luồng rẽ nhánh***

* 1. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Yêu cầu đặc biệt:**

Không có

* **Tiền điều kiện:**

Không có

* **Hậu điều kiện:**

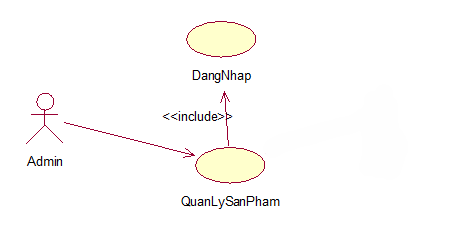
Không có

* **Điểm mở rộng:**

Không có

### **3.3.10. Use case quản lý sản phẩm**

a) Biểu đồ use case:



Hình 3.17. Biểu đồ use case quản lý sản phẩm

b) Đặc tả use case:

Use case này cho phép người dùng thêm, sửa, xóa, xem chi tiết sản phẩm trong hệ thống.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use case này bắt đầu khi người dùng kích vào “Danh sách sản phẩm” trên menu quản trị. Hệ thống sẽ lấy thông tin của sản phẩm gồm: ảnh, tên sản phẩm, danh mục sản phẩm, đơn giá, trạng thái sản phảm từ bảng SanPham trong cơ sở dữ liệu và hiển thị danh sách các sản phẩm lên màn hình.
  2. Xem sản phẩm

Khi người dùng kích vào nút “Xem” trên một dòng sản phẩm. Hệ thống sẽ lấy thông tin chi tiết về sản phẩm gồm: ảnh sản phẩm, tên sản phẩm, tên danh mục, giá bán, mô tả sản phẩm, ngày tạo, ngày sửa, trạng thái sản phẩm từ bảng SanPham trong cơ sở dữ liệu và hiển thị chi tiết sản phẩm lên màn hình.

* 1. Sửa sản phẩm

Người dùng kích vào nút “Sửa” trong trang Chi tiết sản phẩm. Hệ thống lấy thông tin cũ của sản phẩm được chọn gồm: tên sản phẩm, danh mục sản phẩm, mô tả sản phẩm, hình ảnh, giá sản phẩm, trạng thái sản phẩm, ngày tạo, ngày sửa từ bảng SanPham và hiển thị lên màn hình. Người dùng nhập thông tin cần sửa và kích nút “Sửa”, hệ thống sẽ nhận các thông tin nhận vào và sửa nó lên trên hệ thống, sau đó hiển thị sản phẩm đã sửa.

* 1. Xóa sản phẩm

Người dùng kích vào nút “Xóa” trong trang chi tiết sản phẩm. Hệ thống sẽ hiển thị thông tin sản phẩm mà khách hàng muốn xóa, người dùng kích vào nút “Xóa”, hệ thống sẽ hiển thị ra thông báo xác nhận xóa, người dùng ấn nút “OK”. Hệ thống sẽ xóa bản ghi về sản phẩm đó trong bảng SanPham và hiển thị lên màn hình danh sách các sản phẩm sau khi đã cập nhật.

* 1. Thêm sản phẩm

Người dùng kích vào nút “Thêm sản phẩm” trên cửa sổ danh sách các sản phẩm. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho sản phẩm gồm: tên sản phẩm, tên danh mục, trạng thái sản phẩm, mô tả sản phẩm, ngày tạo, hình ảnh, tự sinh một mã sản phẩm mới. Người dùng kích vào nút “Thêm”, hệ thống sẽ tạo thêm một bản ghi mới trong bảng SanPham và hiển thị danh sách các sản phẩm sau khi đã được thêm.

o ***Luồng rẽ nhánh***

* 1. Tại bước 3 hoặc 5 trong luồng cơ bản, nếu người dùng nhập thông tin không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại, người dùng có thể nhập lại hoặc hủy bỏ để kết thúc.
  2. Tại bước 4 trong luồng cơ bản, nếu người dùng click nút “Hủy” trên thanh thông báo, hệ thống sẽ dừng việc xóa sản phẩm. Use case kết thúc.
  3. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Yêu cầu đặc biệt:**

Use case này chỉ cho phép người quản lý thực hiện

* **Tiền điều kiện:**

Người quản lý phải đăng nhập vào hệ thống

* **Hậu điều kiện:**

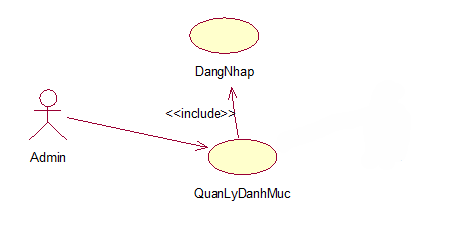
Không có

* **Điểm mở rộng:**

Không có

### **3.3.11. Use case quản lý danh mục**

a) Biểu đồ use case:



Hình 3.18. Biểu đồ use case quản lý danh mục

b) Đặc tả use case:

Use case này cho phép người dùng thêm, sửa, xóa, xem chi tiết danh mục trong hệ thống.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use case này bắt đầu khi người dùng kích vào “Danh sách danh mục” trên menu quản trị. Hệ thống sẽ lấy thông tin của các danh mục gồm: ảnh danh mục, tên danh mục, trạng thái của danh mục, mô tả danh mục trong cơ sở dữ liệu và hiển thị danh sách các danh mục lên màn hình.
  2. Xem danh mục

Khi người dùng kích vào nút “Xem” trên một dòng danh mục. Hệ thống sẽ lấy thông tin chi tiết về danh mục gồm: mã danh mục, ảnh danh mục, tên danh mục, mô tả danh mục, trạng thái danh mục từ bảng DanhMuc trong cơ sở dữ liệu và hiển thị chi tiết danh mục cần xem lên màn hình.

* 1. Sửa danh mục

Người dùng kích vào nút “Sửa” trong trang Chi tiết danh mục. Hệ thống lấy thông tin cũ của danh mục được chọn gồm: mã danh mục, ảnh danh mục, tên danh mục, mô tả danh mục, trạng thái danh mục từ bảng DanhMuc trong cơ sở dữ liệu và hiển thị lên màn hình. Người dùng nhập thông tin cần sửa và kích nút “Sửa”, hệ thống sẽ nhận các thông tin nhận vào và cập nhập nó lên trên hệ thống, sau đó hiển thị danh mục đã sửa.

* 1. Xóa danh mục

Người dùng kích vào nút “Xóa” trong trang Chi tiết sản danh mục. Hệ thống sẽ hiển thị thông tin danh mục mà khách hàng muốn xóa, người dùng kích vào nút “Xóa”, hệ thống sẽ hiển thị ra thông báo xác nhận xóa, người dùng ấn nút “OK”. Hệ thống sẽ xóa bản ghi về danh mục đó trong bảng DanhMuc và hiển thị lên màn hình danh sách các danh mục sau khi đã cập nhật.

* 1. Thêm danh mục

Người dùng kích vào nút “Thêm danh mục” trên cửa sổ danh sách các sản phẩm. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho sản phẩm gồm: ảnh danh mục, tên danh mục, mô tả danh mục, trạng thái danh mục tự sinh một mã sản phẩm mới người dùng kích vào nút “Thêm”, hệ thống sẽ tạo thêm một bản ghi mới trong bảng DanhMuc và hiển thị danh sách các danh mục sau khi đã được thêm.

o ***Luồng rẽ nhánh***

* 1. Tại bước 4 trong luồng cơ bản, nếu người dùng click nút “Hủy” trên thanh thông báo, hệ thống sẽ dừng việc xóa danh mục. Use case kết thúc.
  2. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Yêu cầu đặc biệt:**

Use case này chỉ cho phép người quản lý thực hiện

* **Tiền điều kiện:**

Người quản lý phải đăng nhập vào hệ thống

* **Hậu điều kiện:**

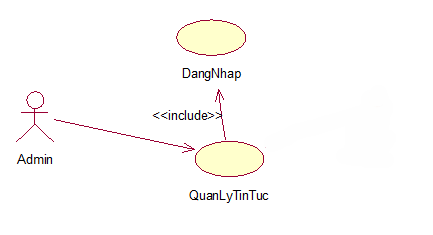
Không có

* **Điểm mở rộng:**

Không có

### **3.3.12. Use case quản lý tin tức**

a) Biểu đồ use case:



Hình 3.19. Biểu đồ use case quản lý tin tức

b) Đặc tả use case:

Use case này cho phép người dùng thêm, sửa, xóa, xem chi tiết tin tức trong hệ thống.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use case này bắt đầu khi người dùng kích vào “Danh sách tin tức” trên menu quản trị. Hệ thống sẽ lấy thông tin của tin tức gồm: ảnh tin tức, tiêu đề tin tức, tác giả, ngày tạo và hiển thị danh sách tin tức lên màn hình.
  2. Xem tin tức

Khi người dùng kích vào nút “Xem” trên một dòng tin tức. Hệ thống sẽ lấy thông tin chi tiết về tin tức gồm: ảnh tin tức, tiêu đề tin tức, tác giả, ngày đăng, mô tả ngắn, mô tả chi tiết từ bảng TinTuc trong cơ sở dữ liệu và hiển thị chi tiết danh mục cần xem lên màn hình.

* 1. Sửa tin tức

Người dùng kích vào nút “Sửa” trong trang Chi tiết tin tức. Hệ thống lấy thông tin cũ của tin tức được chọn gồm: ảnh tin tức, tiêu đề tin tức, tác giả, ngày đăng, mô tả ngắn, mô tả chi tiết từ bảng TinTuc trong cơ sở dữ liệu và hiển thị chi tiết danh mục cần xem lên form sửa. Người dùng nhập thông tin cần sửa và kích nút “Sửa”, hệ thống sẽ nhận các thông tin nhận vào và cập nhập nó lên trên hệ thống, sau đó hiển thị tin tức đã sửa.

* 1. Xóa tin tức

Người dùng kích vào nút “Xóa” trong trang Chi tiết tin tức. Hệ thống sẽ hiển thị thông tin tin mà khách hàng muốn xóa, người dùng kích vào nút “Xóa”, hệ thống sẽ hiển thị ra thông báo xác nhận xóa, người dùng ấn nút “OK”. Hệ thống sẽ xóa bản ghi về danh mục đó trong bảng TinTuc và hiển thị lên màn hình danh sách các danh mục sau khi đã cập nhật.

* 1. Thêm tin tức

Người dùng kích vào nút “Thêm tin tức” trên cửa sổ danh sách các sản phẩm. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho sản phẩm gồm: ảnh danh mục, tên danh mục, mô tả danh mục, trạng thái danh mục tự sinh một mã sản phẩm mới. Người dùng kích vào nút “Thêm”, hệ thống sẽ tạo thêm một bản ghi mới trong bảng DanhMuc và hiển thị danh sách các danh mục sau khi đã được thêm.

o ***Luồng rẽ nhánh***

* 1. Tại bước 4 trong luồng cơ bản, nếu người dùng click nút “Hủy” trên thanh thông báo, hệ thống sẽ dừng việc xóa danh mục. Use case kết thúc.
  2. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Yêu cầu đặc biệt:**

Use case này chỉ cho phép người quản lý thực hiện

* **Tiền điều kiện:**

Người quản lý phải đăng nhập vào hệ thống

* **Hậu điều kiện:**

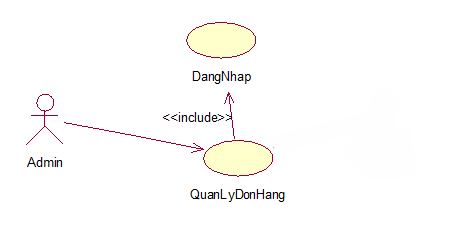
Không có

* **Điểm mở rộng:**

Không có

### **3.3.13. Use case quản lý đơn hàng**

a) Biểu đồ use case:



Hình 3.20. Biểu đồ use case quản lý đơn hàng

b) Đặc tả use case:

Use case này cho phép người dùng cập nhập trạng thái, xóa, xem chi tiết đơn hàng.

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use case này bắt đầu khi người dùng kích vào “Danh sách đơn hàng” trên menu quản trị. Hệ thống sẽ lấy hiển thị ra 4 loại đơn hàng gồm: Đơn hàng mới, đơn hàng đã xác nhận, đơn hàng đã hủy, đơn hàng đã giao lên trên màn hình. Người dùng chọn 1 trong 4 loại đơn hàng để thực hiện các chức năng.
  2. Cập nhập trạng thái đơn hàng

Khi người dùng kích vào 2 loại đơn hàng (đơn hàng mới, đơn hàng đã xác nhận). Hệ thống sẽ lấy thông tin chi tiết về tin tức gồm: Mã đơn hàng, tên người đặt, ngày đặt, trạng thái đơn hàng từ bảng DonHang trong cơ sở dữ liệu và hiển thị lên trên màn hình. Người dùng có thể lựa chọn các nút Xác nhận, Hủy đơn, Đã giao theo yêu cầu của mình, hệ thống sẽ cập nhập lại trạng thái đơn hàng theo yêu cầu của người dùng và hiển thị lên trên màn hình sau khi cập nhập.

* 1. Xem chi tiết đơn hàng

Người dùng kích vào nút “Xem chi tiết” của 1 trong 4 loại đơn hàng Hệ thống lấy thông tin của của đơn hàng được chọn gồm: mã đơn hàng, ngày đặt hàng, mã khách hàng, tên sản phẩm, số lượng sản phẩm, tổng tiền, tổng thanh toán, trạng thái đơn hàng, trạng thái thanh toán, thông tin người nhận từ 2 bảng DonHang và ChiTietDonHang trong cơ sở dữ liệu và hiển thị lên màn hình.

* 1. Xóa đơn hàng

Người dùng kích vào nút “Xóa” trong trang Đơn hàng hủy, hệ thống sẽ hiển thị ra thông báo xác nhận xóa, người dùng ấn nút “OK”. Hệ thống sẽ xóa bản ghi về đơn hàng đó trong bảng DonHang và bảng ChiTietDonHang sau đó hiển thị lên màn hình danh sách các đơn hàng còn lại trong DonHang.

o ***Luồng rẽ nhánh***

* 1. Tại bước 4 trong luồng cơ bản, nếu người dùng click nút “Hủy” trên thanh thông báo, hệ thống sẽ dừng việc xóa danh mục. Use case kết thúc.
  2. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Yêu cầu đặc biệt:**

Use case này chỉ cho phép người quản lý thực hiện

* **Tiền điều kiện:**

Người quản lý phải đăng nhập vào hệ thống

* **Hậu điều kiện:**

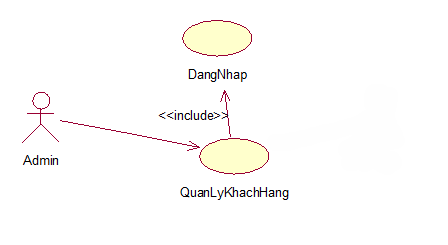
Không có

* **Điểm mở rộng:**

Không có

### **3.3.14. Use case quản lý khách hàng**

a) Biểu đồ use case:



Hình 3.21. Biểu đồ use case quản lý khách hàng

b) Đặc tả use case:

Use case này cho phép người dùng thêm, xóa, xem chi tiết thông tin của khách hàng đăng ký

* **Luồng sự kiện**

O ***Luồng cơ bản***

* 1. Use case này bắt đầu khi người dùng kích vào “Danh sách khách hàng” trên menu quản trị. Hệ thống sẽ hiển thị lên màn hình thông tin của khách hàng: Tên khách hàng, ngày sinh, giới tính, email, số điện thoại, ngày tạo từ bảng KhachHang trên CSDL.
  2. Xem chi tiết khách hàng

Khi người dùng kích vào nút “Xem” trên trang danh sách khách hàng. Hệ thống sẽ lấy thông tin chi tiết về khách hàng gồm tên khách hàng, ngày sinh, giới tính, email, mật khẩu, số điện thoại, địa chỉ, ngày tạo, từ bảng KhachHang trong cơ sở dữ liệu và hiển thị chi tiết danh mục cần xem lên màn hình.

* 1. Thêm khách hàng

Người dùng kích vào nút “Thêm khách hàng” trên cửa sổ danh sách các sản phẩm. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho khách hàng gồm: tên khách hàng, ngày sinh, giới tính, địa chỉ, điện thoại, email, mật khẩu và tự sinh một mã khách hàng mới. Người dùng kích vào nút “Thêm”, hệ thống sẽ tạo thêm một bản ghi mới trong bảng KhachHang và hiển thị danh sách khách hàng sau khi đã được thêm.

* 1. Xóa khách hàng

Người dùng kích vào nút “Xóa” trong trang Danh sách khách hàng, hệ thống sẽ hiển thị ra thông báo xác nhận xóa, người dùng ấn nút “OK”. Hệ thống sẽ xóa bản ghi về khách hàng đó và hiển thị lên màn hình danh sách các khách hàng còn lại sau khi xóa.

o ***Luồng rẽ nhánh***

* 1. Tại bước 3 trong luồng cơ bản, nếu người dùng để trống thông tin khách hàng hoặc nhập sai định dạng thông tin khách hàng thì hệ thống sẽ yêu cầu người dùng nhập lại.
  2. Tại bước 3 trong luồng cơ bản, nếu người dùng nhập vào một email hoặc số điện thoại đã tồn tại ở trong bảng KhachHang thì hệ thống sẽ đưa ra thông báo lỗi và yêu cầu người dùng nhập lại để tiếp tục.
  3. Tại bước 4 trong luồng cơ bản, nếu người dùng click nút “Hủy” trên thanh thông báo, hệ thống sẽ dừng việc xóa danh mục. Use case kết thúc.
  4. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* **Yêu cầu đặc biệt:**

Use case này chỉ cho phép người quản lý thực hiện

* **Tiền điều kiện:**

Người quản lý phải đăng nhập vào hệ thống

* **Hậu điều kiện:**

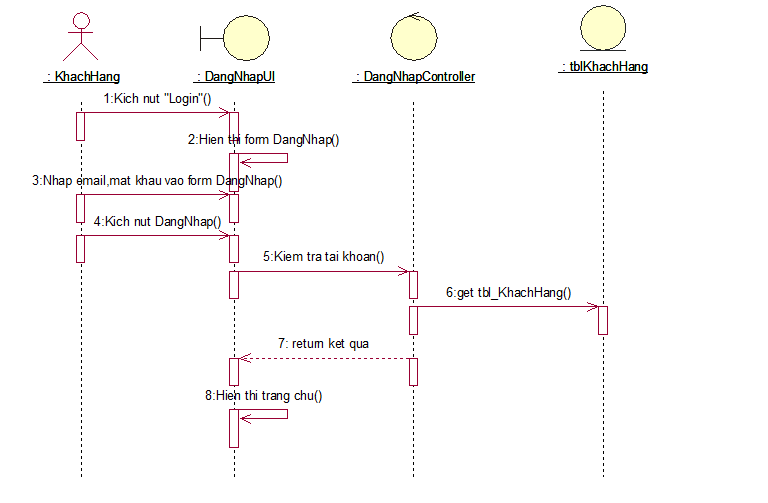
Không có

* **Điểm mở rộng:**

Không có

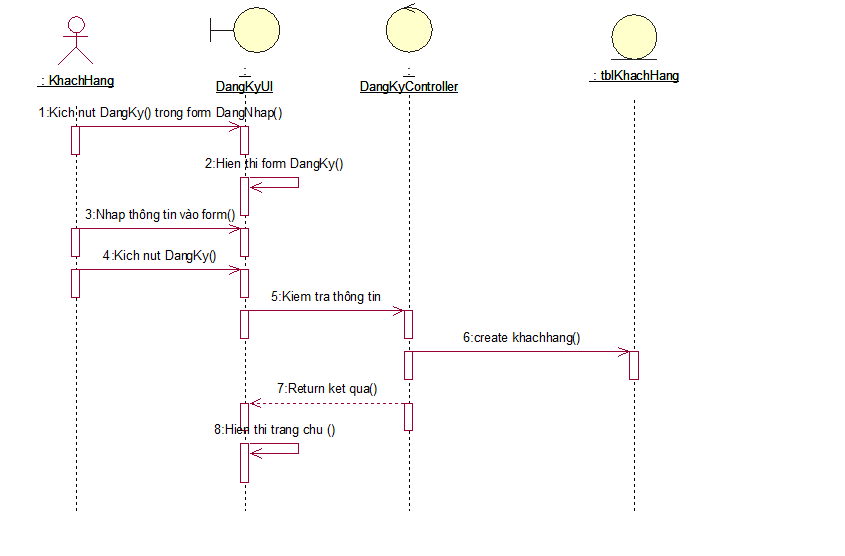
## 3.4 Phân tích use case

### **3.4.1. Biểu đồ trình tự chức năng đăng nhập**



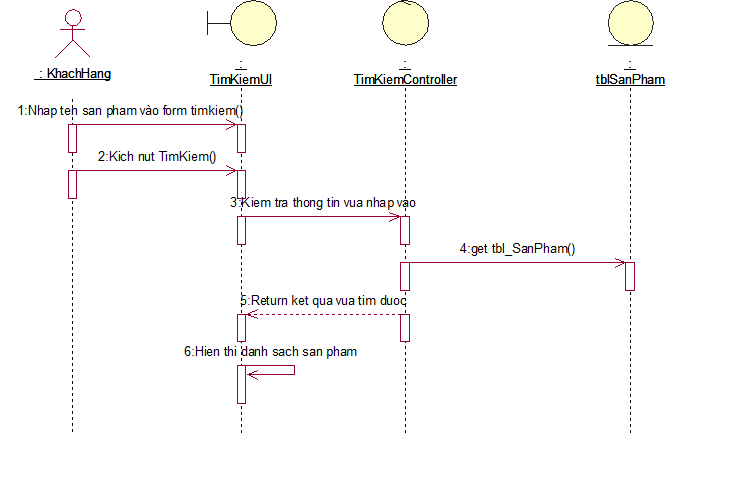
Hình 3.22. Biểu đồ trình tự chức năng đăng nhập

### **3.4.2 Biểu đồ trình tự chức năng đăng ký**



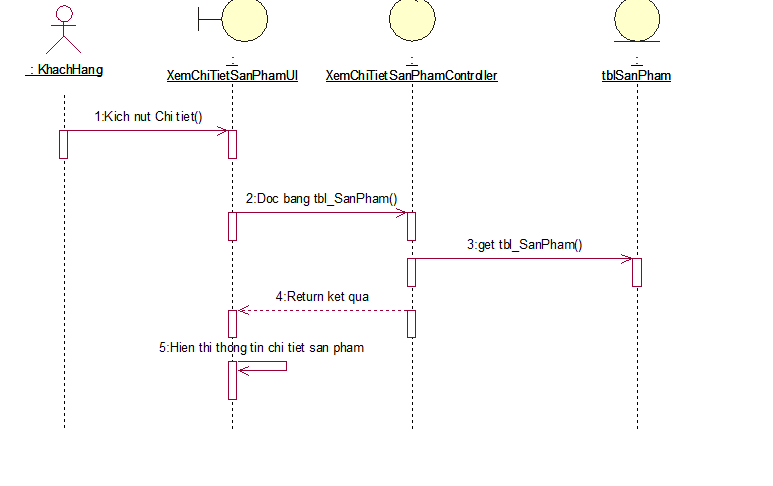
3.23. Biểu đồ trình tự chức năng đăng ký

### **3.4.3 Biểu đồ trình tự chức năng tìm kiếm sản phẩm theo tên**



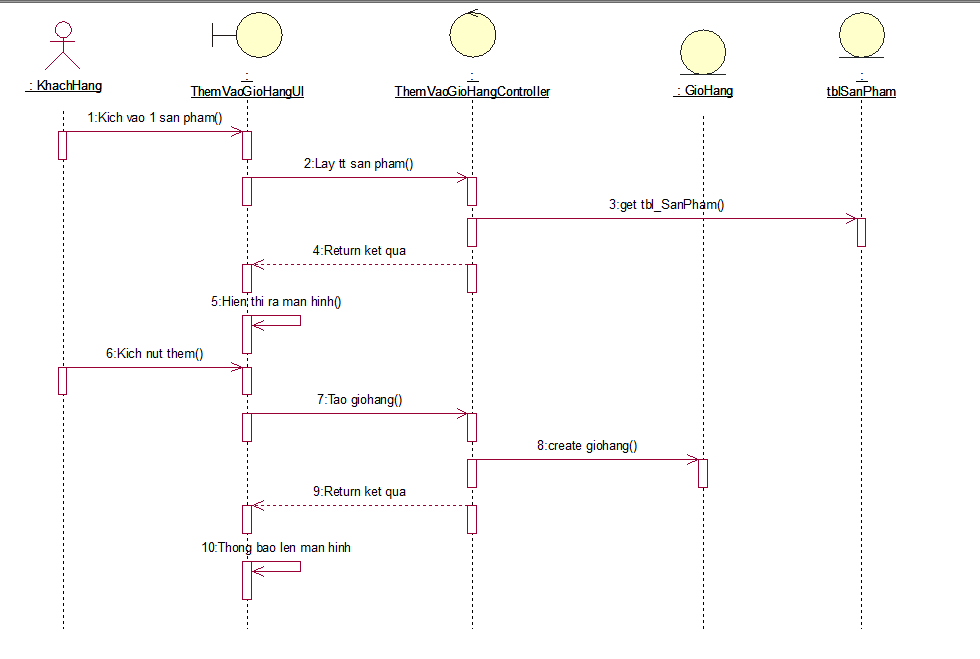
3.24. Biểu đồ trình tự chức năng tìm kiếm sản phẩm theo tên

### **3.4.4 Biểu đồ trình tự chức năng xem chi tiết sản phẩm**



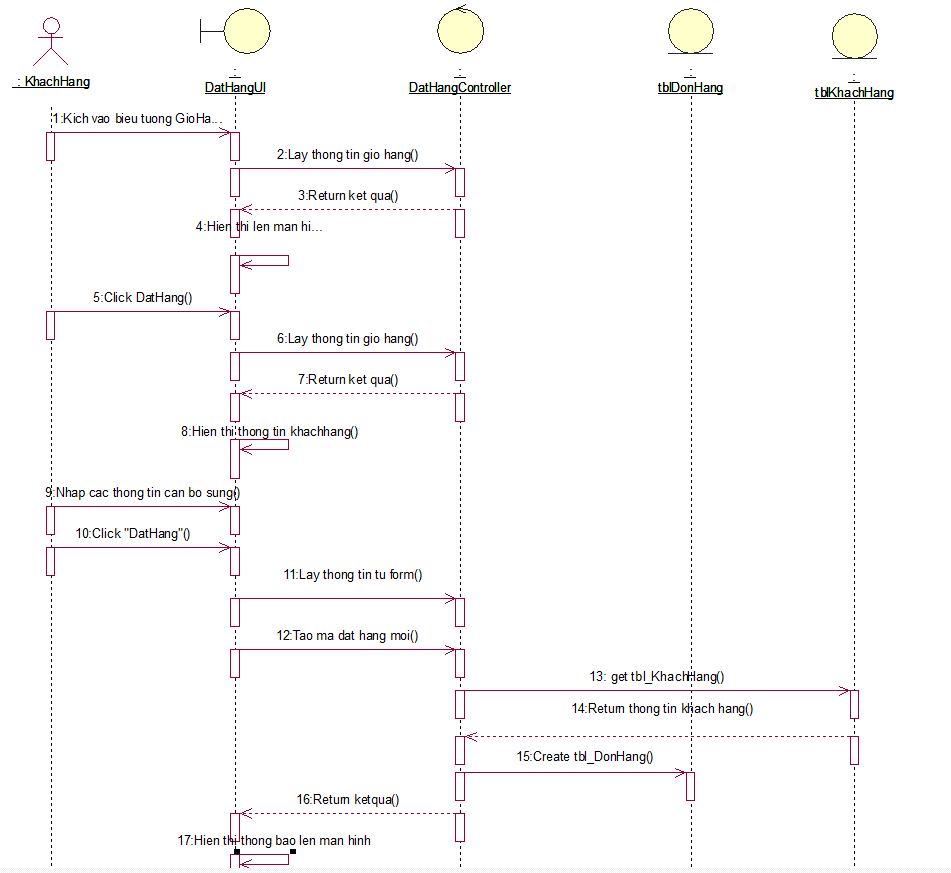
3.25. Biểu đồ trình tự chức năng xem chi tiết sản phẩm

### **3.4.5 Biểu đồ trình tự chức năng thêm sản phẩm vào giỏ hàng**



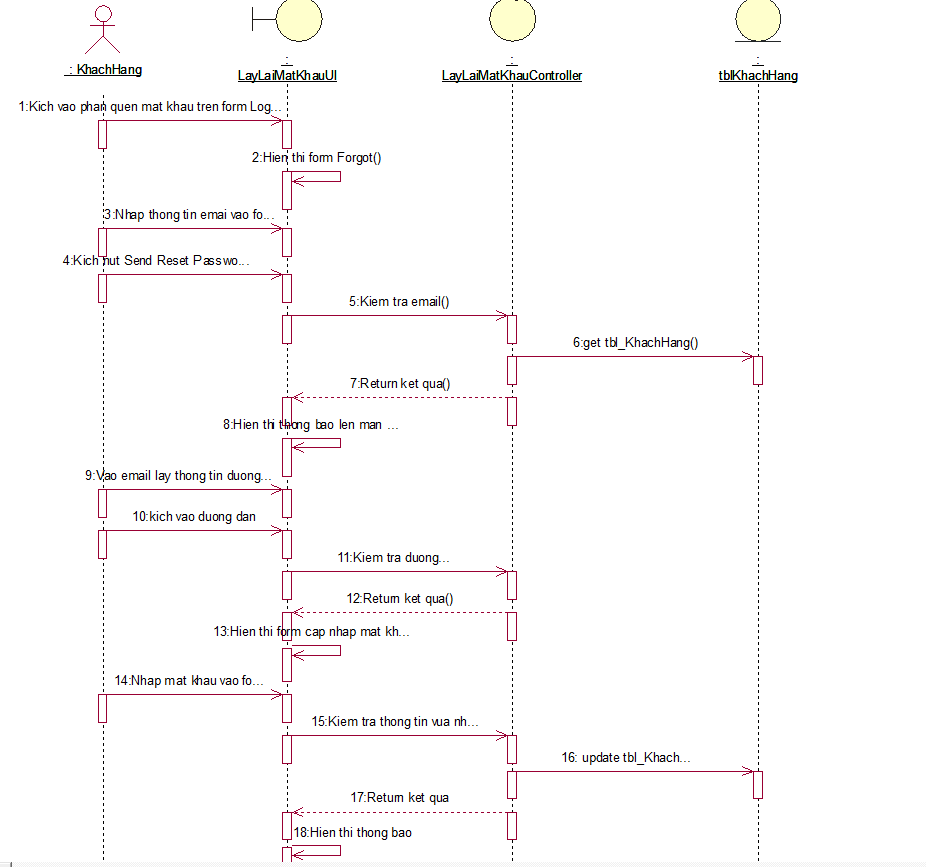
3.26. Biểu đồ trình tự chức năng thêm sản phẩm vào giỏ hàng

### **3.4.6 Biểu đồ trình tự chức năng đặt hàng**



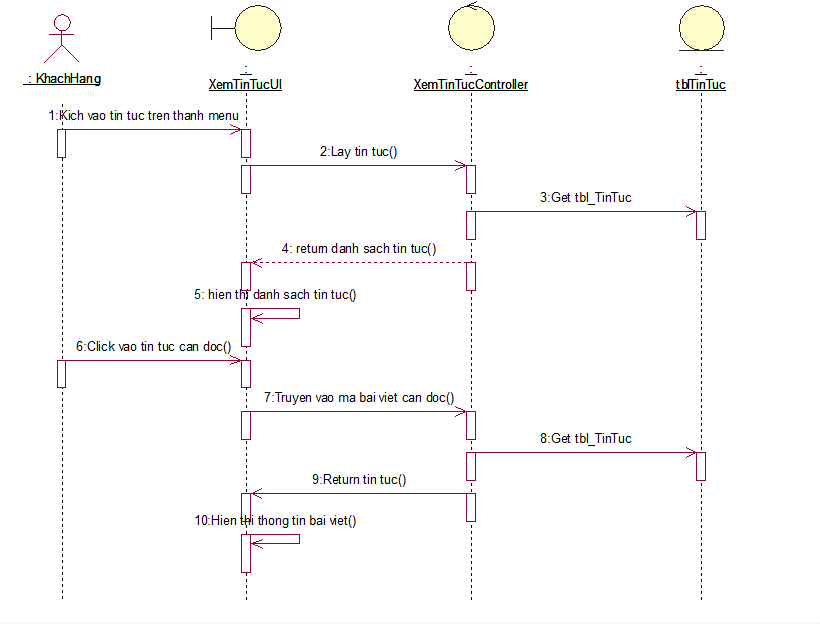
Hình 3.27. Biểu đồ trình tự chức năng Đặt hàng

### **3.4.7 Biểu đồ trình tự chức năng lấy lại mật khẩu**



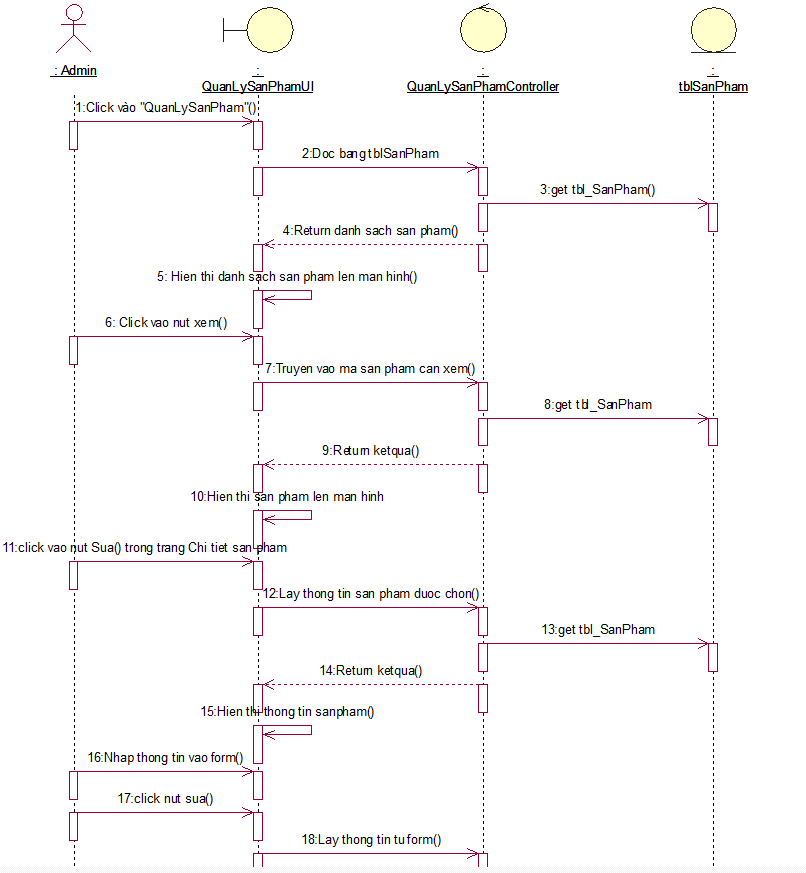
Hình 3.28. Biểu đồ trình tự chức năng lấy lại mật khẩu

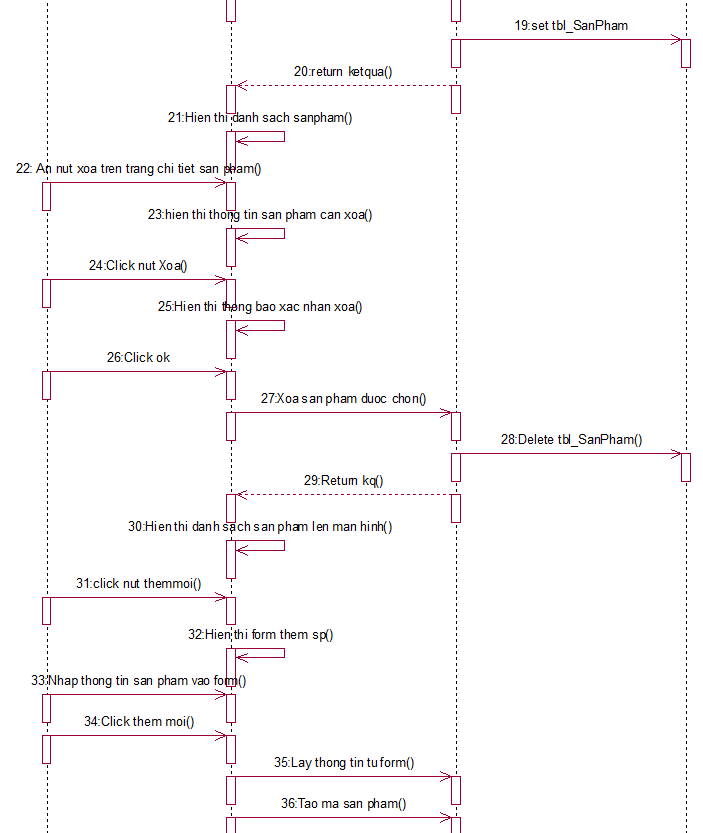
### **3.4.8 Biểu đồ trình tự chức năng xem tin tức**

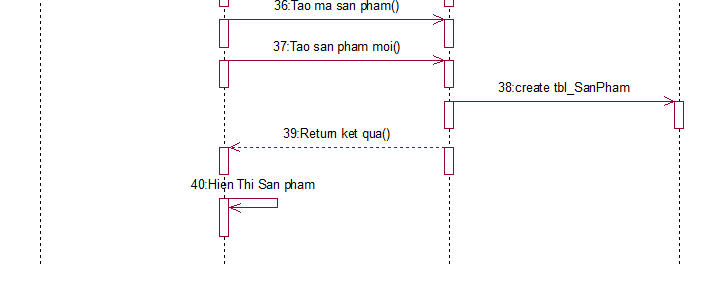


Hình 3.29. Biểu đồ trình tự chức năng xem tin tức

### **3.4.9 Biểu đồ trình tự chức năng quản lý sản phẩm**

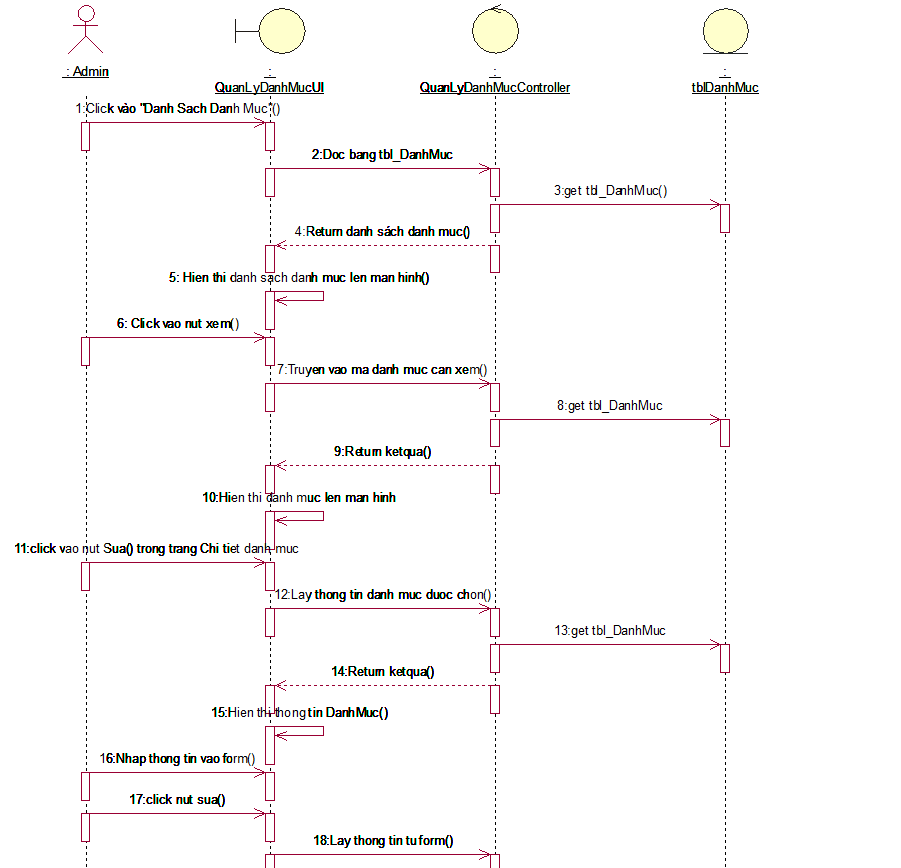


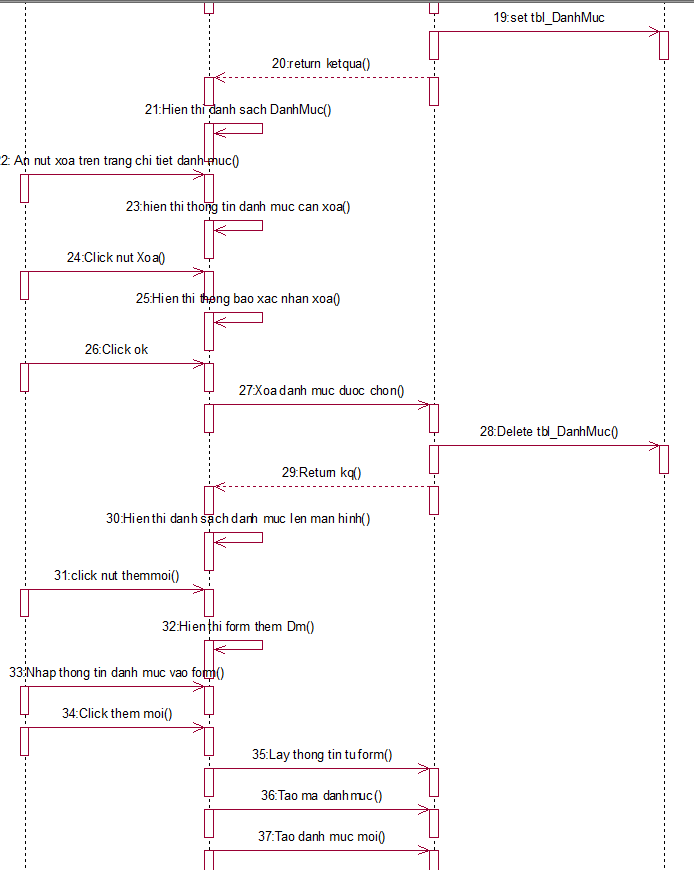


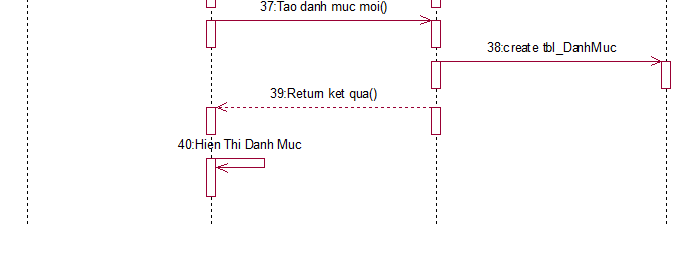


Hình 3.30. Biểu đồ trình tự chức năng quản lý sản phẩm

### **3.4.10 Biểu đồ trình tự chức năng quản lý danh mục**

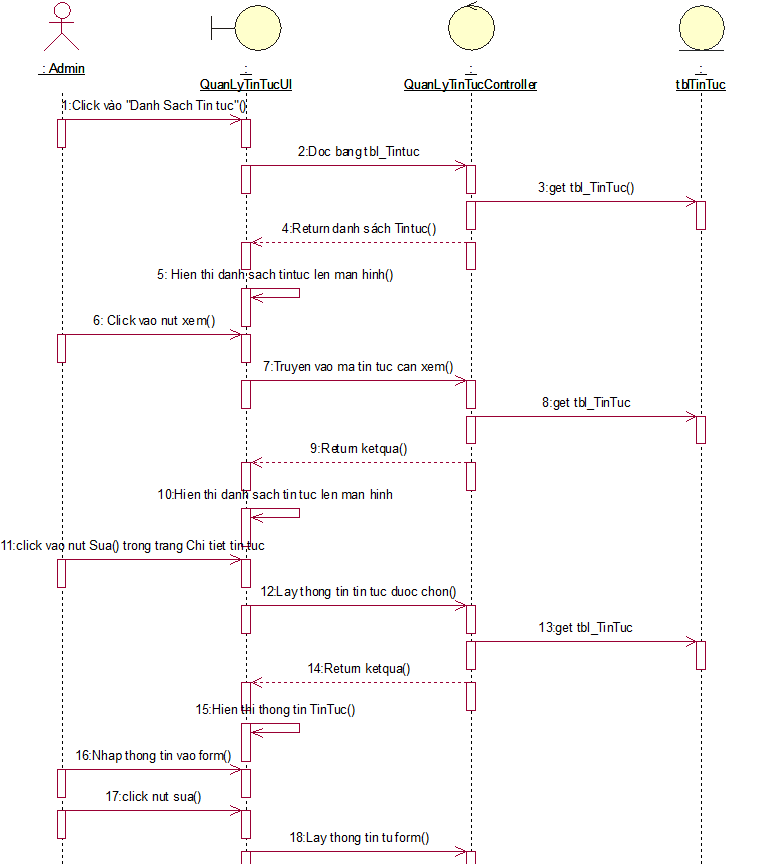


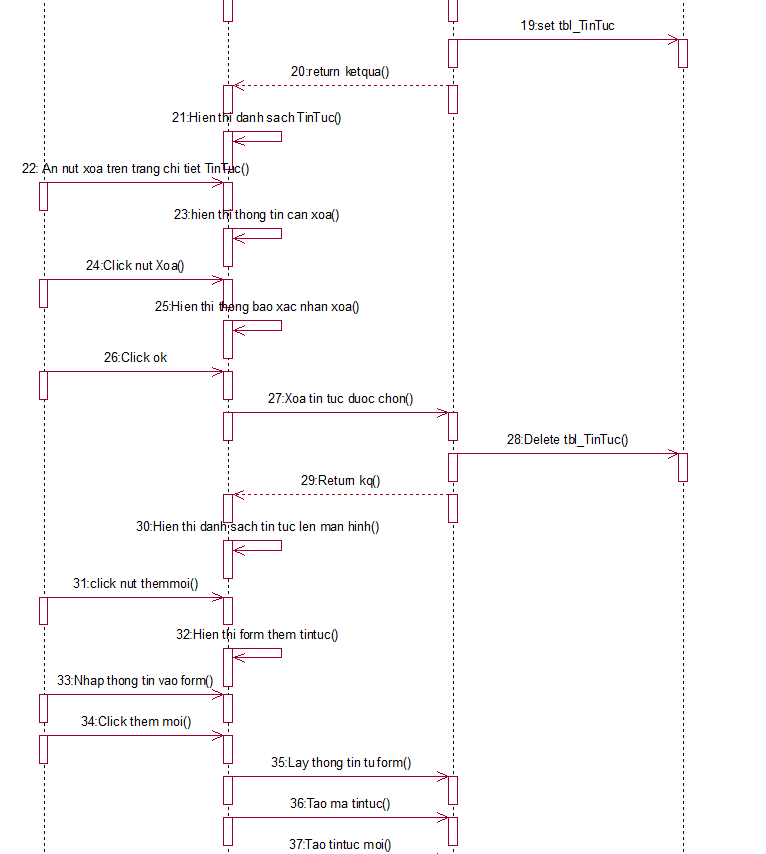


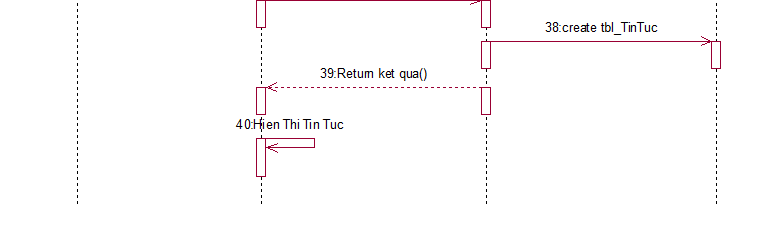


Hình 3.31. Biểu đồ trình tự chức năng quản lý danh mục

### **3.4.11 Biểu đồ trình tự chức năng quản lý tin tức**

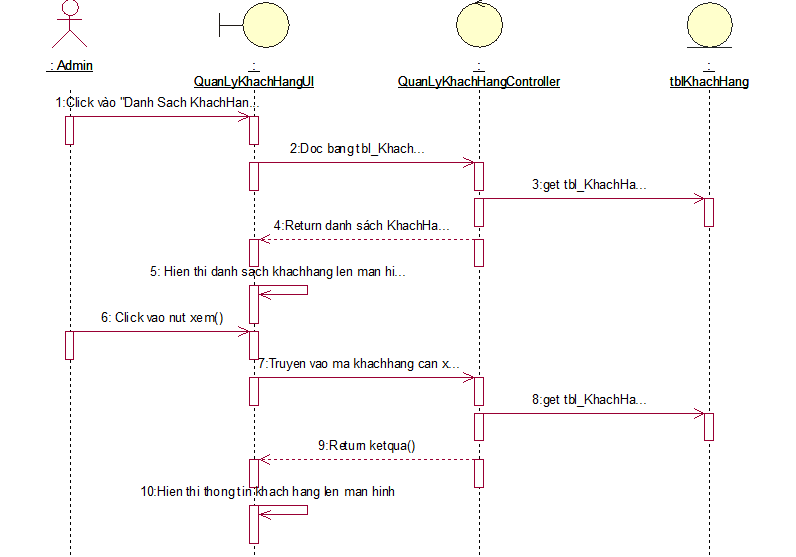


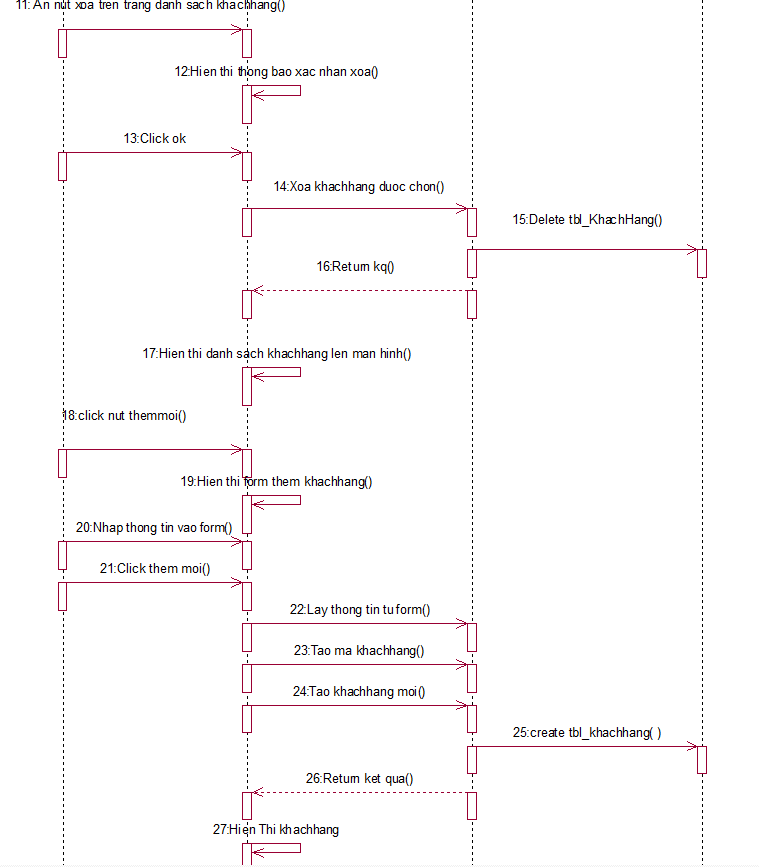




Hình 3.32. Biểu đồ trình tự chức năng quản lý tin tức

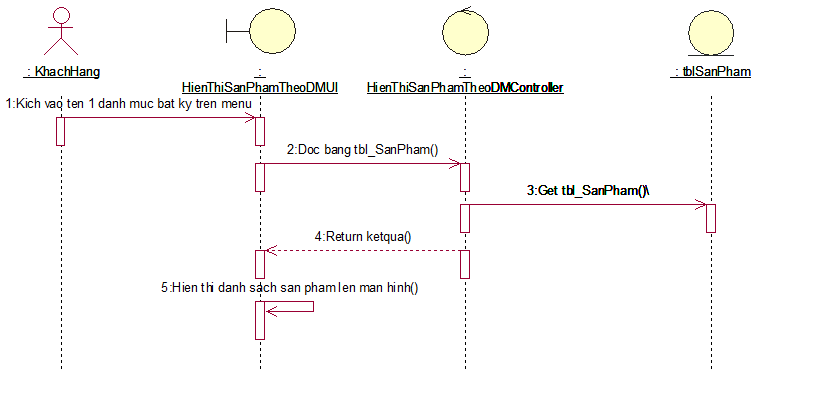
### **3.4.12 Biểu đồ trình tự chức năng quản lý khách hàng**





Hình 3.33. Biểu đồ trình tự chức năng quản lý khách hàng

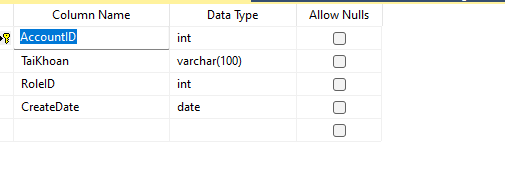
### **3.4.13 Biểu đồ trình tự chức năng hiển thị sản phẩm theo danh mục**

****

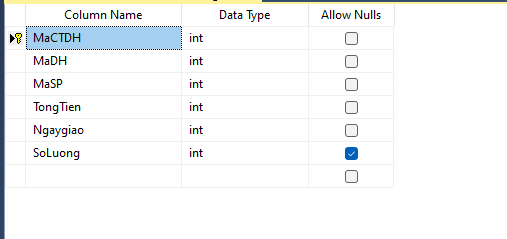
Hình 3.34. Biểu đồ trình tự chức năng hiển thị sản phẩm theo danh mục

## 3.5 Mô hình cơ sở dữ liệu

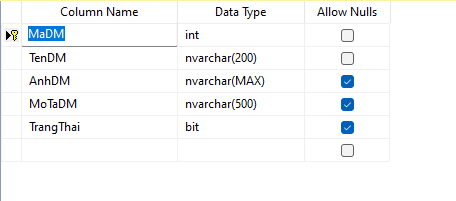
### **3.5.1 Thiết kế bảng**

****

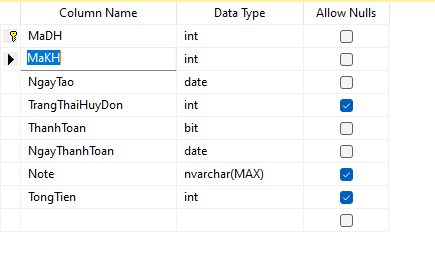
Hình 3.35. Bảng account

**

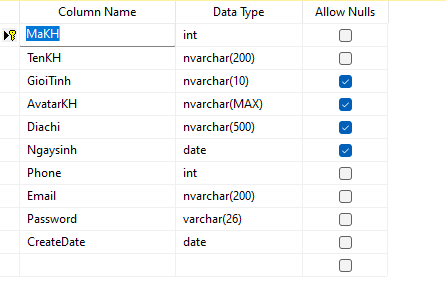
Hình 3.36. Bảng chi tiết đơn hàng

**

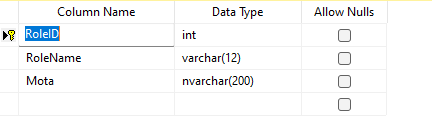
Hình 3.37. Bảng danh mục

**

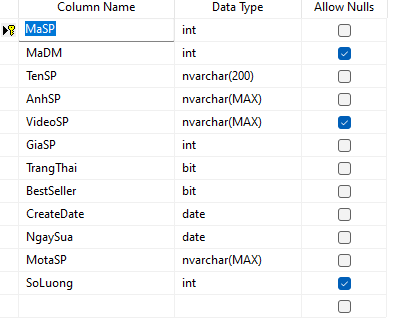
Hình 3.38. Bảng đơn hàng

**

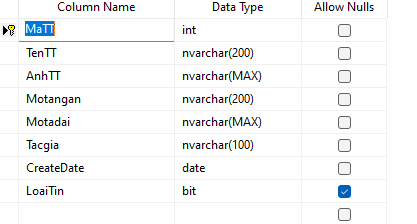
Hình 3.39. Bảng khách hàng

**

Hình 3.40. Bảng quyền truy cập

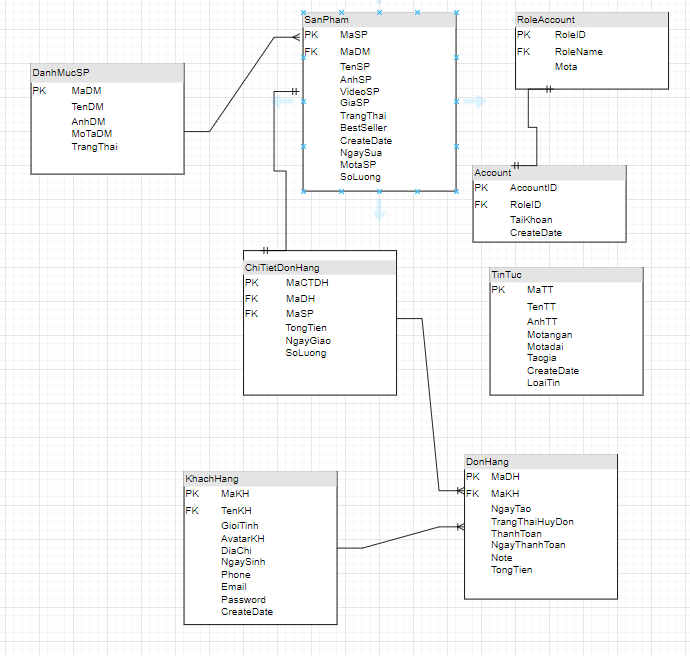
**

Hình 3.41. Bảng sản phẩm

**

Hình 3.42. Bảng tin tức

### **3.5.2 Biểu đồ quan hệ thực thể**

**

Hình 3.44. Biểu đồ quan hệ thực thể

# **CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG**

* 1. **Công cụ sử dụng**

### **4.1.1 Microsoft Visual Studio**

* **Microsoft Visual Studio** là một trong những công cụ phát triển phần mềm phổ biến và mạnh mẽ nhất hiện nay. Được Microsoft giới thiệu lần đầu tiên vào năm 1997, đến nay Visual Studio đã trở thành lựa chọn hàng đầu của các nhà phát triển phần mềm chuyên nghiệp.
* **Microsoft Visual Studio** là môi trường phát triển tích hợp (IDE) được thiết kế dành cho giới lập trình viên và các nhà phát triển ứng dụng. Đây là công cụ hỗ trợ phát triển phần mềm mạnh mẽ của Microsoft, cho phép người dùng viết, dịch mã và gỡ lỗi các ứng dụng dựa trên nhiều ngôn ngữ lập trình khác nhau như C++, C#, Visual Basic, Python, JavaScript... Visual Studio bao gồm một trình biên tập mã nguồn, các công cụ gỡ lỗi và xây dựng ứng dụng đa nền tảng. Nó giúp tăng năng suất và hiệu quả công việc cho các lập trình viên.

**Các tính năng nổi bật của Microsoft Visual Studio:**

* Hỗ trợ đa ngôn ngữ lập trình
* Hỗ trợ gỡ lỗi mạnh mẽ
* Tích hợp Git
* Tự động hoàn thành mã (IntelliSense)
* Hỗ trợ đa nền tảng

Với Visual Studio, lập trình viên có thể dễ dàng xây dựng các ứng dụng chạy trên nhiều nền tảng, tiết kiệm thời gian phát triển và chi phí. Đây là một tính năng rất hữu ích và mạnh mẽ của Visual Studio.

### **4.1.2 Rational Rose**

**Rational Rose** là một công cụ CASE (Computer-Aided Software Engineering) phổ biến được sử dụng trong quá trình phát triển phần mềm. Được phát triển bởi Rational Software, sau này được IBM mua lại, Rational Rose cung cấp các tính năng và công cụ cho việc phân tích, thiết kế, và triển khai phần mềm.

**Các tính năng nổi bật của Rational Rose:**

**Phân tích và Thiết kế:**

* Rational Rose cho phép nhóm phát triển tạo ra các mô hình phân tích và thiết kế phần mềm bằng cách sử dụng các định dạng mô hình phổ biến như UML (Unified Modeling Language).
* Các mô hình có thể bao gồm các biểu đồ lớp, biểu đồ use case, biểu đồ tuần tự, biểu đồ hoạt động, và nhiều loại khác.

**Tích hợp và Đồng bộ hóa:**

* Rational Rose có thể tích hợp với nhiều công cụ và môi trường phát triển phần mềm khác nhau, như Rational ClearCase và Rational RequisitePro.
* Tích hợp này cho phép nhóm phát triển làm việc cùng nhau và đồng bộ hóa các phần của dự án một cách hiệu quả hơn.

**Hỗ trợ đa nền tảng:**

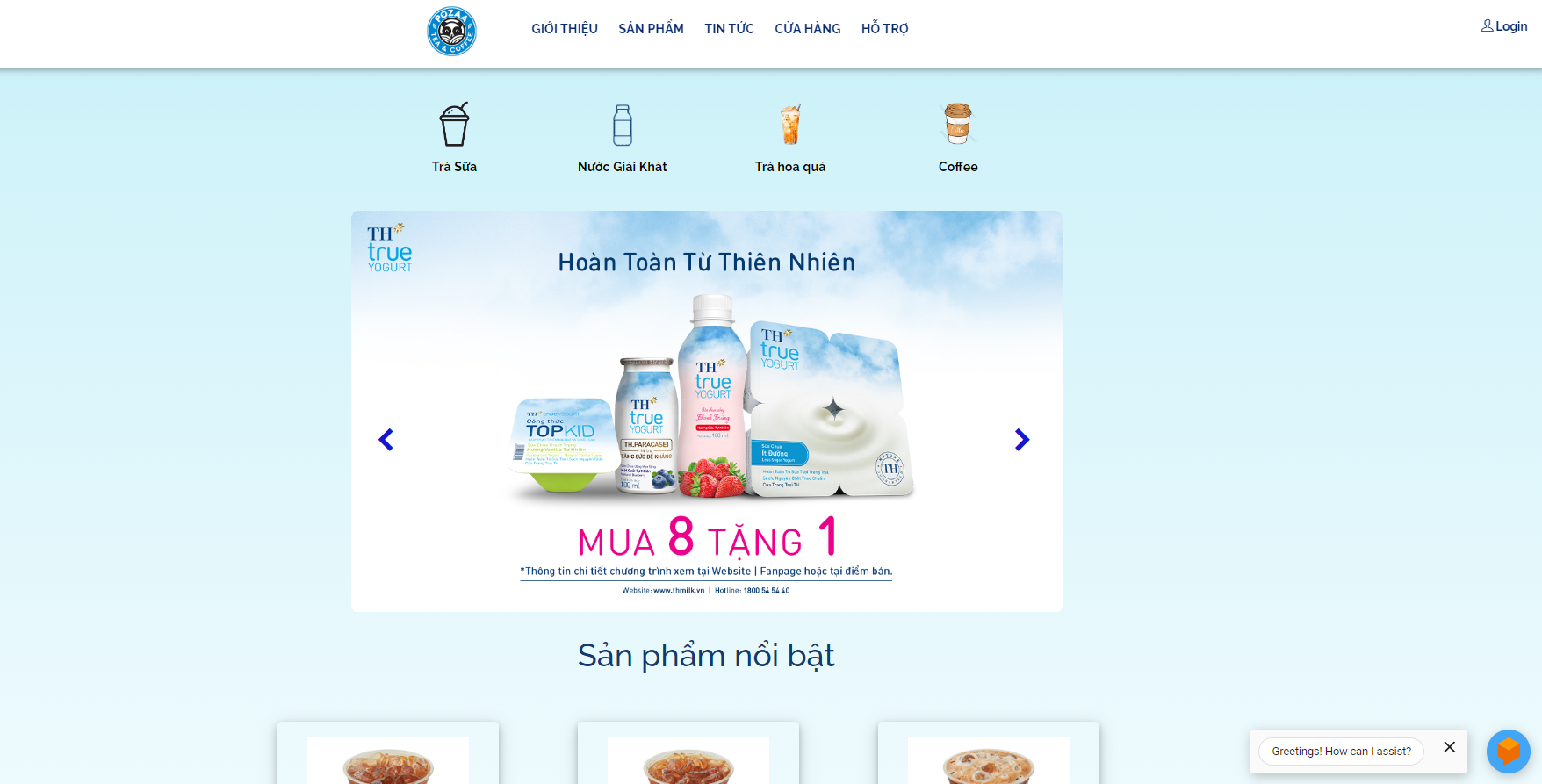
* Rational Rose hỗ trợ phát triển phần mềm trên nhiều nền tảng và ngôn ngữ lập trình khác nhau. Điều này cho phép nhóm phát triển làm việc trên các dự án đa dạng và đa nền tảng.
* Hỗ trợ cho các ngôn ngữ lập trình phổ biến như C++, Java, và C#.

Rational Rose là một trong những công cụ quan trọng trong quá trình phát triển phần mềm và đã được sử dụng rộng rãi trong ngành công nghiệp. Tính linh hoạt, tính toàn diện, và khả năng tích hợp là những đặc điểm mạnh mẽ của nó.

## Giao diện các chức năng

**Giao diện trang chủ trang web**

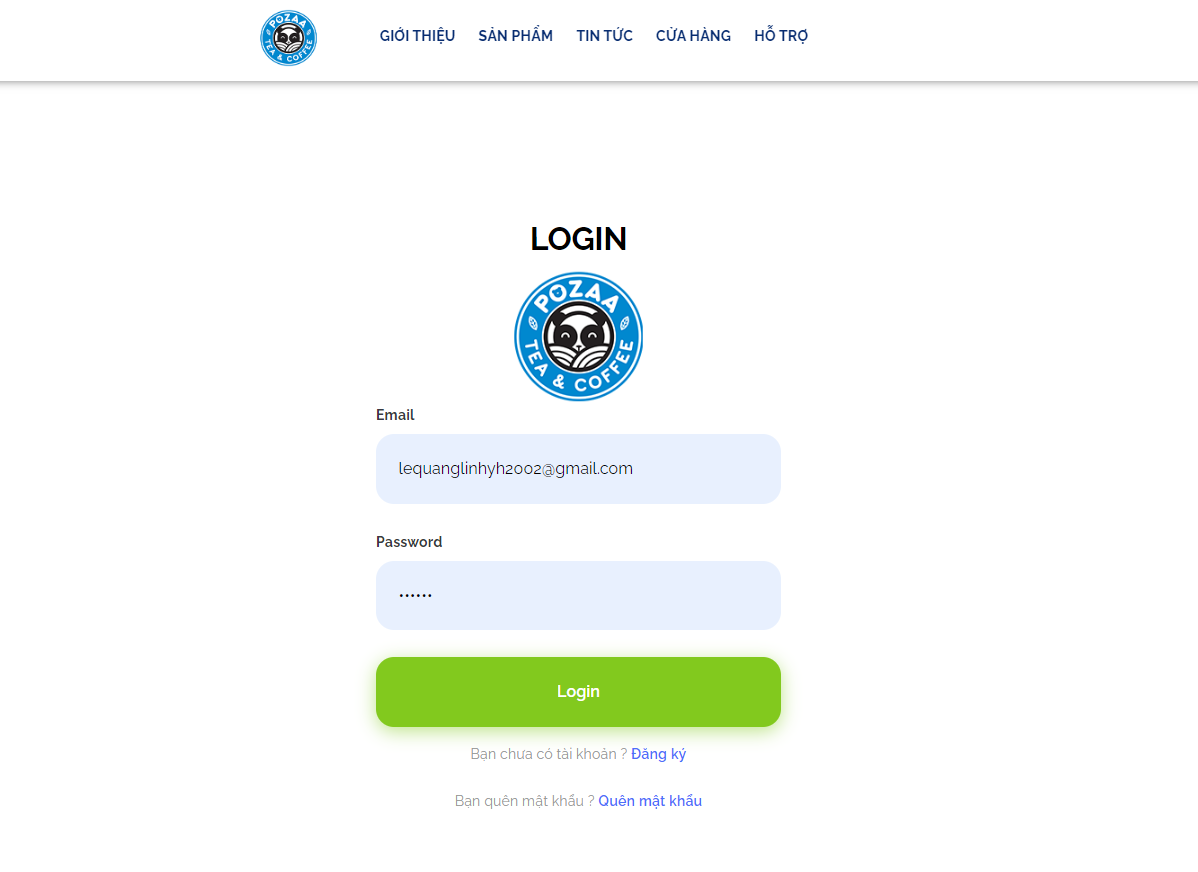
* Trang chủ trang web cho khách hàng, bao gồm các chức năng cơ bản của trang web. Người dùng có thể xem thông tin về các sản phẩm, các danh mục sản phẩm, và tin tức của cửa hàng.
* Người thực hiện: Khách hàng.
* Mẫu thiết kế:



Hình 4.1. Giao diện trang chủ

**Giao diện trang đăng nhập**

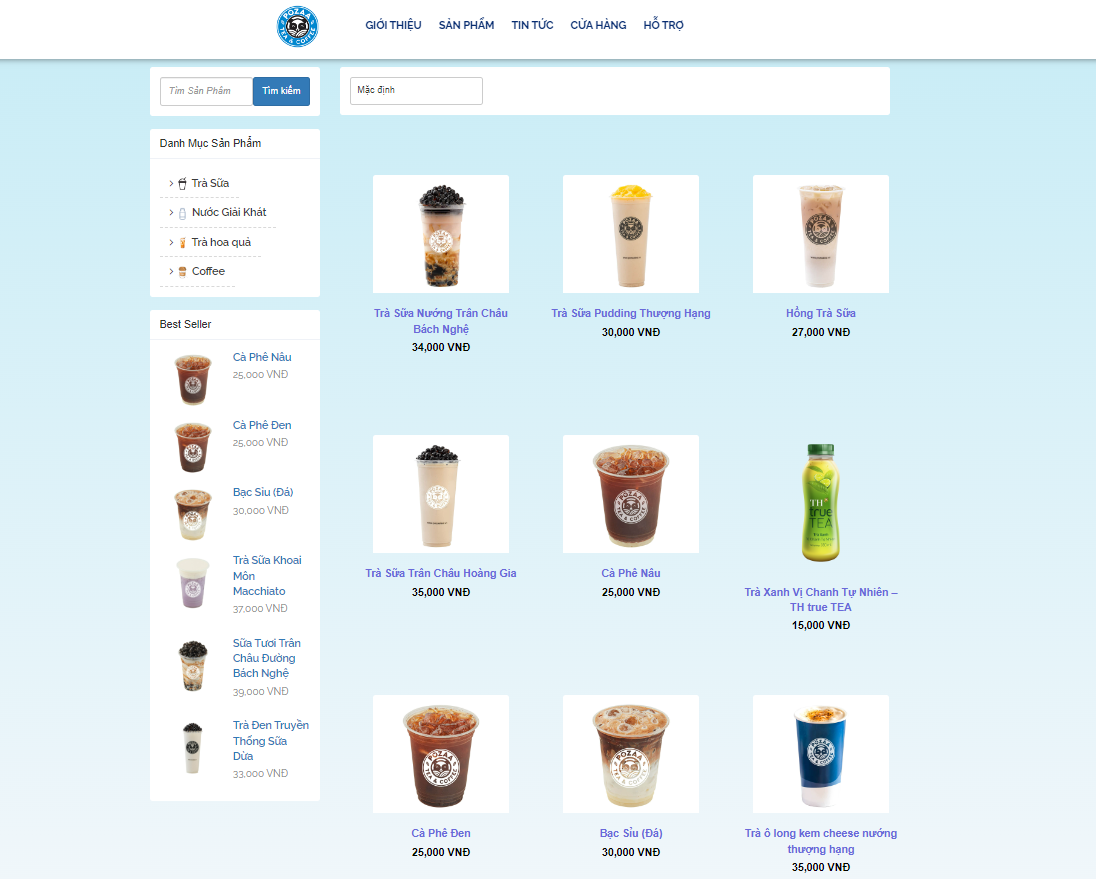
* Trang đăng nhập cho phép người dùng thực hiện các chức năng cơ bản như đăng nhập, đăng ký tài khoản, hoặc lấy lại mật khẩu thông qua email đã đăng ký.
* Người thực hiện: Khách hàng, quản lý cửa hàng.
* Mẫu thiết kế:



Hình 4.2. Giao diện trang đăng nhập

**Giao diện trang danh sách sản phẩm**

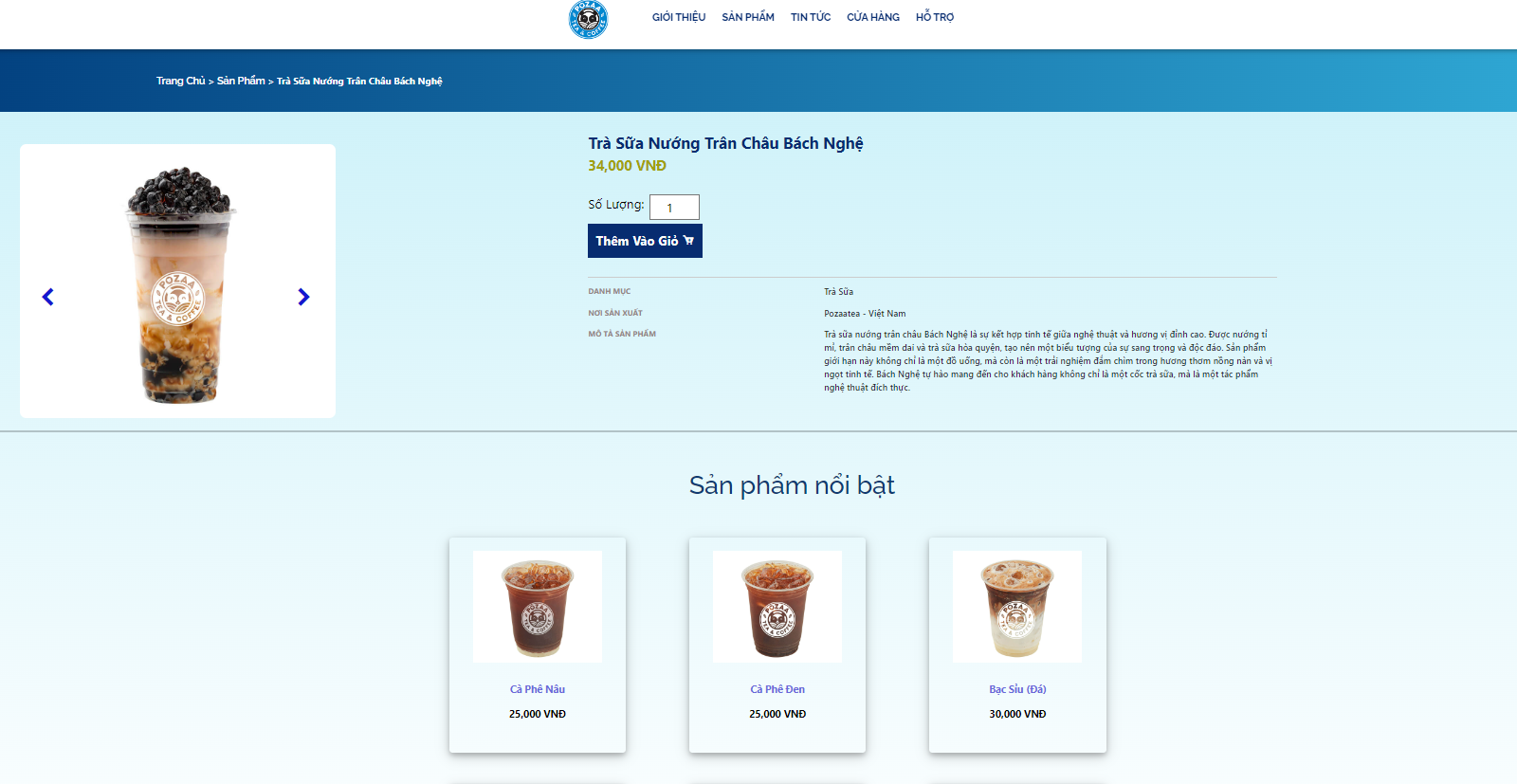
* Trang danh sách sản phẩm cho phép khách hàng có thể xem được các sản phẩm có trong cửa hàng, lấy sản phẩm theo danh mục, tìm kiếm sản phẩm theo tên và sắp xếp sản phẩm theo giá bán.
* Người thực hiện: Khách hàng.
* Mẫu thiết kế:

**

Hình 4.3. Giao diện trang danh sách sản phẩm

**Giao diện trang chi tiết sản phẩm**

* Trang chi tiết sản phẩm cho phép khách hàng xem thông tin của sản phẩm như: tên sản phẩm, giá bán sản phẩm, danh mục sản phẩm, mô tả sản phẩm và thêm sản phẩm vào giỏ hàng.
* Người thực hiện: Khách hàng.
* Mẫu thiết kế:

**

*Hình 4.4. Giao diện chi tiết sản phẩm*

**Giao diện trang giỏ hàng**

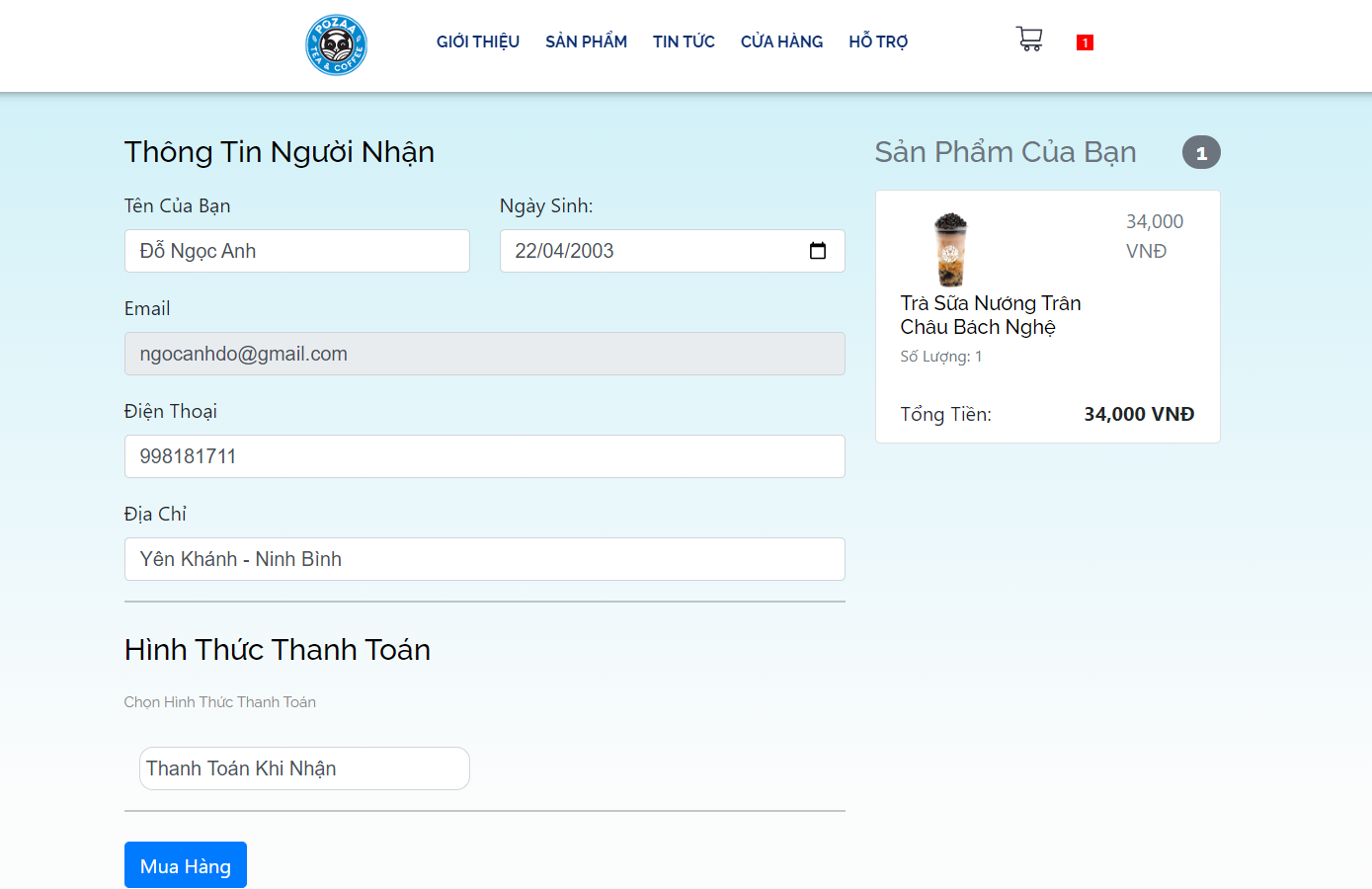
* Trang giỏ hàng cho phép người dùng xem thông tin về sản phẩm, cập nhập số lượng sản phẩm muốn mua, xóa sản phẩm và đặt hàng
* Người thực hiện: Khách hàng
* Mẫu thiết kế:

**

Hình 4.5. Giao diện giỏ hàng

**Giao diện trang đặt hàng**

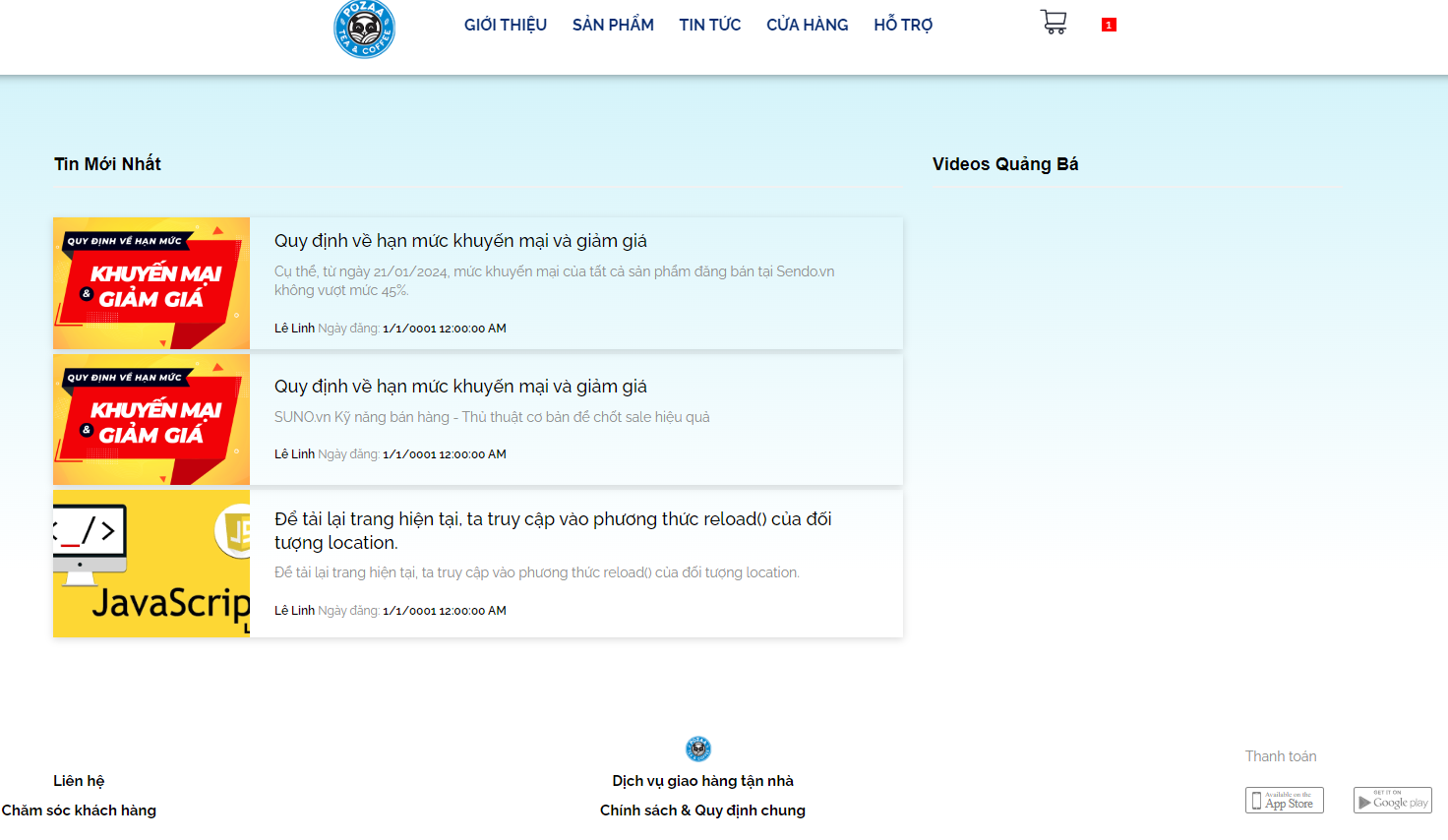
* Trang đặt hàng cho phép khách hàng xem thông tin sản phẩm mà khách hàng muốn mua, khách hàng có thể cập nhập lại thông tin người nhận cho phù hợp với yêu cầu của mình và đặt mua sản phẩm trong cửa hàng.
* Người thực hiện: Khách hàng.
* Mẫu thiết kế:

**

Hình 4.6. Giao diện đặt hàng

**Giao diện trang tin tức**

* Trang tin tức cho phép khách hàng có thể xem được các tin tức có trong cửa hàng. Khách hàng có thể kích vào tiêu đề tin tức để biết thêm nhiều thông tin hơn.
* Người thực hiện: Khách hàng.
* Mẫu thiết kế:

**

Hình 4.7. Giao diện trang tin tức

**Giao diện trang chủ admin**

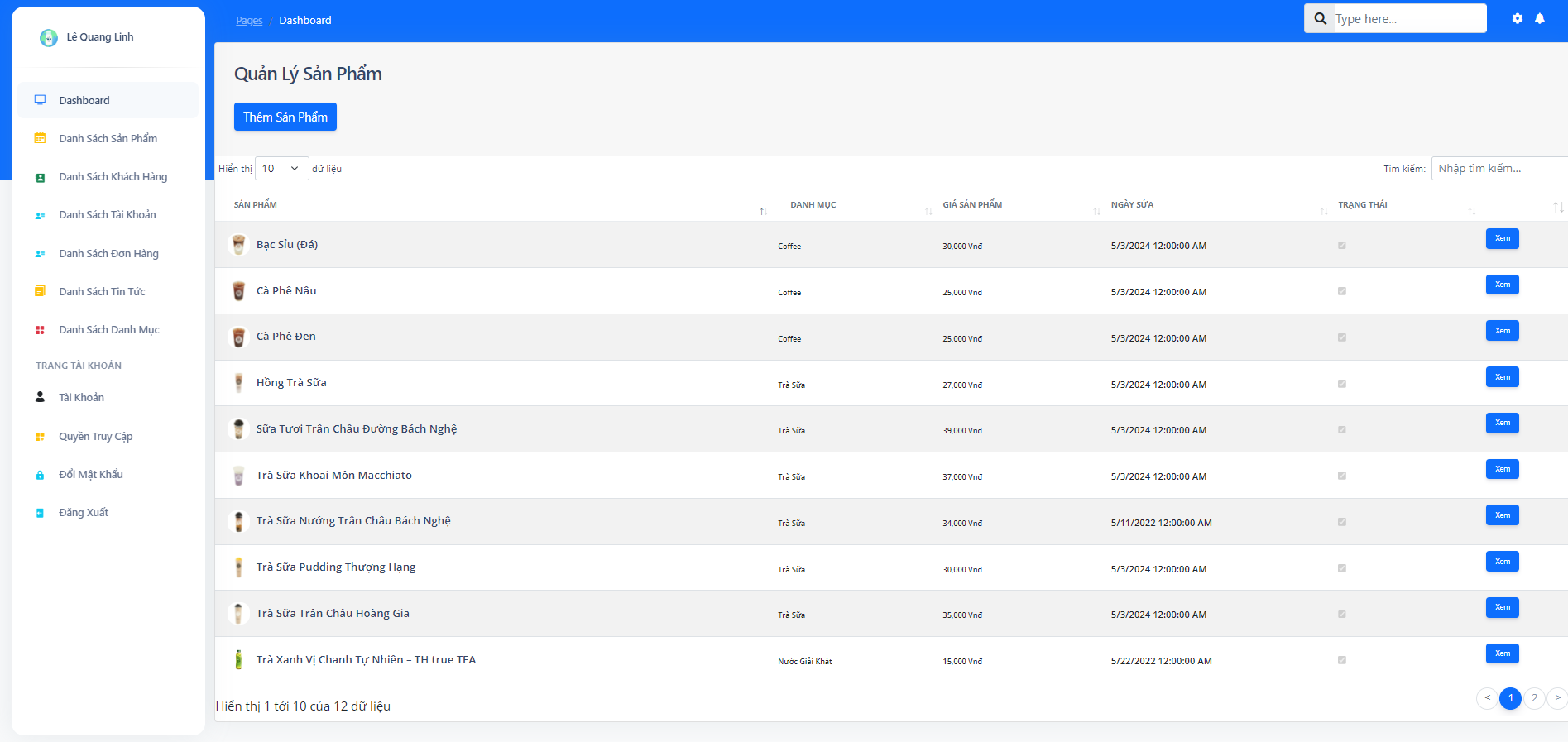
* Trang chủ admin cho phép người dùng có thể xem thống kê doanh thu của cửa hàng, thống kê được khách hàng mới, các chức năng liên quan đến quản lý.
* Người thực hiện: Quản lý cửa hàng
* Mẫu thiết kế:

**

Hình 4.8. Giao diện trang chủ admin

**Giao diện trang quản lý sản phẩm**

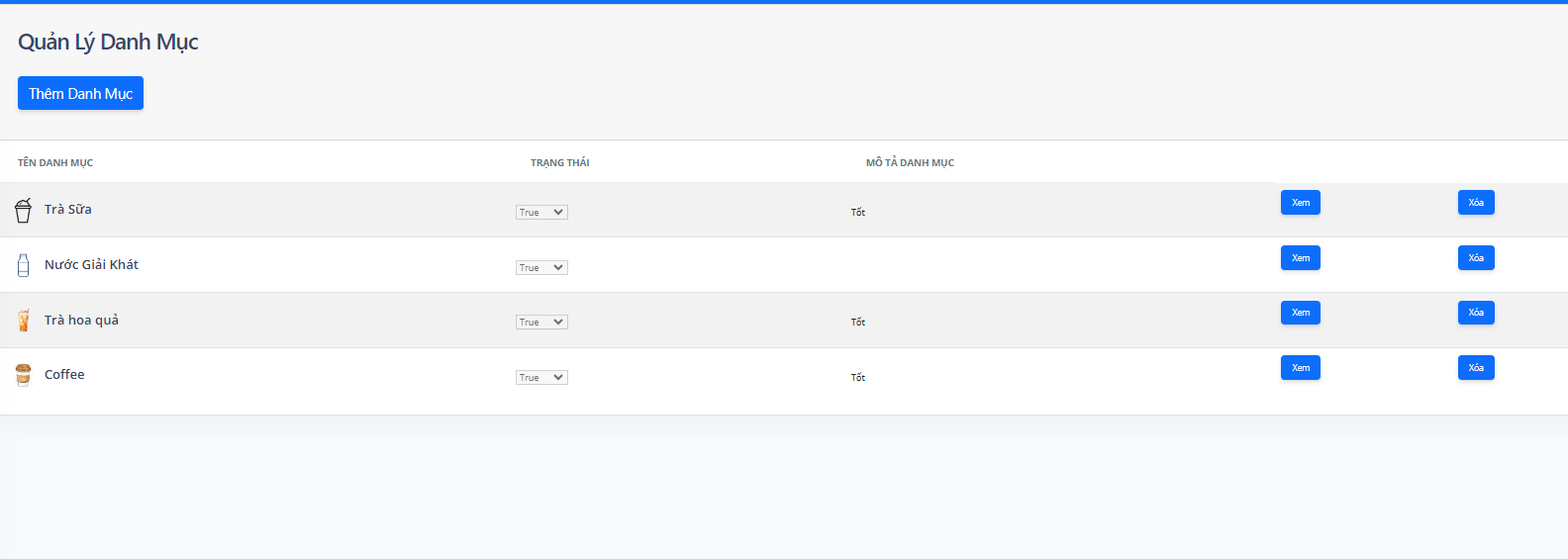
* Trang quản lý sản phẩm cho phép người dùng xem được thông tin của các sản phẩm có trong cửa hàng. Người dùng có thể thực hiện các chức năng thêm, sửa, xóa các sản phẩm có trong cửa hàng.
* Người thực hiện: Quản lý cửa hàng.
* Mẫu thiết kế:

**

Hình 4.9. Giao diện quản lý sản phẩm

**Giao diện trang quản lý danh mục**

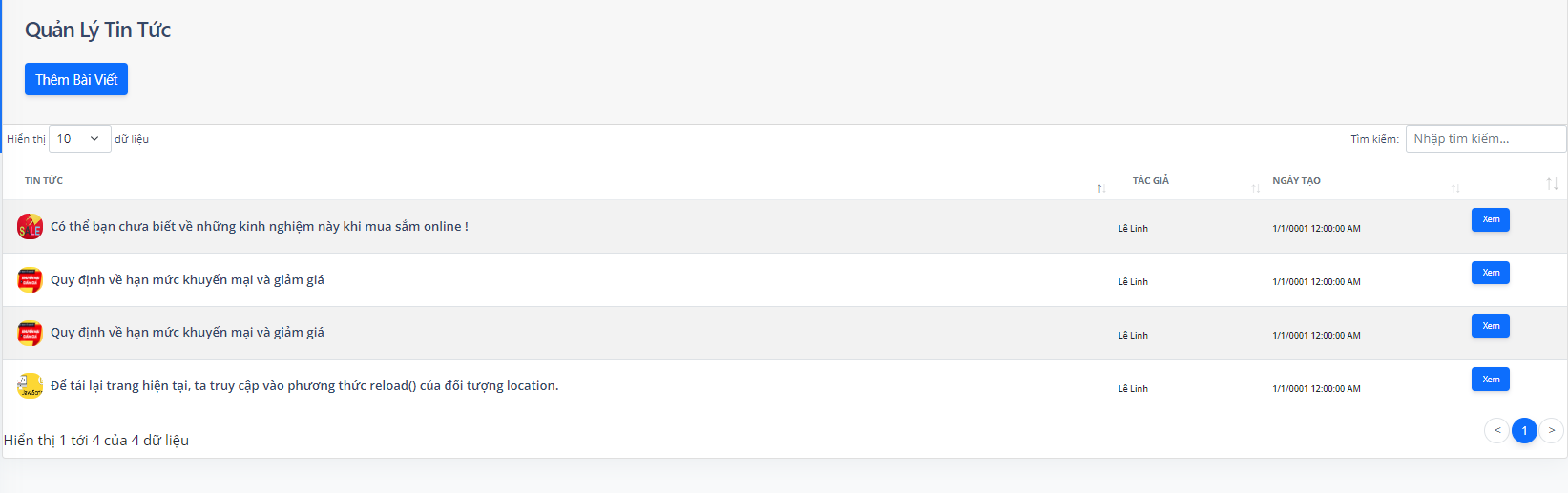
* Trang quản lý sản phẩm cho phép người dùng xem được thông tin của các danh mục có trong cửa hàng. Người dùng có thể thực hiện các chức năng thêm, sửa, xóa các danh mục có trong cửa hàng.
* Người thực hiện: Quản lý cửa hàng.
* Mẫu thiết kế:

**

Hình 4.10. Giao diện quản lý danh mục

**Giao diện trang quản lý tin tức**

* Trang quản lý sản phẩm cho phép người dùng xem được thông tin của các tin tức có trong cửa hàng. Người dùng có thể thực hiện các chức năng thêm, sửa, xóa các tin tức có trong cửa hàng.
* Người thực hiện: Quản lý cửa hàng.
* Mẫu thiết kế:

**

Hình 4.11. Giao diện quản lý tin tức

**KẾT LUẬN**

Đề tài “**Xây dựng website bán đồ uống bằng ASP.NET**” xuất phát từ nhu cầu học tập, trau dồi kiến thức của em trong những năm tháng học tập trên giảng đường. Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và vận dụng được những kiến thức công nghệ được học từ trường, từ nơi làm việc và đã hoàn thành được những phần sau đây.

**Kết quả đạt được**

**Đối với cá nhân:**

* Củng cố được kiến thức nền vững chắc về CSDL, phân tích thiết kế hệ thống, các ngôn ngữ lập trình web.
* Tiếp thu được nhiều công nghệ mới, tối ưu hóa thời gian thực hiện dự án.
* Áp dụng được các kiến thức, kỹ năng, công nghệ mới vào việc giải quyết bài toán thực tiễn.

**Đối với sản phẩm:**

* Cho phép người dùng có thể tìm và đặt hàng một cách nhanh chóng. Người quản lý cũng dễ dàng hơn trong việc quản lý cửa hàng.
* Xây dựng một hệ thống mua bán, quản lý cho cửa hàng PozaaTea, người quản lý nhân sự có thể quản lý thông tin hồ sơ khách hàng, biết được doanh thu, quản lý đơn hàng hiệu quả nhanh chóng.

**Yêu cầu chưa đạt được**

Do chưa có nhiều kinh nghiệm trong dự án thực tế cũng như thời gian hạn chế, ứng dụng của em vẫn còn thiếu một số chức năng. Vì vậy, em mong rằng các quý thầy cô có thể đóng góp ý kiến để giúp em hoàn thiện ứng dụng tốt hơn trong tương lai.

**Các chức năng còn thiếu:**

* Chưa có chức năng như bình luận, đánh giá sản phẩm để khảo sát được mức độ hài lòng của khách hàng.
* Chưa tích hợp chức năng thanh toán online để quá trình đặt hàng của khách hàng có thêm nhiều lựa chọn hơn.
* Chưa tích hợp chức năng đăng nhập bằng các tài khoản facebook, gmail,…

**Hướng nghiên cứu trong tương lai**

* Tiếp tục hoàn thiện các tính năng mở rộng phần mềm nhằm cải tiến và nâng cấp chương trình.
* Cải thiện tính bảo mật.
* Tích hợp Single Sign On như facebook, gmail…
* Tối ưu cách viết code và cơ sở dữ liệu.

\

# **TÀI LIỆU THAM KHẢO**

**Tiếng việt**

**[1].** Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, Nhập môn công nghệ phần mềm, NXB Đại học Công nghiệp Hà Nội.

**[2].** Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, Giáo trình phân tích thiết kế hệ thống, NXB Giáo dục VN.

**[3].** Dương Quang Thiện, .NET toàn tập, NXB tổng hợp TP.HCM

**[4].** Giáo trình thiết kế web, Trường đại học Công Nghiệp Hà Nội.

**[5].** Đỗ Ngọc Sơn, Nguyễn Phương Nga, Phan Văn Viên, Giáo trình Hệ quản trị cơ sở dữ liệu, NXB Đại học Công Nghiệp Hà Nội

**[6].** <https://xuanthulab.net/>