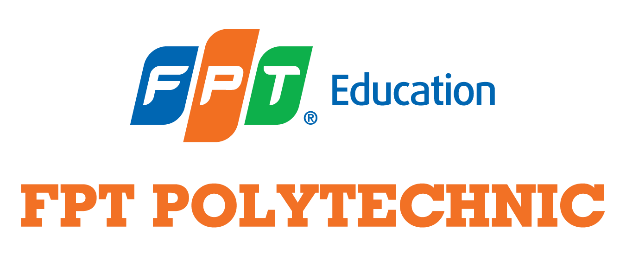
*TP. Hồ Chí Minh, 1 tháng 3 năm 2025*

**TRƯỜNG CAO ĐẲNG FPT POLYTECHNIC**

**Bộ Môn: Lập trình C#6 – NET1061**

**Chuyên Ngành: Phát Triển phần Mềm**



**ASSIGNMENT**

**DỰ ÁN KINH DOANH THỨC ĂN NHANH**

**MÃ MÔN HỌC: NET1061\_SD19310**

**Giảng viên hướng dẫn :** Thầy Trần Thanh Phương

**Nhóm thực hiện :** 1**.** Cao Quốc Thịnh - PS37749 (Leader)

2. Trương Thanh Bạch - PS41294

3. Nguyễn Đình Vương - PS41292

*TP. Hồ Chí Minh, ngày 28 tháng 3, 2025*

# **LỜI CẢM ƠN**

Trong suốt quá trình thực hiện dự án này, chúng tôi đã nhận được rất nhiều sự hỗ trợ và đồng hành từ các cá nhân và tổ chức, và chúng tôi xin gửi lời cảm ơn sâu sắc đến tất cả những người đã góp phần quan trọng vào thành công của công trình nghiên cứu này.

Trước hết, chúng tôi xin cảm ơn Thầy Trần Thanh Phương, người đã luôn kèm cặp và hướng dẫn chúng tôi từng bước, chia sẻ những kiến thức quý giá và kinh nghiệm thực tế, giúp nhóm vượt qua các thử thách trong suốt quá trình làm việc. Lời cảm ơn cũng xin được gửi đến các thầy cô và bạn bè trong trường đã tạo điều kiện và động viên chúng tôi.

Chúng tôi cũng xin cảm ơn các thành viên trong nhóm vì sự hợp tác, nỗ lực không ngừng nghỉ và sự sáng tạo, đóng góp hết mình để đưa dự án này đến thành công. Cuối cùng, xin gửi lời cảm ơn đến gia đình và những người thân yêu đã luôn bên cạnh, động viên và tạo động lực cho chúng tôi trong suốt quá trình thực hiện dự án.

Nhóm thực hiện!

….

# **MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc163308101)

[THÔNG TIN VỀ ĐỀ TÀI 2](#_Toc163308102)

[1. Lý do thực hiện đề tài 2](#_Toc163308103)

[2. Mục tiêu đề tài 2](#_Toc163308104)

[3. Nội dung thực hiện 2](#_Toc163308105)

[4. Phương pháp thực hiện 2](#_Toc163308106)

[5. Dự kiến kết quả đạt được 2](#_Toc163308107)

[CHƯƠNG 1: TỔNG QUAN 3](#_Toc163308108)

[1.1. Tổng quan về vấn đề được nghiên cứu 3](#_Toc163308109)

[1.1.1. Khảo sát thực trạng 3](#_Toc163308110)

[1.1.2. Đánh giá 3](#_Toc163308111)

[1.2. Phân tích đồ án 3](#_Toc163308112)

[1.2.1. Đối tượng và phạm vi của ứng dụng 3](#_Toc163308113)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 5](#_Toc163308114)

[2.1. Các khái niệm và cơ chế hoạt động 5](#_Toc163308115)

[2.1.1. Khái niệm lập trình mô hình MVC 5](#_Toc163308116)

[2.1.2. Lợi ích của lập trình theo mô hình MVC 6](#_Toc163308117)

[2.2. Các công nghệ sử dụng 6](#_Toc163308118)

[2.2.1. Visual Studio 2022 6](#_Toc163308119)

[2.2.2. SMSS (Microsoft SQL Server 2022) 7](#_Toc163308120)

[CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM 9](#_Toc163308121)

[3.1 Tài liệu Yêu cầu Phần mềm 9](#_Toc163308122)

[3.2. Sơ đồ use case 9](#_Toc163308123)

[3.3. Các thuộc tính của các thực thể 11](#_Toc163308124)

[3.4. Mô hình kết nối dữ liệu 14](#_Toc163308125)

[3.5. Thiết kế, mô tả giao diện 16](#_Toc163308126)

[CHƯƠNG 4: KẾT LUẬN 28](#_Toc163308127)

[4.1. Kết quả thực hiện 28](#_Toc163308128)

[4.2. Đánh giá 28](#_Toc163308129)

[4.2.1. Ưu điểm 28](#_Toc163308130)

[4.2.2. Khuyết điểm 29](#_Toc163308131)

[4.2.3. Thuận lợi 29](#_Toc163308132)

[4.2.4. Khó khăn 29](#_Toc163308133)

[4.3. Hướng phát triển 29](#_Toc163308134)

[TÀI LIỆU THAM KHẢO 30](#_Toc163308135)

**THÔNG TIN ĐỀ TÀI**

1. **Lý do chọn đề tài**

Trong xu hướng tiêu dùng hiện đại, đồ ăn nhanh ngày càng trở nên phổ biến nhờ vào tính tiện lợi và nhanh chóng. Việc xây dựng một website bán đồ ăn nhanh trực tuyến không chỉ giúp người tiêu dùng dễ dàng đặt món mà còn giúp các cửa hàng nâng cao hiệu quả kinh doanh. Do đó, tôi chọn đề tài "Xây dựng website bán đồ ăn nhanh" nhằm phục vụ nhu cầu và tính thuận lời khi mua đồ ăn nhanh mà không cần ra ngoài.

1. **Mục tiêu đề tài**

Xây dựng một website bán đồ ăn nhanh có giao diện đẹp mắt và dễ dàng sử dụng cho người dùng.

1. **Nội dung thực thiện**

Gồm 5 phần: Phân tích yêu cầu, thiết kế cơ sở dữ liệu, thiết kế giao diện, kiểm thử, vận hành.

1. Phương pháp thực hiện

* Phần 1: Thiết kế cơ sở dữ liệu
* Phần 2: Đáp ứng được các chức năng ( Đăng nhập, thêm xóa sửa người dùng, thêm xóa sửa sản phẩm, thêm xóa sửa sản phẩm trong giỏ hàng, thanh toán API VNPAY, quản lý đơn hàng, thống kê)
* Phần 3: Tạo Template và layout đồng nhất cho ứng dụng.
* Phần 4: Thực hiện yêu cầu về truy vấn.
* Phần 5: Kiểm tra tổng quát lần cuối sau đó hoàn thiện website.

**CHƯƠNG 1: TỔNG QUAN**

* 1. **Tổng quan về vấn đề nghiên cứu thị trường**
     1. **Khảo sát thị trường**

Các trang web bán đồ ăn nhanh hiện nay đang trở thành xu hướng phổ biến trong ngành thực phẩm, thu hút sự quan tâm của nhiều doanh nghiệp và người tiêu dùng.

* + 1. **Đánh giá**

**1.1.2.1. Ưu điểm**

* **Tiện lợi và nhanh chóng:** Người dùng có thể đặt món ăn trực tuyến mà không cần ra ngoài, tiết kiệm thời gian.
* **Đa dạng sản phẩm:** Người dùng có thể dễ dàng tham khảo nhiều món ăn khác nhau, từ món ăn vặt đến các combo đặc biệt, ngay trên trang web mà không bị giới hạn bởi không gian vật lý.
* **Dịch vụ giao hàng nhanh chóng:** Với các dịch vụ giao hàng nhanh, khách hàng có thể nhận món ăn ngay tại nhà hoặc văn phòng, tạo sự thuận tiện tối đa.
* **Khả năng tiếp cận cao:** Trang web có thể phục vụ khách hàng ở mọi nơi, mở rộng phạm vi tiếp cận và tiếp cận khách hàng mới.

**1.1.2.2 Nhược điểm**

* **Không thể trải nghiệm trực tiếp món ăn:** Khách hàng không thể thử món ăn trước khi mua, điều này có thể khiến một số người còn do dự.
* **Phụ thuộc vào dịch vụ giao hàng:** Chất lượng món ăn có thể bị ảnh hưởng trong quá trình vận chuyển, đôi khi không được như khi ăn tại quán.
* **Cần thiết bị điện tử và kết nối internet:** Người mua cần có thiết bị điện tử (smartphone, máy tính) và kết nối internet để sử dụng dịch vụ.
* **Giới hạn trong việc lựa chọn địa chỉ giao hàng:** Nếu khách hàng ở khu vực xa hoặc không hỗ trợ giao hàng, việc sử dụng dịch vụ có thể gặp khó khăn.
  1. **Phân tích đồ ăn.**

**1.2.1. Đối tượng & phạm vi của ứng dụng**

**1.2.1.1. Phạm vi**

* Tổng quan dự án:
* **Đơn vị sử dụng:** Website bán đồ ăn nhanh Anh Bar.
* **Tên dự án:** Website bán đồ ăn nhanh.
* Mục tiêu:
* Tiếp cận nhiều khách hàng tiềm năng hơn thông qua nền tảng trực tuyến.
* Cung cấp cho khách hàng một trải nghiệm mua sắm thuận tiện, nhanh chóng và tiết kiệm thời gian.
* Mô tả:
* Hiển thị đầy đủ các sản phẩm đồ ăn nhanh có thể đặt mua trực tuyến.
* Cung cấp thông tin về các ưu đãi, khuyến mãi, và sản phẩm mới.
* **Các chức năng của khách hàng:** Xem lịch sử mua hàng, tạo và quản lý tài khoản cá nhân, theo dõi trạng thái đơn hàng, thanh toán đơn hàng, xem chi tiết đơn hàng, thêm sản phẩm vào giỏ hàng.
* **Các chức năng của quản trị viên:** Thêm, sửa, xóa thông tin sản phẩm, quản lý đơn hàng, quản lý thông tin người dùng (User, Admin), thống kê doanh thu.
* Các bước thực hiện để hoàn thành dự án:
* Nghiên cứu và tìm hiểu kiến trúc **Web API** & **Lazor** (Sever, Assembly).
* Thiết kế cơ sở dữ liệu & **UseCase** phù hợp.
* Phân tích hành vi người dùng để tối ưu giao diện.
* Thiết kế giao diện người dùng đơn giản và thân thiện.
* Lập trình các chức năng chính và triển khai hiệu quả.
* Kiểm thử, sửa lỗi và tối ưu hiệu suất hệ thống.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **2.1** **Các khái niệm và cơ chế hoạt động**

**2.1.1 Khái niệm về lập trình Web API & Lazor**

MVC (Model-View-Controller) là một mô hình thiết kế phần mềm phổ biến được sử dụng trong phát triển ứng dụng web. Mô hình này giúp chia tách mã nguồn thành ba phần riêng biệt, mỗi phần có vai trò và chức năng riêng biệt, giúp dễ dàng bảo trì và phát triển ứng dụng.

1. **Model (M):**

Chịu trách nhiệm quản lý dữ liệu và logic nghiệp vụ của ứng dụng. Model là nơi tương tác với cơ sở dữ liệu hoặc các dịch vụ khác để lưu trữ, truy xuất và xử lý dữ liệu. Trong Web API, Model có thể là các đối tượng dữ liệu (DTOs) hoặc lớp tương tác với cơ sở dữ liệu.

1. **View (V):**

View là phần giao diện của ứng dụng, nơi người dùng tương tác và hiển thị dữ liệu. Tuy nhiên, trong Web API, View không thực sự tồn tại vì API chủ yếu trả về dữ liệu (thường là JSON hoặc XML) chứ không phải giao diện. View trong trường hợp của Web API có thể hiểu là phần giao diện mà người tiêu dùng API (ví dụ: ứng dụng di động hoặc web) nhận được qua các phản hồi từ API.

1. **Controller (C):**

Controller nhận các yêu cầu từ người dùng và xử lý các yêu cầu đó. Trong Web API, Controller là nơi tiếp nhận các HTTP requests (như GET, POST, PUT, DELETE), xử lý chúng, tương tác với Model để lấy dữ liệu và trả lại kết quả cho người dùng. Controller sẽ đảm bảo phân phối dữ liệu tới client dưới định dạng dễ sử dụng (thường là JSON).

**2.1.2. Lợi ích của lập trình theo mô hình MVC trong Web API**

* **Tối ưu hóa và dễ bảo trì:**

MVC giúp phân chia rõ ràng giữa các thành phần của ứng dụng, từ đó dễ dàng bảo trì và phát triển ứng dụng mà không làm ảnh hưởng đến các thành phần khác. Khi sử dụng trong Web API, việc phân tách Model, View và Controller giúp tối ưu hóa quản lý dữ liệu và logic nghiệp vụ.

* **Kiểm tra dễ dàng:**

Nhờ việc phân tách mã nguồn, việc kiểm thử trong mô hình MVC trở nên dễ dàng hơn. Bạn có thể kiểm thử các phần như Model, Controller độc lập mà không lo bị ảnh hưởng tới các thành phần khác.

* **Quản lý dữ liệu tốt hơn:**

Mô hình MVC giúp tách biệt rõ ràng giữa việc lưu trữ và xử lý dữ liệu (Model), logic xử lý yêu cầu (Controller) và giao diện (View). Điều này giúp giảm thiểu sự phức tạp trong ứng dụng và dễ dàng bảo trì.

* **Separation of Concerns (SoC):**

SoC cho phép phân tách các vấn đề nghiệp vụ, dữ liệu và giao diện. Điều này giúp dễ dàng phát triển, thay đổi, và mở rộng ứng dụng mà không gây ảnh hưởng tới các phần còn lại của hệ thống. Web API cũng thừa hưởng được lợi ích này khi phần xử lý yêu cầu được chia rõ ràng giữa Controller và Model.

* **Tính kết hợp và mở rộng:**

MVC hỗ trợ tốt việc kết hợp các API và giao diện người dùng. Bạn có thể dễ dàng tích hợp các API mới hoặc thay đổi giao diện mà không làm ảnh hưởng đến các phần khác của ứng dụng.

* **Đơn giản và dễ sử dụng:**

Mô hình MVC giúp đơn giản hóa quá trình phát triển phần mềm bằng cách phân tách rõ ràng các thành phần của ứng dụng. Điều này giúp các lập trình viên dễ dàng làm quen và phát triển ứng dụng, đặc biệt là trong các dự án Web API nơi mà các phần như HTTP requests và responses được xử lý rõ ràng.

**2.1.3. Khái niệm lập trình mô hình Razor**

**Razor** là một engine template được sử dụng trong các ứng dụng web để tạo ra các trang động, đặc biệt là trong các ứng dụng sử dụng ASP.NET MVC hoặc ASP.NET Core. Razor giúp tách biệt phần mã HTML với mã logic, cho phép tạo ra các trang web linh hoạt và dễ bảo trì.

Razor là một cú pháp nhẹ và dễ sử dụng để kết hợp HTML với mã C#. Đây là một phần quan trọng trong việc xây dựng giao diện người dùng (View) trong mô hình MVC.

**2.1.4. Thành phần Razor**

* **Razor Views (Razor Pages)**: Đây là các file .cshtml chứa mã HTML kết hợp với mã C# để tạo ra giao diện động. Khi người dùng truy cập trang, Razor sẽ thực thi mã C# và tạo ra HTML để gửi đến trình duyệt.
* **Razor Syntax**: Cú pháp Razor rất đơn giản, sử dụng ký tự @ để bắt đầu chèn mã C# vào trong HTML.
* Ví dụ: @DateTime.Now sẽ hiển thị ngày và giờ hiện tại.

**Model Binding trong Razor**: Trong Razor, bạn có thể dễ dàng truyền dữ liệu từ Controller vào View thông qua cơ chế Model Binding. Model Binding cho phép bạn sử dụng các dữ liệu từ Model và hiển thị chúng trong View.

**2.1.5. Lợi ích của Razor trong Web API và MVC**

* **Tích hợp dễ dàng với MVC**:
  + Razor dễ dàng tích hợp với mô hình MVC, đặc biệt là phần View. Bạn có thể dễ dàng sử dụng Razor để tạo ra giao diện người dùng động, giúp tách biệt rõ ràng logic và giao diện trong các ứng dụng web.
* **Chức năng mạnh mẽ và linh hoạt**:
  + Razor cho phép bạn nhúng logic C# vào trong HTML để tạo ra các trang động, ví dụ: duyệt qua danh sách, điều kiện hiển thị nội dung, hay nhập dữ liệu từ người dùng.
* **Dễ sử dụng và học hỏi**:
  + Razor cung cấp cú pháp ngắn gọn, dễ hiểu và dễ học. Bạn không cần phải viết mã HTML phức tạp hay sử dụng JavaScript nhiều, vì phần lớn logic sẽ được xử lý bằng C#.
* **Tính tương thích với Web API**:
  + Trong Web API, Razor có thể được sử dụng để tạo ra các trang web động mà API có thể trả về. Khi cần thiết, các API có thể trả về các trang Razor thay vì chỉ trả về dữ liệu (JSON hoặc XML). Điều này giúp tạo ra giao diện người dùng động khi cần.

**2.1.6. Razor trong ASP.NET Core**

Trong ASP.NET Core, Razor được sử dụng rộng rãi để tạo các View trong MVC hoặc các trang Razor Pages. Razor Pages là một cách đơn giản hơn để xây dựng ứng dụng web mà không cần phải quản lý Controller riêng biệt, rất hữu ích cho các ứng dụng web nhẹ và trang động.

Razor giúp cho việc xây dựng giao diện trở nên dễ dàng và mạnh mẽ trong việc kết hợp với các tính năng của MVC và Web API.

**2.2 Công nghệ sử dụng.**

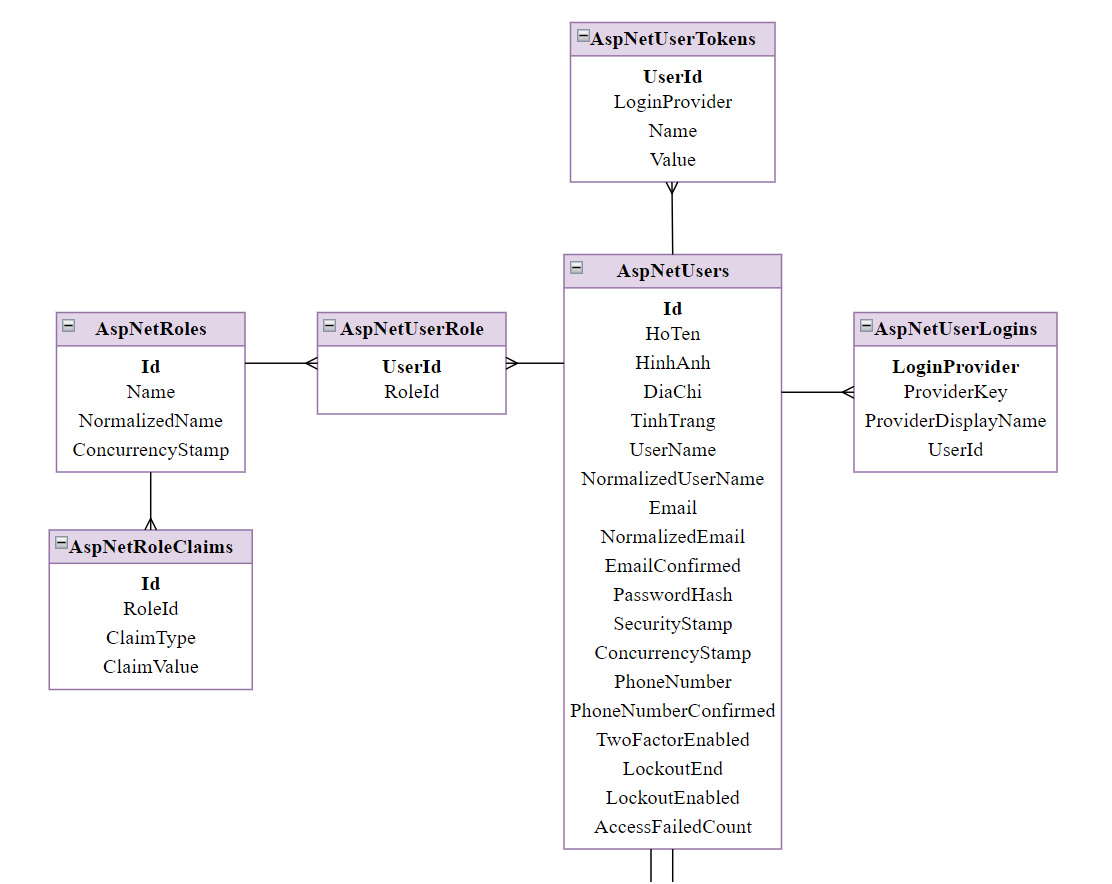
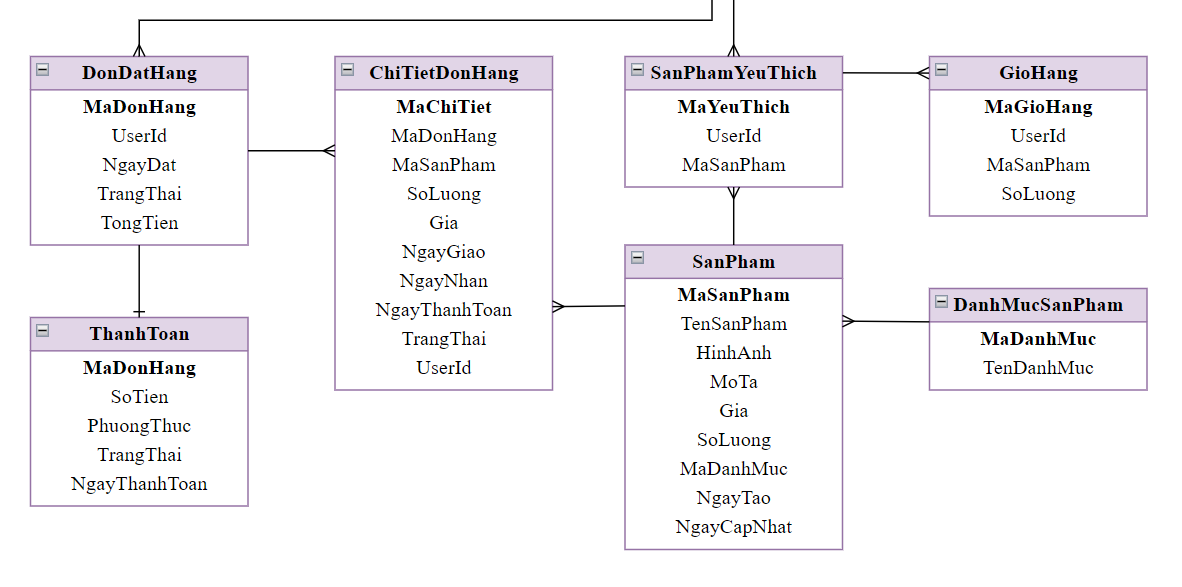
**2.2.1 Visual Studio 2022 (version 8.0)**

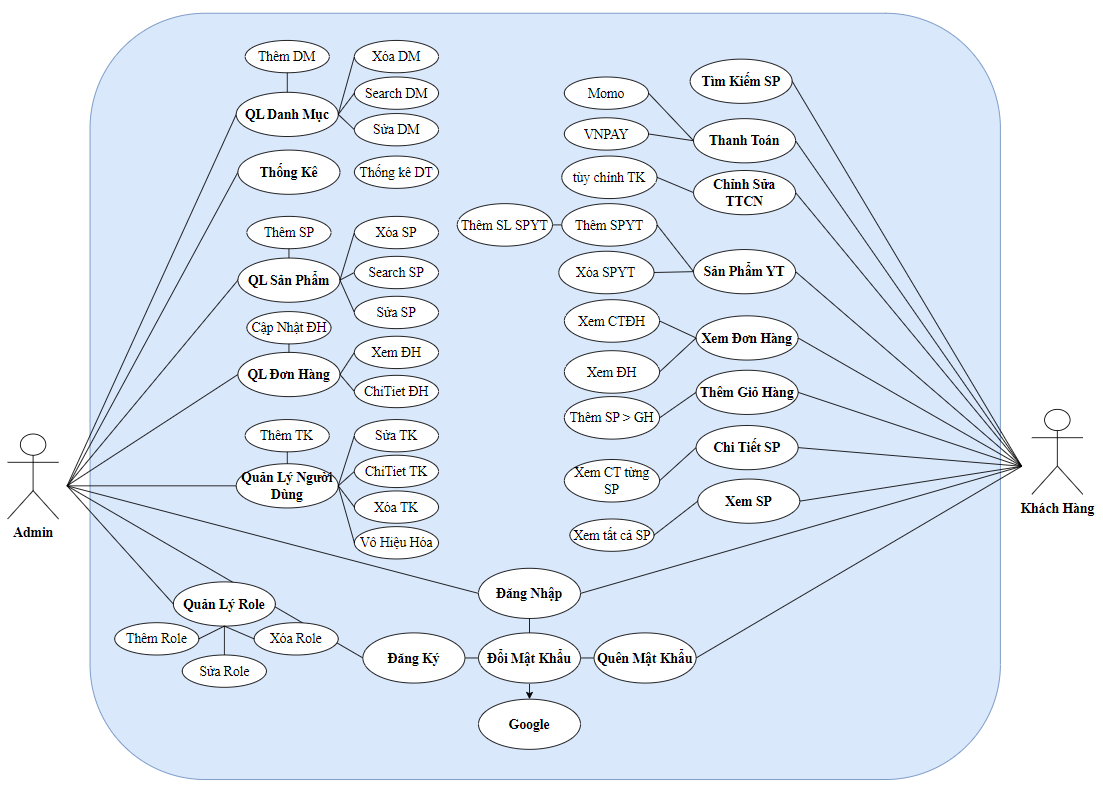
* **Visual Studio** là một công cụ lập trình phổ biến của Microsoft, được thiết kế để hỗ trợ lập trình các ứng dụng Web và phần mềm một cách hiệu quả nhất. Nó hỗ trợ nhiều ngôn ngữ lập trình như **C#**, **VB.NET**, **HTML**, **CSS**, **Javascript** và nhiều ngôn ngữ khác, giúp lập trình viên phát triển và triển khai các ứng dụng một cách nhanh chóng và dễ dàng.
* **Visual Studio** được phát triển và cập nhật thường xuyên, cung cấp các tính năng mạnh mẽ như công cụ gỡ lỗi (**Debugging**), quản lý dự án và hỗ trợ tích hợp các công cụ khác trong hệ sinh thái của Microsoft.
* **Hệ quản trị cơ sở dữ liệu:** SQL Server
* **Front-end:** Lazor (Sever, Assembly)
* **Back-end:** ASP.NET.

**2.3 Phân công công việc**

**2.3.1. Giai đoạn (1, 2)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nội**  **dung** | **Thực hiện** | **Bắt**  **đầu** | **Kết**  **thúc** | **Chỉnh sửa** | **Kiểm duyệt** | **Trạng thái** |
| Phân tích yêu cầu |  |  |  |  |  |  |
| Soạn tài liệu |  |  |  |  |  |  |
| ERD |  |  |  |  |  |  |
| UseCase |  |  |  |  |  |  |
| Thiết kế UI/UX |  |  |  |  |  |  |
| Nội dung |  |  |  |  |  |  |
| Lazor |  |  |  |  |  |  |
| Home\_API |  |  |  |  |  |  |
| Login\_API |  |  |  |  |  |  |
| Register\_API |  |  |  |  |  |  |
| Roles\_API |  |  |  |  |  |  |
| SanPham\_API |  |  |  |  |  |  |
| SanPhamYeuThich\_API |  |  |  |  |  |  |
| GioHang\_API |  |  |  |  |  |  |
| DonHang\_API |  |  |  |  |  |  |
| DanhMucSanPham\_API |  |  |  |  |  |  |
| ChiTietDonHang\_API |  |  |  |  |  |  |
| AccountAdmin\_API |  |  |  |  |  |  |
| Test các chức năng |  |  |  |  |  |  |
| Báo cáo dự án |  |  |  |  |  |  |
| Thiết kế slide |  |  |  |  |  |  |

**2.4 Phân tích và thiết kế ERD**

**2.5 Phân tích và thiết kế UseCase**

**Mô tả:** Đây là Biểu đồ Use Case mô tả các chức năng chính của hệ thống dành cho Admin và Khách Hàng.

**1. Các tác nhân (Actors):**

* **Admin:** Người quản trị hệ thống, có quyền quản lý danh mục, sản phẩm, đơn hàng, người dùng, vai trò, và thống kê dữ liệu.
* **Khách Hàng:** Người sử dụng hệ thống để tìm kiếm, mua sắm sản phẩm, quản lý giỏ hàng, đơn hàng, và tài khoản cá nhân.

**2. Các nhóm Use Case:**

**a. Chức năng của Admin:**

1. **Quản lý danh mục (QL Danh Mục):**
   * Thêm DM (Danh mục)
   * Xóa DM
   * Search DM (Tìm kiếm danh mục)
   * Sửa DM (Chỉnh sửa danh mục)
2. **Quản lý sản phẩm (QL Sản Phẩm):**
   * Thêm SP (Sản phẩm)
   * Xóa SP
   * Search SP (Tìm kiếm sản phẩm)
   * Sửa SP
   * Xem ĐH (Đơn hàng)
3. **Quản lý đơn hàng (QL Đơn Hàng):**
   * Cập nhật ĐH
   * Xem ĐH
   * Chi tiết ĐH
4. **Quản lý người dùng (Quản Lý Người Dùng):**
   * Thêm TK (Tài khoản)
   * Sửa TK
   * Chi Tiết TK
   * Xóa TK
   * Vô hiệu hóa tài khoản
5. **Quản lý vai trò (Quản Lý Role):**
   * Thêm Role
   * Xóa Role
   * Sửa Role
6. **Thống kê (Thống Kê):**
   * Thống kê doanh thu (Thống kê DT)

**b. Chức năng của Khách Hàng:**

1. **Tìm kiếm và mua sản phẩm**:
   * Tìm kiếm SP (Sản phẩm)
   * Xem SP
   * Xem chi tiết SP
   * Thêm SP vào giỏ hàng
   * Xem đơn hàng
2. **Quản lý giỏ hàng**:
   * Thêm giỏ hàng
3. **Thanh toán**:
   * Hỗ trợ các phương thức: Momo, VNPAY
4. **Quản lý tài khoản**:
   * Đăng nhập, đăng ký
   * Đổi mật khẩu, quên mật khẩu
   * Chỉnh sửa thông tin cá nhân (Tùy chỉnh TK)
5. **Sản phẩm y tế (Sản Phẩm YT)**:
   * Thêm SP YT
   * Xóa SP YT
   * Xem chi tiết đơn hàng (CT ĐH)
   * Xem chi tiết từng SP
   * Xem tất cả sản phẩm

**3. Các chức năng chung (dùng chung cho cả Admin & Khách hàng):**

* Đăng nhập, đăng ký
* Đổi mật khẩu, quên mật khẩu
* Hỗ trợ đăng nhập Google

**3. Quy trình nghiệp vụ**

**3.1 Quản lý danh mục**

**Mô tả:** Admin thực hiện các thao tác thêm, sửa, xóa, tìm kiếm danh mục sản phẩm. **Bước thực hiện:**

1. Admin đăng nhập vào hệ thống.
2. Truy cập vào trang Quản lý danh mục.
3. Chọn một trong các chức năng:
   * **Thêm danh mục:** Nhập thông tin danh mục và lưu vào hệ thống.
   * **Sửa danh mục:** Chỉnh sửa thông tin danh mục đã có.
   * **Xóa danh mục:** Xóa danh mục khỏi hệ thống.
   * **Tìm kiếm danh mục:** Nhập từ khóa để tìm danh mục cần quản lý.
4. Hệ thống xử lý và hiển thị kết quả.

**3.2 Quản lý sản phẩm**

**Mô tả:** Admin có thể thêm, sửa, xóa, tìm kiếm sản phẩm

**Bước thực hiện:**

1. Admin đăng nhập vào hệ thống.
2. Truy cập vào Quản lý sản phẩm.
3. **Chọn một trong các thao tác sau:**
   * **Thêm sản phẩm:** Nhập thông tin sản phẩm, chọn danh mục, hình ảnh, giá bán.
   * **Sửa sản phẩm:** Chỉnh sửa thông tin sản phẩm đã có.
   * **Xóa sản phẩm:** Xóa sản phẩm khỏi hệ thống.
   * **Tìm kiếm sản phẩm:** Nhập từ khóa để tìm sản phẩm.
4. Hệ thống cập nhật thông tin và phản hồi kết quả.
   1. **Quản lý đơn hàng**

**Mô tả: Admin có thể theo dõi, cập nhật trạng thái đơn hàng.**

**Bước thực hiện**:

1. Admin đăng nhập vào hệ thống.
2. Truy cập vào **Quản lý đơn hàng**.
3. Chọn **Xem đơn hàng** để kiểm tra danh sách đơn.

* Nhấp vào từng đơn hàng để **xem chi tiết**.
* Chọn **Cập nhật đơn hàng** để thay đổi trạng thái đơn (Chờ xử lý, Đang giao, Hoàn thành, Hủy).

1. Hệ thống ghi nhận thay đổi và thông báo kết quả.
   1. **Quản lý người dùng**

**Mô tả**: Admin có thể thêm, chỉnh sửa, xóa, vô hiệu hóa tài khoản người dùng.  
**Bước thực hiện**:

1. Admin đăng nhập vào hệ thống.
2. Truy cập vào **Quản lý người dùng**.
3. Chọn một trong các thao tác:
   * **Thêm tài khoản**: Nhập thông tin người dùng, cấp quyền truy cập.
   * **Sửa tài khoản**: Chỉnh sửa thông tin tài khoản người dùng.
   * **Xem chi tiết tài khoản**: Xem thông tin cá nhân của người dùng.
   * **Xóa tài khoản**: Xóa tài khoản khỏi hệ thống.
   * **Vô hiệu hóa tài khoản**: Khóa tài khoản người dùng tạm thời.
4. Hệ thống xử lý và hiển thị kết quả.
   1. **Quản lý vai trò (Role)**

**Mô tả**: Admin có thể thêm, sửa, xóa quyền hạn của người dùng.  
**Bước thực hiện**:

1. Admin đăng nhập vào hệ thống.
2. Truy cập vào **Quản lý Role**.
3. Chọn một trong các thao tác:
   * **Thêm Role**: Tạo mới vai trò (Admin, Nhân viên, Khách hàng...).
   * **Sửa Role**: Chỉnh sửa quyền của một vai trò.
   * **Xóa Role**: Xóa một vai trò khỏi hệ thống.
4. Hệ thống ghi nhận thay đổi và thông báo kết quả.
   1. **Thống kê doanh thu**

**Mô tả**: Admin có thể xem báo cáo doanh thu.  
**Bước thực hiện**:

1. Admin đăng nhập vào hệ thống.
2. Truy cập vào **Thống kê**.
3. Chọn loại báo cáo cần xem (Doanh thu theo ngày, tháng, năm).
4. Hệ thống hiển thị biểu đồ và số liệu thống kê.

**4. Quy trình nghiệp vụ của Khách hàng**

**4.1. Tìm kiếm và xem sản phẩm**

**Mô tả**: Khách hàng có thể tìm kiếm, xem chi tiết sản phẩm trước khi mua.  
**Bước thực hiện**:

1. Khách hàng truy cập trang web.
2. Sử dụng chức năng **Tìm kiếm sản phẩm** để tìm mặt hàng mong muốn.
3. Nhấp vào sản phẩm để xem **chi tiết sản phẩm**.

**4.2. Thêm sản phẩm vào giỏ hàng**

**Mô tả**: Khách hàng có thể thêm sản phẩm vào giỏ để mua sau.  
**Bước thực hiện**:

1. Khách hàng đăng nhập vào hệ thống.
2. Tìm sản phẩm muốn mua.
3. Nhấp **Thêm vào giỏ hàng**.
4. Hệ thống xác nhận sản phẩm đã được thêm vào giỏ.

**4.3. Đặt hàng và thanh toán**

**Mô tả**: Khách hàng có thể đặt hàng và thanh toán trực tuyến.  
**Bước thực hiện**:

1. Khách hàng truy cập **Giỏ hàng**.
2. Xem lại danh sách sản phẩm, chọn số lượng, ghi chú.
3. Nhấn **Thanh toán**.
4. Chọn phương thức thanh toán (Momo, VNPAY).
5. Xác nhận đơn hàng và hoàn tất thanh toán.
6. Hệ thống thông báo kết quả đặt hàng.

**4.4. Xem đơn hàng**

**Mô tả**: Khách hàng có thể theo dõi tình trạng đơn hàng của mình.  
**Bước thực hiện**:

1. Khách hàng đăng nhập.
2. Truy cập **Xem đơn hàng**.
3. Xem danh sách đơn hàng đã đặt.
4. Nhấp vào từng đơn hàng để xem **Chi tiết đơn hàng**.

**4.5. Quản lý tài khoản**

**Mô tả**: Khách hàng có thể đăng ký, đăng nhập, thay đổi mật khẩu và cập nhật thông tin cá nhân.  
**Bước thực hiện**:

1. **Đăng ký**:
   * Truy cập trang đăng ký.
   * Nhập thông tin cá nhân, email, mật khẩu.
   * Xác nhận đăng ký.
2. **Đăng nhập**:
   * Nhập email và mật khẩu.
   * Hệ thống xác thực và cho phép truy cập.
3. **Đổi mật khẩu**:
   * Truy cập **Đổi mật khẩu**.
   * Nhập mật khẩu cũ, mật khẩu mới.
   * Hệ thống xác nhận và cập nhật.
4. **Quên mật khẩu**:
   * Nhập email để nhận liên kết đặt lại mật khẩu.
5. **Chỉnh sửa thông tin cá nhân**:
   * Cập nhật thông tin cá nhân (Tên, địa chỉ, số điện thoại...).

**5. Yêu cầu chức năng**

|  |  |  |
| --- | --- | --- |
| **Đối tượng** | **Chức năng** | **Ý nghĩa** |
| **Admin** | Quản lý danh mục | Cho phép Admin thêm, sửa, xóa, tìm kiếm danh mục sản phẩm. |
| **Admin** | |  | | --- | |  |  |  | | --- | | Quản lý sản phẩm | | Hỗ trợ Admin thêm, chỉnh sửa, xóa, tìm kiếm sản phẩm trong hệ thống. |
| **Admin** | Quản lý đơn hàng | |  | | --- | |  |  |  | | --- | | Giúp Admin theo dõi, cập nhật trạng thái đơn hàng để đảm bảo giao dịch diễn ra suôn sẻ. | |
| **Admin** | Quản lý người dùng | Cung cấp quyền thêm, sửa, xóa hoặc vô hiệu hóa tài khoản khách hàng hoặc nhân viên. |
| **Admin** | Quản lý vai trò (Role) | Quản lý và phân quyền cho từng nhóm người dùng khác nhau trong hệ thống. |
| **Admin** | Thống kê doanh thu | Xem báo cáo doanh thu theo ngày, tháng, năm để đưa ra quyết định kinh doanh. |
| **Khách Hàng** | Đăng ký tài khoản | Cho phép khách hàng tạo tài khoản mới để sử dụng dịch vụ. |
| **Khách Hàng** | Đăng nhập/Đăng xuất | Hỗ trợ khách hàng truy cập hệ thống và bảo vệ tài khoản cá nhân. |
| **Khách Hàng** | Tìm kiếm sản phẩm | Giúp khách hàng nhanh chóng tìm được sản phẩm mong muốn. |
| **Khách Hàng** | Xem chi tiết sản phẩm | |  | | --- | |  |  |  | | --- | | Cung cấp thông tin cụ thể về sản phẩm như giá, mô tả, đánh giá. | |
| **Khách Hàng** | Thêm sản phẩm vào giỏ hàng | Cho phép khách hàng lưu sản phẩm vào giỏ để mua sau. |
| **Khách Hàng** | Đặt hàng và thanh toán | Hỗ trợ khách hàng mua hàng và thanh toán qua Momo, VNPAY. |
| **Khách Hàng** | Xem và theo dõi đơn hàng | Cung cấp thông tin về trạng thái đơn hàng sau khi đặt mua. |
| **Khách Hàng** | Đổi mật khẩu | Cho phép khách hàng thay đổi mật khẩu để bảo vệ tài khoản. |
| **Khách Hàng** | Quên mật khẩu | Hỗ trợ khách hàng lấy lại mật khẩu khi quên thông tin đăng nhập. |
| **Khách Hàng** | |  | | --- | |  |  |  | | --- | | Gửi thông báo | | Gửi thông báo xác nhận đơn hàng, cập nhật trạng thái đơn hàng cho khách hàng. |
| **Khách Hàng** | Xác thực thanh toán | Đảm bảo giao dịch thanh toán được xử lý an toàn qua cổng Momo, VNPAY. |