|  |  |
| --- | --- |
| **NGUYỄN BẢO NGỌC** | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  --------------------------------------        **ĐỒ ÁN TỐT NGHIỆP**  NGÀNH HỆ THỐNG THÔNG TIN  **XÂY DỰNG WEBSITE BÁN HÀNG THỂ THAO SUPERSPORTS**  **CBHD: TS. Đặng Trọng Hợp**  **Sinh viên: Nguyễn Bảo Ngọc**  **Mã số sinh viên: 2021602053**  **Hà Nội – 2025** |
| **HỆ THỐNG THÔNG TIN** |

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

--------------------------------------

**ĐỒ ÁN TỐT NGHIỆP**

NGÀNH HỆ THỐNG THÔNG TIN

**XÂY DỰNG WEBSITE BÁN HÀNG THẾ THAO SUPERSPORTS**

**CBHD: TS. Đặng Trọng Hợp**

**Sinh viên: Nguyễn Bảo Ngọc**

**Mã số sinh viên: 2021602053**

**Hà Nội – 2025**

**LỜI CẢM ƠN**

Lời đầu tiên, em muốn bày tỏ lòng biết ơn sâu sắc đến các thầy cô Trường Đại học Công nghiệp Hà Nội, đặc biệt là các thầy cô trong khoa Công nghệ thông tin đã tận tình dạy dỗ và hướng dẫn em trong suốt quãng thời gian học tập ở trường. Sự hiểu biết, tận tâm và kiến thức chuyên môn của các thầy, cô chính là nguồn động viên to lớn, là hành trang cho em trong quá trình chinh phục con đường học tập và sự nghiệp.

Đặc biệt, em xin gửi lời cảm ơn chân thành tới thầy Đặng Trọng Hợp, người đã giúp em vượt qua những thách thức và phát triển bản thân, cảm ơn thầy đã tận tình dõi theo, giải đáp thắc mắc trong suốt quãng thời gian vừa qua. Những lời khuyên, định hướng và sự hỗ trợ của thầy đã giúp em hoàn thành đề tài của mình một cách tốt nhất.

Học phần Đồ án tốt nghiệp là một học phần vô cùng quan trọng, đó là minh chứng cho những kiến thức đã có sau bốn năm học tập tại trường.Quá trình thực hiện đồ án không chỉ là thách thức còn là cơ hội để em có thể áp dụng những kiến thức đã học vào thực tế. Em đã cố gắng hết mình để có thể hoàn thiện đồ án, song em vẫn luôn trân trọng và mong nhận được những sự đóng góp ý kiến quý báu của các thầy cô để bài làm của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Nguyễn Bảo Ngọc

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc198146571)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 3](#_Toc198146572)

[1.1. Giới thiệu về ASP.NET Core 3](#_Toc198146573)

[1.2.1. Khái niệm 3](#_Toc198146574)

[1.2.2. Cơ chế hoạt động của ASP.Net Core MVC 3](#_Toc198146575)

[1.2.3. Ưu điểm của ASP.NET Core 4](#_Toc198146576)

[1.2. Giới thiệu ngôn ngữ C# 4](#_Toc198146577)

[1.3. Giới thiệu về Entity Framework 8](#_Toc198146578)

[1.4. Giới thiệu về cơ sở dữ liệu SQL Server 10](#_Toc198146579)

[CHƯƠNG 2. KHẢO SÁT, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 14](#_Toc198146580)

[2.1. Khảo sát hệ thống 14](#_Toc198146581)

[2.1.1 Giới thiệu về hệ thống 14](#_Toc198146582)

[2.1.2 Các yêu cầu của hệ thống 15](#_Toc198146583)

[2.2. Phân tích hệ thống 18](#_Toc198146584)

[2.2.1. Mô hình use case 18](#_Toc198146585)

[2.2.2. Mô hình dữ liệu 35](#_Toc198146586)

[2.2.3. Phân tích use case 36](#_Toc198146587)

[2.3. Thiết kế hệ thống 52](#_Toc198146588)

[2.3.1 Thiết kế cơ sở dữ liệu 52](#_Toc198146589)

[2.3.2 Thiết kế giao diện 61](#_Toc198146590)

[CHƯƠNG 3. CÀI ĐẶT VÀ KẾT QUẢ 62](#_Toc198146591)

[3.1. Cài đặt hệ thống 62](#_Toc198146592)

[3.1.1. Yêu cầu cấu hình 62](#_Toc198146593)

[3.1.2. Hướng dẫn cài đặt 63](#_Toc198146594)

[3.2. Kết quả 64](#_Toc198146595)

[3.2.1. Giao diện người dùng 64](#_Toc198146596)

[3.2.2. Giao diện quản trị 64](#_Toc198146597)

[3.3. Kiểm thử 64](#_Toc198146598)

[3.3.1. Kế hoạch kiểm thử 64](#_Toc198146599)

[3.3.2. Lịch trình kiểm thử 64](#_Toc198146600)

[KẾT LUẬN 65](#_Toc198146601)

[TÀI LIỆU THAM KHẢO 66](#_Toc198146602)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1 Cơ chế hoạt động ASP.NET Core MVC 3](#_Toc198147698)

[Hình 2.1. Biểu đồ use case tổng quan của hệ thống 18](#_Toc198147699)

[Hình 2.2. Usecase đăng ký 18](#_Toc198147700)

[Hình 2.3: Use Đăng nhập 20](#_Toc198147701)

[Hình 2.4. Usecase quên mật khẩu 21](#_Toc198147702)

[Hình 2.5. Usecase tìm kiếm sản phẩm 22](#_Toc198147703)

[Hình 2.6. Usecase thanh toán 24](#_Toc198147704)

[Hình 2.7. Usecase đánh giá sản phẩm 25](#_Toc198147705)

[Hình 2.8. Usecase quản lý sản phẩm 26](#_Toc198147706)

[Hình 2.9. Usecase quản lý đơn hàng 28](#_Toc198147707)

[Hình 2.10. Usecase xem báo cáo tài chính 29](#_Toc198147708)

[Hình 2.11. Usecase quản lý nhân viên 30](#_Toc198147709)

[Hình 2.12. Usecase quản lý người dùng 32](#_Toc198147710)

[Hình 2.13. Mô hình dữ liệu 35](#_Toc198147711)

[Hình 2.14. Sơ đồ VOPC use case đăng nhập 36](#_Toc198147712)

[Hình 2.15. Sơ đồ trình tự use case “Đăng nhập” 36](#_Toc198147713)

[Hình 2.16. Sơ đồ VOPC use case đăng ký 37](#_Toc198147714)

[Hình 2.17. Sơ đồ trình tự use case “Đăng ký” 37](#_Toc198147715)

[Hình 2.18. Sơ đồ VOPC use case quên mật khẩu 38](#_Toc198147716)

[Hình 2.19. Sơ đồ tuần tự use case “Quên mật khẩu” 38](#_Toc198147717)

[Hình 2.20. Sơ đồ VOPC use case Tìm kiếm sản phẩm 39](#_Toc198147718)

[Hình 2.21. Biểu đồ trình tự use case “Tìm kiếm sản phẩm” 40](#_Toc198147719)

[Hình 2.22. Sơ đồ VOPC use case “Thanh toán” 41](#_Toc198147720)

[Hình 2.23. Sơ đồ trình tự use case “Thanh toán” 42](#_Toc198147721)

[Hình 2.24. Sơ đồ VOPC use case thêm sản phẩm 43](#_Toc198147722)

[Hình 2.25. Sơ đồ trình tự use case “Thêm sản phẩm” 44](#_Toc198147723)

[Hình 2.26. Sơ đồ VOPC tham gia use case cập nhật sản phẩm 45](#_Toc198147724)

[Hình 2.27. Sơ đồ trình tự use case “Cập nhật sản phẩm” 46](#_Toc198147725)

[Hình 2.28. Sơ đồ VOPC tham gia use case xóa sản phẩm 47](#_Toc198147726)

[Hình 2.29. Sơ đồ trình tự use case “Xóa sản phẩm” 48](#_Toc198147727)

[Hình 2.30. Sơ đồ VOPC use case đánh giá sản phẩm 48](#_Toc198147728)

[Hình 2.31. Sơ đồ trình tự use case “Đánh giá sản phẩm” 49](#_Toc198147729)

[Hình 2.32. Sơ đồ VOPC use case xóa người dùng 50](#_Toc198147730)

[Hình 2.33. Sơ đồ trình tự use case “Xóa người dùng” 50](#_Toc198147731)

[Hình 2.34. Sơ đồ VOPC use case cập nhật thuộc tính sản phẩm 51](#_Toc198147732)

[Hình 2.35. Sơ đồ trình tự use case “Cập nhật thuộc tính sản phẩm” 51](#_Toc198147733)

[Hình 2.36: Thiết kế cơ sở dữ liệu 52](#_Toc198147734)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1. Đặc tả usecase “Đăng ký” 18](#_Toc198146219)

[Bảng 2.2. Đặc tả usecase “Đăng nhập” 20](#_Toc198146220)

[Bảng 2.3. Đặc tả usecase “Quên mật khẩu” 21](#_Toc198146221)

[Bảng 2.4. Đặc tả usecase “Tìm kiếm sản phẩm” 23](#_Toc198146222)

[Bảng 2.5. Đặc tả usecase “Thanh toán” 24](#_Toc198146223)

[Bảng 2.6. Đặc tả usecase “Đánh sản phẩm” 25](#_Toc198146224)

[Bảng 2.7. Đặc tả usecase “Quản lý sản phẩm” 26](#_Toc198146225)

[Bảng 2.8. Đặc tả usecase “Quản lý đơn hàng” 28](#_Toc198146226)

[Bảng 2.9. Đặc tả usecase “Xem báo cáo tài chính” 29](#_Toc198146227)

[Bảng 2.10. Đặc tả usecase “Quản lý nhân viên” 31](#_Toc198146228)

[Bảng 2.11. Đặc tả usecase “Quản lý người dùng” 32](#_Toc198146229)

[Bảng 2.12. Mô tả bảng “Role” 56](#_Toc198146230)

[Bảng 2.13. Mô tả bảng “User” 56](#_Toc198146231)

[Bảng 2.14. Mô tả bảng “Order” 56](#_Toc198146232)

[Bảng 2.15. Mô tả bảng “Product” 57](#_Toc198146233)

[Bảng 2.16. Mô tả bảng “Employee” 57](#_Toc198146234)

[Bảng 2.17. Mô tả bảng “Cart” 57](#_Toc198146235)

[Bảng 2.18. Mô tả bảng “Image” 58](#_Toc198146236)

[Bảng 2.19. Mô tả bảng “OrderDetail” 58](#_Toc198146237)

[Bảng 2.20. Mô tả bảng “Discount” 58](#_Toc198146238)

[Bảng 2.21. Mô tả bảng “Feedback” 59](#_Toc198146239)

[Bảng 2.22. Mô tả bảng “Province” 59](#_Toc198146240)

[Bảng 2.23. Mô tả bảng “Category” 59](#_Toc198146241)

[Bảng 2.24. Mô tả bảng “SubCategory” 60](#_Toc198146242)

[Bảng 2.25. Mô tả bảng “Ward” 60](#_Toc198146243)

[Bảng 2.26. Mô tả bảng “District” 60](#_Toc198146244)

[Bảng 2.27. Mô tả bảng “Brand” 60](#_Toc198146245)

# MỞ ĐẦU

**1. Lý do chọn đề tài:**

Trong thời đại số, mua sắm trực tuyến đã trở thành một phần không thể thiếu trong cuộc sống của con người. Sự bùng nổ của công nghệ đã làm thay đổi hoàn toàn cách chúng ta tìm kiếm và mua các sản phẩm thời trang. Việc mua quần áo trực tuyến đã trở thành xu hướng tất yếu, mang lại sự tiện lợi cho khách hàng và hiệu quả cao cho các đơn vị kinh doanh.

Tuy nhiên, việc quản lý các hoạt động bán hàng truyền thống vẫn còn nhiều hạn chế, từ việc quản lý thủ công thông tin đơn hàng, xác nhận đơn qua điện thoại, đến việc đối mặt với tình trạng quá tải khi có chương trình khuyến mãi hay vào mùa cao điểm. Để giải quyết những vấn đề này, việc xây dựng một hệ thống quản lý bán quần áo trực tuyến trở nên cấp thiết.

Đồ án này tập trung vào việc thiết kế và phát triển **Supersport** – một website bán quần áo trực tuyến dựa trên nền tảng ASP.NET Core MVC. Với sự hỗ trợ của công nghệ hiện đại, hệ thống sẽ giúp khách hàng dễ dàng tìm kiếm, so sánh và mua sắm các sản phẩm thời trang trực tuyến, đồng thời cung cấp cho các đơn vị kinh doanh một công cụ quản lý hiệu quả, chuyên nghiệp. Việc lựa chọn ASP.NET Core MVC là một quyết định sáng suốt, giúp đảm bảo tính bảo mật, ổn định và khả năng mở rộng của hệ thống.

**2. Mục đích nghiên cứu**

* Tạo ra một hệ thống hiệu quả và tiện lợi giúp khách hàng dễ dàng tìm kiếm, lựa chọn và mua sắm các sản phẩm quần áo trực tuyến.
* Cung cấp cho các nhà quản lý cửa hàng những thông tin về đơn hàng, tồn kho sản phẩm, doanh thu và tình hình kinh doanh để đưa ra các quyết định kịp thời và chính xác.

**3. Đối tượng nghiên cứu và phạm vi nghiên cứu**

* Tìm hiểu các tác vụ của người dùng có trong website.
* Nghiên cứu và sử dụng ngôn ngữ HTML, CSS, JavaScript và thư viện Bootstrap để xây dựng giao diện website.
* Sử dụng hệ quản trị cơ sở dữ liệu SQL Server để xây dựng và quản trị cơ sở dữ liệu của website.
* Tìm hiểu về mô hình MVC (Model-View-Controller) và các tính năng có trong ASP.NET Core MVC như Routing, Authentication, Authorization, Loggin, SingalR...
* Sử dụng framework ASP.NET Core MVC có thể kết nối với cơ sở dữ liệu và kết hợp giao diện để xây dựng website bán hàng thể thao.

**4. Phương pháp nghiên cứu**

* Tự học các kiến thức ở trên mạng, xem các video hướng dẫn liên quan đến đề tài.
* Tham khảo các website bán hàng thể thao có trên mạng (SuperSports).
* Tiếp thu bài giảng từ giảng viên hướng dẫn.
* Áp dụng những điều đã học để xây dựng website quản lý hệ thống đặt phòng khách sạn trên framework ASP.NET Core MVC.

**5. Kết quả dự kiến**

* Hiểu được các nghiệp vụ của người dùng có trong website quản lý hệ thống đặt phòng khách sạn.
* Hiểu được mô hình MVC (Model-View-Controller) và các tính năng cơ bản có trong ASP.NET Core MVC.

**6. Ý nghĩa khoa học – thực tiễn**

* Ứng dụng công nghệ hiện đại, góp phần vào phát triển lĩnh vực công nghệ thông tin
* Tích hợp đa dạng công nghệ sử dụng
* Hỗ trợ doanh nghiệp, tang cường trải nghiệm người dung
* Giải quyết hạn chế của phương pháp truyền thống, thúc đẩy chuyển đổi số

**7. Bố cục báo cáo**

Đồ án gồm 3 chương như sau:

* CHƯƠNG 1: Cơ sở lý thuyết.
* CHƯƠNG 2: Khảo sát, phân tích và thiết kế hệ thống.
* CHƯƠNG 3: Cài đặt và kết quả.

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

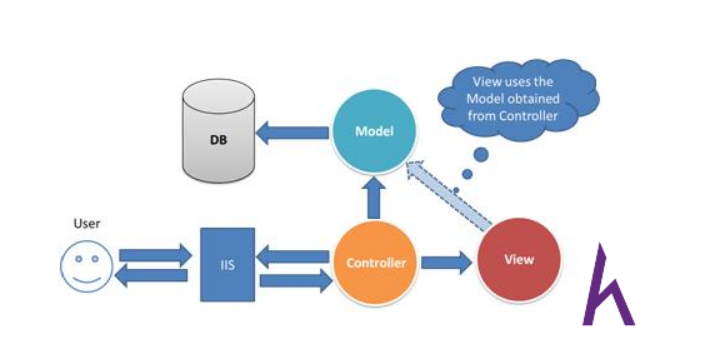
## Giới thiệu về ASP.NET Core

### 1.2.1. Khái niệm

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.

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.

### 1.2.2. Cơ chế hoạt động của ASP.Net Core MVC

****

Hình 1.1 Cơ chế hoạt động ASP.NET Core MVC

* User gửi một yêu cầu tới server bằng cách truyền vào 1 URL trong browser.
* Yêu cầu đó được gửi tới Controller đầu tiên, Controller sẽ xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì Controller sẽ chuyển qua tầng Model.
* Tại tầng Model, dữ liệu được truy xuất từ database và sau đó truyền qua View thông qua Controller.
* Controller sẽ giúp dữ liệu được chuyển từ Model qua View.
* View là tầng cuối cùng giao tiếp với User, mọi dữ liệu sẽ được hiển thị cho. User thông qua tầng View.

### 1.2.3. Ưu điểm của ASP.NET Core

* 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.

## Giới thiệu ngôn ngữ C#

C# là một ngôn ngữ lập trình mạnh mẽ và đa mục đích, được phát triển bởi Microsoft vào những năm 2000. C# là một phần của nền tảng phát triển phần mềm .NET Framework, và nó được sử dụng rộng rãi để xây dựng ứng dụng từ các ứng dụng máy tính đơn giản cho đến các ứng dụng di động, ứng dụng web và hệ thống phần mềm phức tạp.

Dưới đây là một số đặc điểm chính của ngôn ngữ C#:

* Tính đa nền tảng: C# là một ngôn ngữ đa nền tảng, có thể chạy trên nhiều hệ điều hành như Windows, macOS và Linux. Điều này cho phép nhà phát triển viết mã một lần và chạy ứng dụng trên nhiều nền tảng khác nhau, giảm thiểu công sức phát triển và tăng tính linh hoạt.
* Tính dễ học và dễ sử dụng: C# có cú pháp gần gũi với ngôn ngữ C++ và Java, điều này làm cho việc học và chuyển đổi từ các ngôn ngữ khác sang C# trở nên dễ dàng. Nó hỗ trợ mô hình lập trình hướng đối tượng (OOP), cho phép nhà phát triển xây dựng các đối tượng có tính chất, phương thức và sự tương tác giữa chúng.
* Công cụ phát triển tích hợp: C# được tích hợp chặt chẽ với Visual Studio, một môi trường phát triển tích hợp (IDE) mạnh mẽ từ Microsoft. Visual Studio cung cấp các công cụ hỗ trợ mạnh mẽ như gỡ lỗi, thông báo lỗi cú pháp, hoàn thành mã tự động và nhiều tính năng khác để tăng hiệu suất và sức sáng tạo của nhà phát triển.
* Hỗ trợ rộng rãi cho các loại ứng dụng: C# có thể được sử dụng để phát triển rất nhiều loại ứng dụng, bao gồm ứng dụng máy tính, ứng dụng di động (qua Xamarin), ứng dụng web (qua ASP.NET và Blazor), ứng dụng trò chơi và nhiều hơn nữa. Điều này cho phép nhà phát triển tận dụng sức mạnh của C# để xây dựng các ứng dụng đa dạng và đáp ứng nhu cầu của nhiều lĩnh vực khác nhau.
* Hệ sinh thái mạnh mẽ: C# có một cộng đồng lập trình viên đông đảo và hệ sinh thái phong phú. Có rất nhiều tài liệu, sách, diễn đàn và nguồn học trực tuyến để hỗ trợ nhà phát triển trong việc học và phát triển ứng dụng bằng C#. Microsoft và cộng đồng cũng cung cấp các thư viện và framework mạnh mẽ như Entity Framework, ASP.NET Core, Xamarin và nhiều công nghệ khác để giúp nhà phát triển nhanh chóng xây dựngcác ứng dụng và giảm thiểu công việc lặp lại.
* Tính bảo mật và kiểm soát lỗi: C# cung cấp các tính năng bảo mật tích hợp như kiểm soát truy cập, quản lý bộ nhớ tự động và xử lý ngoại lệ. Điều này giúp ngăn chặn các lỗ hổng bảo mật và giảm rủi ro lỗi trong quá trình phát triển và triển khai ứng dụng.
* Hỗ trợ cho phát triển đội ngũ: C# được thiết kế để hỗ trợ phát triển đội ngũ và làm việc nhóm. Nó cung cấp các công cụ như quản lý mã nguồn, hệ thống kiểm soát phiên bản và tích hợp liên kết dữ liệu, giúp các nhà phát triển làm việc cùng nhau và duy trì mã nguồn dễ dàng.
* Tương thích và tích hợp: C# tương thích tốt với các ngôn ngữ và công nghệ khác trong hệ sinh thái .NET, bao gồm VB.NET, F#, ASP.NET và nhiều hơn nữa. Nó cũng có khả năng tích hợp với các công nghệ khác như SQL Server, Azure và các dịch vụ web khác, tạo điều kiện cho việc phát triển ứng dụng phức tạp và kết nối với các hệ thống khác.
* Hỗ trợ và cập nhật liên tục: C# được Microsoft hỗ trợ và phát triển liên tục. Cập nhật và phiên bản mới được phát hành thường xuyên để cải thiện hiệu suất, tăng tính năng và sửa các lỗi. Điều này đảm bảo rằng ngôn ngữ C# luôn giữ được tính tương thích và đáp ứng các xu hướng công nghệ mới.

Các phiên bản của ngôn ngữ C#:

* C# 1.0 (2002): Đây là phiên bản đầu tiên của ngôn ngữ C#. Nó được giới thiệu cùng với .NET Framework 1.0. C# 1.0 cung cấp các tính năng cơ bản như lớp, đối tượng, kế thừa, giao diện, các kiểu dữ liệu cơ bản, xử lý ngoại lệ, tham số hóa kiểu và quản lý bộ nhớ tự động.
* C# 2.0 (2005): Phiên bản này được phát hành cùng với .NET Framework 2.0 và Visual Studio 2005. C# 2.0 bổ sung nhiều tính năng mới như kiểu đối tượng tùy chỉnh, phạm vi (partial types), phạm vi tùy chỉnh (nullable types), phạm vi tùy chỉnh (anonymous methods), kiểu dữ liệu var, phạm vi mở rộng (extension methods) và lớp generic.
* C# 3.0 (2007): Được phát hành cùng với .NET Framework 3.5 và Visual Studio 2008, C# 3.0 mang đến một số tính năng mới quan trọng như cú pháp LINQ (Language Integrated Query), biểu thức lambda, khởi tạo đối tượng tự động (automatic properties), khởi tạo đối tượng kiểu vô danh (anonymous types) và phạm vi mở rộng LINQ (LINQ extension methods).
* C# 4.0 (2010): Phiên bản này được phát hành cùng với .NET Framework 4.0 và Visual Studio 2010. C# 4.0 bổ sung các tính năng như đặc tả tên tham số tùy chọn (optional parameter), tham số tùy chỉnh (named parameter), kiểu động (dynamic type), kiểu phạm vi động (covariant return type) và phạm vi nghịch đảo (contravariant type parameter).
* C# 5.0 (2012): Được phát hành cùng với .NET Framework 4.5 và Visual Studio 2012, C# 5.0 đưa ra các tính năng như phương thức bất đồng bộ (asynchronous methods), từ khóa await/async, khóa đè (caller info attribute) và biểu thức lambda với các tham số tùy chọn.
* C# 6.0 (2015): Phiên bản này được phát hành cùng với .NET Framework 4.6 và Visual Studio 2015. C# 6.0 mang đến các cải tiến như biểu thức null-conditional (null-conditional expressions), chuỗi kiểu (string interpolation), khai báo using tĩnh (static using directives), khai báo tên tùy chỉnh (custom name declarations), khởi tạo chỉ mục tĩnh (static index initializers) và các tính năng khác.
* C# 7.0 (2017): Phiên bản này được phát hành cùng với .NET Core 2.0 và Visual Studio 2017. C# 7.0 bổ sung nhiều tính năng mới như tuple, mẫu khớp (pattern matching), phương thức mở rộng chỉ mục (indexers), tham số khởi tạo (ref locals and returns), bắt ngoại lệ (throw expressions) và các cải tiến khác.
* C# 8.0 (2019): Được phát hành cùng với .NET Core 3.0 và Visual Studio 2019, C# 8.0 giới thiệu các tính năng như: danh sách không hoàn chỉnh vì giới hạn độ dài. Để biết thêm thông tin chi tiết về các phiên bản và lịch sử phát triển của ngôn ngữ C#, bạn có thể tham khảo tài liệu chính thức từ Microsoft hoặc các nguồn tài liệu lập trình C# khác.

Tóm lại, C# là một ngôn ngữ lập trình mạnh mẽ, đa nền tảng và dễ học. Nó cung cấp các công cụ và hệ sinh thái phong phú để phát triển ứng dụng từ đơn giản đến phức tạp, đáp ứng nhu cầu của nhiều lĩnh vực khác nhau. Với tính linh hoạt, tính bảo mật và khả năng tích hợp, C# là một lựa chọn tốt cho các nhà phát triển muốn xây dựng các ứng dụng đa dạng trên nhiều nền tảng và môi trường phát triển.

## Giới thiệu về Entity Framework

Entity Framework (EF) là một khung làm việc mã nguồn mở được phát triển bởi Microsoft cho nền tảng .NET. Nó là một công cụ hỗ trợ ánh xạ đối tượng-quan hệ (ORM) cho phép phát triển ứng dụng dựa trên cơ sở dữ liệu quan hệ một cách dễ dàng và hiệu quả.

EF cung cấp một cách tiếp cận trừu tượng hóa dữ liệu, cho phép nhà phát triển làm việc với đối tượng trong mã nguồn và tự động ánh xạ chúng với cơ sở dữ liệu. Điều này giúp giảm bớt công việc lặp lại và mã code phức tạp, đồng thời tăng cường khả năng bảo trì và linh hoạt của ứng dụng.

Một số lợi ích chính của Entity Framework bao gồm:

* Tính trừu tượng hóa: EF ẩn đi chi tiết cơ sở dữ liệu và cho phép nhà phát triển tập trung vào thiết kế và phát triển đối tượng. Điều này giúp tăng tính trừu tượng và linh hoạt của mã nguồn.
* Lập trình theo mô hình Code First: EF cho phép nhà phát triển xây dựng cơ sở dữ liệu bằng cách viết mã nguồn trực tiếp, mà không cần tạo ra cơ sở dữ liệu trước. Điều này giúp tăng tốc quá trình phát triển và giảm bớt công việc quản lý phiên bản cơ sở dữ liệu.
* Tích hợp LINQ: EF tích hợp với ngôn ngữ truy vấn LINQ (Language Integrated Query), cho phép nhà phát triển truy vấn dữ liệu bằng cú pháp gần gũi và linh hoạt. Điều này giúp tạo ra mã nguồn dễ đọc, dễ bảo trì và hiệu quả.
* Quản lý thay đổi tự động: EF theo dõi và quản lý các thay đổi với cơ sở dữ liệu một cách tự động. Khi có sự thay đổi trong đối tượng, EF sẽ tự động tạo và thực thi câu truy vấn cập nhật tương ứng, giúp giảm bớt công việc lặp lại và đảm bảo tính nhất quán của cơ sở dữ liệu.
* Hỗ trợ đa dạng cơ sở dữ liệu: EF hỗ trợ nhiều hệ quản trị cơ sở dữ liệu quan hệ như SQL Server, MySQL, PostgreSQL, Oracle và SQLite. Điều này cho phép nhà phát triển lựa chọn cơ sở dữ liệu phù hợp với yêu cầu dự án.

Các phiên bản của Entity Framework:

* Entity Framework 1.0 (2008): Đây là phiên bản đầu tiên của Entity Framework, được giới thiệu cùng với .NET Framework 3.5 SP1. Phiên bản này mang lại khả năng tạo bản đồ đối tượng-quan hệ (object-relational mapping) cơ bản và hỗ trợ truy vấn LINQ to Entities.
* Entity Framework 4.0 (2010): Phiên bản này được phát hành cùng với .NET Framework 4.0 và Visual Studio 2010. EF 4.0 giới thiệu một số cải tiến quan trọng như hỗ trợ POCO (Plain Old CLR Object) và Code First Development, cho phép lập trình viên tạo các đối tượng mô hình dữ liệu mà không cần sử dụng các lớp tạo mô hình (model-first) hoặc cơ sở dữ liệu hiện có (database-first).
* Entity Framework 5.0 (2012): Phiên bản này được phát hành cùng với .NET Framework 4.5 và Visual Studio 2012. EF 5.0 tập trung vào việc cải thiện hiệu suất và hỗ trợ cho các tính năng mới như hỗ trợ cho truy vấn không đồng bộ (asynchronous querying) và truy vấn batch (batch querying).
* Entity Framework 6.0 (2013): Được phát hành cùng với .NET Framework 4.5.1 và Visual Studio 2013, EF 6.0 tập trung vào việc cải thiện hiệu suất và khả năng mở rộng, bao gồm cache thứ cấp, hỗ trợ cho các kiểu dữ liệu mới như datetime2 và enum, hỗ trợ truy vấn không đồng bộ toàn diện và nhiều tính năng khác.
* Entity Framework Core (EF Core) 1.0 (2016): Phiên bản này là phiên bản đầu tiên của Entity Framework Core, là một phiên bản tái thiết và đa nền tảng của Entity Framework. EF Core 1.0 được phát triển để chạy trên .NET Core và hỗ trợ một loạt các cơ sở dữ liệu, bao gồm cả SQL Server, SQLite, MySQL và PostgreSQL.
* Entity Framework Core (EF Core) 2.0 (2017): Phiên bản này được phát hành cùng với .NET Core 2.0 và tiếp tục cải thiện hiệu suất và tính năng của EF Core. EF Core 2.0 bổ sung hỗ trợ cho các tính năng như Tự động phát hiện thay đổi (Automatic Change Tracking), Hỗ trợ cơ sở dữ liệu quan hệ (Relational Database Support) và Hỗ trợ truy vấn không đồng bộ (Asynchronous Query Support).
* Entity Framework Core (EF Core) 3.0 (2019): Phiên bản này được phát hành cùng với .NET Core 3.0 và EF Core 3.0 tập trung vào việc cải thiện hiệu suất và khả năng mở rộng, bao gồm hỗ trợ cho CSDL không quan hệ (NoSQL), sử dụng kiểu dữ liệu JSON, hỗ trợ Index tùy chỉnh và nhiều cải tiến khác.
* Entity Framework Core (EF Core) 5.0 (2020): Phiên bản này được phát hành cùng với .NET 5.0 và EF Core 5.0 tập trung vào việc cải thithi hiệu suất và tính năng của EF Core. EF Core 5.0 bổ sung hỗ trợ cho nhiều kiểu cơ sở dữ liệu mới, bao gồm SQLite, SQL Server, MySQL, PostgreSQL, Oracle và nhiều hơn nữa. Nó cũng cung cấp nhiều cải tiến hiệu suất, bao gồm truy vấn không đồng bộ, kéo dài truy vấn và cache thứ cấp.

Entity Framework đã trở thành một công cụ phổ biến và phát triển mạnh mẽ trong việc phát triển các ứng dụng dựa trên .NET. Với khả năng giảm thiểu công việc lặp lại, tăng cường hiệu suất và linh hoạt, EF cung cấp một cách tiếp cận tiện ích và hiệu quả cho việc làm việc với cơ sở dữ liệu quan hệ trong môi trường .NET.

## Giới thiệu về cơ sở dữ liệu SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phát triển bởi Microsoft. Nó cung cấp một nền tảng mạnh mẽ và đáng tin cậy để lưu trữ, truy vấn và quản lý dữ liệu trong môi trường doanh nghiệp.

SQL Server cung cấp nhiều tính năng và lợi ích quan trọng:

* Quản lý dữ liệu: SQL Server cho phép người dùng tạo, sửa đổi và xóa cơ sở dữ liệu, bảng, chỉ mục và ràng buộc. Nó cung cấp các công cụ quản lý mạnh mẽ để giúp người quản trị cơ sở dữ liệu theo dõi và duy trì tính toàn vẹn của dữ liệu.
* Truy vấn dữ liệu: SQL Server hỗ trợ ngôn ngữ truy vấn SQL (Structured Query Language), cho phép người dùng truy vấn dữ liệu từ cơ sở dữ liệu. Với SQL Server, người dùng có thể viết các câu truy vấn phức tạp để truy xuất, lọc và sắp xếp dữ liệu một cách linh hoạt và hiệu quả.
* Bảo mật và quản lý quyền truy cập: SQL Server cung cấp các công cụ và tính năng để bảo vệ dữ liệu và quản lý quyền truy cập. Người dùng có thể thiết lập các chính sách bảo mật, quản lý người dùng, vai trò và phân quyền để kiểm soát quyền truy cập vào dữ liệu.
* Tích hợp dịch vụ: SQL Server cung cấp tích hợp với các dịch vụ và công nghệ khác của Microsoft như .NET Framework, Integration Services, Analysis Services và Reporting Services. Điều này giúp tạo ra các ứng dụng và giải pháp phức tạp hơn, từ việc xử lý dữ liệu đến phân tích và báo cáo.
* Khả năng mở rộng và hiệu suất cao: SQL Server hỗ trợ khả năng mở rộng và có thể xử lý tải công việc lớn. Nó cung cấp các tính năng như phân vùng dữ liệu, sao chép dữ liệu và nhân rộng, tối ưu hóa truy vấn và đánh chỉ mục để đảm bảo hiệu suất cao và khả năng mở rộng của hệ thống.

Các phiên bản của SQL Server:

* SQL Server 7.0 (1998): Đây là phiên bản đầu tiên của SQL Server dành cho môi trường Windows NT. Nó cung cấp nhiều tính năng quan trọng như truy vấn dựa trên ANSI SQL, xử lý ngoại lệ, giao dịch ACID và hỗ trợ dữ liệu có cấu trúc.
* SQL Server 2000: Phiên bản này được phát triển trong giai đoạn nâng cấp từ SQL Server 7.0 và được phát hành vào năm 2000. SQL Server 2000 bổ sung nhiều tính năng mới như hỗ trợ XML, dịch vụ quản lý SQL Server, dịch vụ tích hợp Full-Text Search và tính năng phân tích dữ liệu dạng OLAP (Online Analytical Processing).
* SQL Server 2005: Được phát hành vào năm 2005, SQL Server 2005 mang đến nhiều cải tiến quan trọng, bao gồm hỗ trợ CLR (Common Language Runtime) để phát triển các stored procedure và hàm bằng các ngôn ngữ lập trình .NET, tích hợp dịch vụ thông tin với tính năng truy vấn XML nâng cao, tính năng snapshot isolation, và quản lý dữ liệu dạng XML.
* SQL Server 2008: Phiên bản này được phát hành vào năm 2008 và giới thiệu một số tính năng mới như dịch vụ báo cáo (Reporting Services), dịch vụ tích hợp toàn văn (Full-Text Search), quản lý chính sách (Policy-Based Management), ghi lại thay đổi dữ liệu (Change Data Capture) và Phân tích dữ liệu dạng tabular (Tabular Data Analysis).
* SQL Server 2012: Được phát hành vào năm 2012, SQL Server 2012 bổ sung các tính năng như AlwaysOn Availability Groups (cho phép cấu hình cao hơn và khả năng chịu lỗi), các tính năng Business Intelligence (bao gồm Power View, PowerPivot và Power Query), Columnstore Indexes (tăng hiệu suất truy vấn dữ liệu phân tích), và tính năng Semantic Search.
* SQL Server 2014: Phiên bản này được phát hành vào năm 2014 và giới thiệu tính năng In-Memory OLTP (Hekaton), cải thiện hiệu suất truy vấn và xử lý giao dịch. Ngoài ra, SQL Server 2014 cũng cung cấp tính năng backup và restore trực tuyến, tích hợp Azure Cloud và tăng cường tính năng AlwaysOn.
* SQL Server 2016: Được phát hành vào năm 2016, SQL Server 2016 mang đến các tính năng như Query Store (lưu trữ và quản lý kế hoạch truy vấn), JSON support, R Services (phân tích dữ liệu thông qua R language), Always Encrypted (bảo mật dữ liệu), và Stretch Database (lưu trữ dữ liệu trên Azure).
* SQL Server 2017: Phiên bản này được phát hành vào năm 2017 và giới thiệu tính năng như hỗ trợ Linux, Graph Database (hỗ trợ lưu trữ và truy vấn dữ liệu đồ thị), Adaptive Query Processing (tối ưu hóa truy vấn), và Machine Learning Services (hỗ trợ Python và R).
* SQL Server 2019: Được phát hành vào năm 2019, SQL Server 2019 mang đến các tính năng như Big Data Clusters (kết hợp dữ liệu cấutrúc và phi cấu trúc trong một hệ thống duy nhất), hỗ trợ dự án dữ liệu lớn (data lakes), Enhanced PolyBase (truy vấn dữ liệu từ các nguồn khác nhau), hỗ trợ kịch bản Python trong stored procedure và SQL Graph (truy vấn dữ liệu đồ thị).

SQL Server đã trở thành một hệ quản trị cơ sở dữ liệu phổ biến và được sử dụng rộng rãi trong các môi trường doanh nghiệp. Với tính năng mạnh mẽ, khả năng mở rộng và tích hợp, SQL Server cung cấp một nền tảng đáng tin cậy cho việc quản lý và truy xuất dữ liệu trong các ứng dụng và hệ thống quan trọng.

# CHƯƠNG 2. KHẢO SÁT, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Khảo sát hệ thống

### Giới thiệu về hệ thống

Trong lĩnh vực thương mại điện tử ngày nay, việc mua sắm quần áo thể thao trực tuyến đã trở nên phổ biến và thuận tiện hơn bao giờ hết nhờ vào sự phát triển mạnh mẽ của các nền tảng mua sắm online. Trước đây, việc tìm kiếm và lựa chọn trang phục thể thao phù hợp thường yêu cầu khách hàng đến trực tiếp cửa hàng, tốn nhiều thời gian và công sức. Tuy nhiên, với sự bùng nổ của công nghệ số, người tiêu dùng hiện nay có thể dễ dàng lựa chọn và đặt mua các sản phẩm thời trang thể thao chỉ với vài thao tác đơn giản, ngay trên điện thoại hoặc máy tính, mang lại sự tiện lợi, linh hoạt và tiết kiệm đáng kể.

Các hệ thống quản lý bán hàng hiện đại không chỉ hỗ trợ xử lý đơn hàng nhanh chóng mà còn cung cấp các công cụ mạnh mẽ giúp các doanh nghiệp quản lý tồn kho, doanh thu, chương trình khuyến mãi và hành vi tiêu dùng của khách hàng. Điều này giúp tối ưu hóa hoạt động kinh doanh và nâng cao chất lượng dịch vụ khách hàng trong môi trường cạnh tranh gay gắt.

Trong số các nền tảng thương mại điện tử phát triển mạnh mẽ, những cái tên như Shopee, Tiki hay Zalora đã và đang ghi dấu ấn với giao diện thân thiện, chức năng tìm kiếm thông minh và nhiều ưu đãi hấp dẫn. Lấy cảm hứng từ sự thành công và mô hình hoạt động của các nền tảng này, website **Supersport** được xây dựng nhằm mang đến cho người dùng một trải nghiệm mua sắm quần áo thể thao trực tuyến toàn diện, hiện đại và tiện lợi.

Supersport không chỉ tập trung vào việc cung cấp các sản phẩm chất lượng cao mà còn hướng tới việc tối ưu hóa quy trình mua sắm – từ tìm kiếm, so sánh đến thanh toán và giao hàng. Với giao diện thân thiện, công nghệ hiện đại và tính năng quản lý hiệu quả, **Supersport** hứa hẹn sẽ trở thành một nền tảng lý tưởng cho những ai yêu thích thời trang thể thao và muốn trải nghiệm mua sắm dễ dàng, nhanh chóng và đáng tin cậy.

### Các yêu cầu của hệ thống

#### Yêu cầu chức năng

* Khách vãng lai (Người dùng chưa đăng ký tài khoản):
* Tìm kiếm sản phẩm
* Xem chi tiết sản phẩm
* Xem danh mục sản phẩm
* Đăng ký tài khoản
* Khách hàng (Người dùng đã đăng ký tài khoản):
* Đăng ký
* Đăng nhập
* Đổi mật khẩu
* Quên mật khẩu
* Tìm kiếm sản phẩm
* Xem chi tiết sản phẩm
* Thêm sản phẩm vào giỏ hàng
* Quản lý giỏ hàng
* Mua hàng và thanh toán
* Đánh giá sản phẩm
* Xem lịch sử đơn hàng
* Hủy đơn hàng
* Nhân viên
* Đăng nhập hệ thống
* Quản lý đơn hàng
* Quản lý hồ sơ cá nhân
* Xem báo cáo cá nhân
* Quản lý cửa hàng(Các khách sạn hoặc công ty đối tác):
* Đăng nhập
* Quản lý đơn hàng
* Quản lý nhân viên
* Quản lý khuyến mãi
* Xem báo cáo tổng hợp
* Người quản trị (Người quản lý hệ thống):
* Đăng nhập
* Quản lý người dùng
* Quản lý vai trò người dùng
* Quản lý danh mục sản phẩm
* Quản lý thương hiệu
* Quản lý sản phẩm

#### Yêu cầu phi chức năng

* **Yêu cầu bảo mật**

Đối với những yêu cầu về bảo mật, em sẽ tập chung vào hai hình thức chính đó là bảo mật theo sự phân quyền và bảo mật qua việc mã hóa mật khẩu khi người dùng đăng nhập.

Về phân quyền, mỗi đối tượng người dùng trong hệ thống như khách hàng, nhân viên, quản lý cửa hàng và quản trị viên sẽ có các quyền truy cập khác nhau phù hợp với vai trò của mình. Khách hàng chỉ được phép sử dụng các chức năng cơ bản như đăng ký, đăng nhập, xem sản phẩm, thêm vào giỏ hàng, mua hàng, xem lịch sử đơn hàng, đánh giá sản phẩm, đổi mật khẩu, lấy lại mật khẩu,... Nhân viên có thể đăng nhập để xem và xử lý đơn hàng, xem báo cáo cá nhân. Quản lý cửa hàng ngoài các chức năng của nhân viên còn có thể quản lý sản phẩm, chương trình khuyến mãi, hóa đơn và doanh thu. Trong khi đó, quản trị viên là người có quyền cao nhất trong hệ thống, có thể quản lý toàn bộ người dùng, phân quyền, danh mục, thương hiệu, sản phẩm, tài chính và báo cáo tổng thể của toàn hệ thống. Việc phân quyền chặt chẽ giúp ngăn ngừa tình trạng truy cập trái phép vào các chức năng không phù hợp.

Ngoài ra, hệ thống Supersport còn áp dụng phương pháp mã hóa mật khẩu người dùng trước khi lưu vào cơ sở dữ liệu. Khi người dùng thực hiện đăng nhập, mật khẩu sẽ được mã hóa và so sánh với bản mã lưu trữ thay vì so sánh trực tiếp, đảm bảo thông tin cá nhân được bảo vệ an toàn, giảm thiểu nguy cơ bị đánh cắp dữ liệu. Trên đây là những chức năng cơ bản của hệ thống. Ở những phần sau em sẽ bắt đầu đi vào thiết kế hệ thống dựa trên những yêu cầu về mặt vận hành, bảo mật và chức năng đã nêu ở phía trên.

* **Yêu cầu về hiệu năng**

Một website cơ bản không nhất thiết là phải theo quy chuẩn cơ bản. Nó có nhiều trang con được liên kết với nhau và các trang sẽ sử dụng script ví dụ như C# cần truy vấn cơ sở dữ liệu và thực hiện chức năng. Tuy nhiên, việc thiết kế và phát triển website chuyên nghiệp luôn luôn theo một khuôn khổ tiêu chuẩn, chẳng hạn như mô hình MVC (Model - View - Controller). Trong mô hình này, website được chia thành 3 lớp: lớp thấp nhất là lớp Model sẽ xử lý tất cả các truy vấn cơ sở dữ liệu. Lớp Controller thực hiện các chức năng web thông qua những tiện ích cơ sở dữ liệu trong lớp Model. Lớp View xử lý giao diện người dùng mà người xem có thể thấy, bằng cách sử dụng các chức năng được xây dựng trong lớp điều khiển. Kiểu kiến trúc này giúp trang web làm việc ổn định hơn, dễ dàng hơn để mở rộng, nâng cấp, hoặc mở rộng.

Đối với một trang web bán hàng online, lượng truy cập của khách hàng sẽ rất nhiều trong cũng một thời điểm, vì vậy hệ thống cần phải có khả năng chịu tải phù hợp, có thể xử lý các request của khách hàng một cách nhanh nhất. Kèm theo đó những tương tác của người sử dụng đối với trang web cần được phản hồi nhanh chóng, vì vậy kĩ thuật tải AJAX cho phép nội dung được tải sẵn trước khi người dùng thực sự xem nội dung đó, điều này sẽ giúp người dùng có trải nghiệm mượt mà hơn.

Tóm lại, về mặt hiệu năng của hệ thống, em sẽ được xây dựng dựa trên mô hình MVC, sử dụng tối đa khả năng của AJAX, kèm theo đó là có một thiết kế về cơ sở dữ liệu phù hợp.

## Phân tích hệ thống

### Mô hình use case

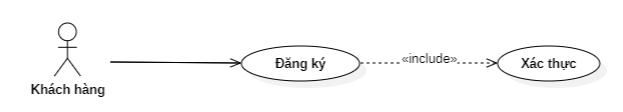
#### Biểu đồ use case



Hình 2.1. Biểu đồ use case tổng quan của hệ thống

#### Đặc tả các use case

* Use case Đăng ký

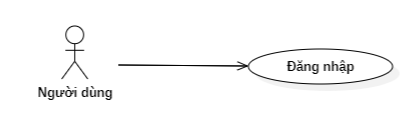


Hình 2.2. Usecase đăng ký

Bảng 2.1. Đặc tả usecase “Đăng ký”

|  |  |
| --- | --- |
| Tên usecase | Đăng ký |
| Mô tả | Usecase cho phép khách hàng đăng ký tài khoản tại website |
| Actor | Khách hàng |
| Điều kiện kích hoạt | Người dùng chọn chức năng đăng ký từ hệ thống |
| Tiền điều kiện | Người dùng chưa có tài khoản trên hệ thống |
| Hậu điều kiện | Đăng ký thành công |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Khách hàng | Chọn chức năng đăng ký | | 2 | Hệ thống | Hiển thị form đăng ký | | 3 | Khách hàng | Nhập thông tin đăng ký | | 4 | Khách hàng | Nhấn nút đăng ký | | 5 | Hệ thống | Kiểm tra tính hợp lệ và đầy đủ của các trường khách hàng nhập | | 6 | Hệ thống | Kiểm tra mật khẩu giống xác nhận mật khẩu | | 7 | Hệ thống | Kiểm tra email đã tồn tại hay chưa | | 8 | Hệ thống | Thông báo và chuyển đến trang chủ | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 5a | Hệ thống | Thông báo lỗi: “vui lòng điền đầy đủ thông tin” nếu nhập thiếu trong các trường trên | | 6a | Hệ thống | Thông báo lỗi: “mật khẩu không trùng nhau” nếu nhập mật khẩu và xác nhận mật khẩu khác nhau | | 7a | Hệ thống | Thông báo lỗi: “email đã tồn tại” nếu email này đã được đăng ký tài khoản | |

* Use case Đăng nhập

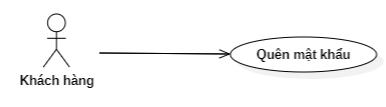


Hình 2.3: Use Đăng nhập

Bảng 2.2. Đặc tả usecase “Đăng nhập”

|  |  |
| --- | --- |
| Tên usecase | Đăng nhập |
| Mô tả | Use case này cho phép khách hàng đăng nhập vào hệ thống để mua sắm, theo dõi đơn hàng, cập nhật thông tin cá nhân. Người quản trị có thể quản lý sản phẩm, đơn hàng và các hoạt động khác trong hệ thống. |
| Actor | Khách hàng, nhân viên,quản lý, người quản trị |
| Điều kiện kích hoạt | Người dùng chọn chức năng đăng nhập từ hệ thống |
| Tiền điều kiện | Người dùng đã đăng ký tài khoản |
| Hậu điều kiện | Đăng nhập thành công |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Khách hàng, admin, quản lý, nhân viên | Chọn chức năng đăng nhập | | 2 | Hệ thống | Hiển thị form đăng nhập | | 3 | Khách hàng, admin, quản lý, nhân viên | Nhập email và mật khẩu | | 4 | Khách hàng, admin, quản lý, nhân viên | Nhấn nút đăng nhập | | 5 | Hệ thống | Kiểm tra tính hợp lệ và đầy đủ của các trường khách hàng, doanh nghiệp, quản trị viên nhập | | 6 | Hệ thống | Kiểm tra email tồn tại hay không | | 7 | Hệ thống | Chuyển đến trang chủ | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 5a | Hệ thống | Thông báo lỗi: “Vui lòng điền đầy đủ thông tin” nếu nhập email, mật khẩu hoặc cả hai | | 6a | Hệ thống | Thông báo lỗi: “email/mật khẩu không đúng” nếu không tìm thấy email hay mật khẩu trong hệ thống | |

* Use case Quên mật khẩu

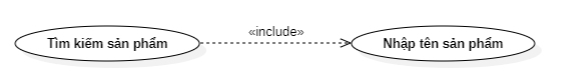


Hình 2.4. Usecase quên mật khẩu

Bảng 2.3. Đặc tả usecase “Quên mật khẩu”

|  |  |
| --- | --- |
| Tên usecase | Quên mật khẩu |
| Mô tả | Usecase cho phép khách hàng lấy lại mật khẩu khi quên |
| Actor | Khách hàng |
| Điều kiện kích hoạt | Người dùng chọn chức năng quên mật khẩu |
| Tiền điều kiện | Người dùng nhập email của rồi bấm nút quên mật khẩu , sau đó check email để nhập mật khẩu mới |
| Hậu điều kiện | Người dùng lấy lại mật khẩu thành công |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Khách hàng | Chọn chức năng quên mật khẩu | | 2 | Hệ thống | Hiển thị form quên mật khẩu | | 3 | Khách hàng | Nhập email của mình | | 4 | Hệ thống | Kiểm tra email xem hợp lệ không và gửi thông tin về cho khách hàng | | 5 | Khách hàng | Kiêm tra email của mình rồi thực hiện lấy lại mật khẩu | | 6 | Hệ thống | Kiểm tra mật khẩu giống xác nhận mật khẩu | | 7 | Hệ thống | Chuyển đến trang chủ | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 4a | Hệ thống | Thông báo lỗi: “email không tồn tại” nếu không tìm thấy email trong hệ thống | | 6a | Hệ thống | Thông báo lỗi: “Mật khẩu không trùng nhau” nếu mật khẩu khác xác nhận mật khẩu | |

* Use case Tìm kiếm sản phẩm:

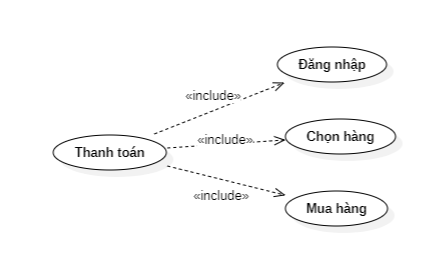


Hình 2.5. Usecase tìm kiếm sản phẩm

Bảng 2.4. Đặc tả usecase “Tìm kiếm sản phẩm”

|  |  |
| --- | --- |
| Tên usecase | Tìm kiếm sản phẩm |
| Mô tả | Usecase cho phép khách hàng tìm kiếm sản phẩm |
| Actor | Khách hàng |
| Điều kiện kích hoạt | Người dùng nhập từ khóa tìm kiếm vào ô tìm kiếm và nhấn tìm kiếm. |
| Tiền điều kiện | Người dùng đã thấy ô tìm kiếm và có thể nhập tên sản phẩm vào đó. |
| Hậu điều kiện | Hệ thống hiển thị danh sách các sản phẩm có tên phù hợp với từ khóa tìm kiếm. |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Hệ thống | Hiển thị ô tìm kiếm sản phẩm trên giao diện. | | 2 | Khách hàng | Nhập tên sản phẩm vào ô tìm kiếm và nhấn "Tìm kiếm". | | 3 | Hệ thống | |  | | --- | | Hiển thị danh sách các sản phẩm có tên phù hợp với từ khóa tìm kiếm. | | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 3a | Hệ thống | Nếu không có sản phẩm nào có tên phù hợp, hệ thống sẽ thông báo "Không tìm thấy sản phẩm nào". | |

* Use case Thanh toán:

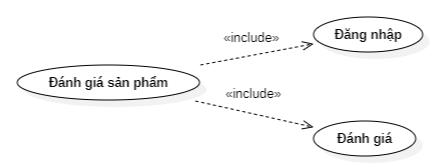


Hình 2.6. Usecase thanh toán

Bảng 2.5. Đặc tả usecase “Thanh toán”

|  |  |  |
| --- | --- | --- |
| Tên usecase | | Thanh toán |
| Mô tả | Use case cho phép khách hàng thanh toán cho các sản phẩm đã chọn trong giỏ hàng. | |
| Actor | Khách hàng | |
| Điều kiện kích hoạt | Người dùng đã chọn các sản phẩm và muốn thanh toán cho đơn hàng của mình. | |
| Tiền điều kiện | Người dùng đã đăng nhập, giỏ hàng có sản phẩm, và người dùng đã chọn phương thức thanh toán. | |
| Hậu điều kiện | Người dùng đã thanh toán thành công hoặc hệ thống thông báo lỗi nếu thanh toán không thành công. | |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Hệ thống | Hiển thị trang thanh toán với thông tin giỏ hàng, phương thức thanh toán và thông tin giao hàng. | | 2 | Khách hàng | Kiểm tra lại thông tin giỏ hàng và nhập thông tin thanh toán | | 3 | Hệ thống | Xác minh thông tin thanh toán và kiểm tra tính hợp lệ. | | 5 | Hệ thống | |  | | --- | | Thực hiện giao dịch thanh toán qua cổng thanh toán (hoặc hệ thống nội bộ). | | | 5 | Hệ thống | Thông báo kết quả thanh toán (thành công hoặc thất bại). | | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 3a | Hệ thống | Thông báo lỗi: “vui lòng điền đầy đủ thông tin” nếu nhập thiếu một trong các trường trên | | 2 | Hệ thống | |  | | --- | | Thực hiện giao dịch thanh toán qua cổng thanh toán (hoặc hệ thống nội bộ). | | | 5a | Hệ thống | Nếu thanh toán không thành công, hệ thống sẽ thông báo lỗi và yêu cầu người dùng thử lại. | | 5b | Hệ thống | Nếu thanh toán thành công, hệ thống thông báo đơn hàng đã được thanh toán và gửi xác nhận qua email hoặc thông báo trên website. | | |

* Use case Đánh giá sản phẩm:

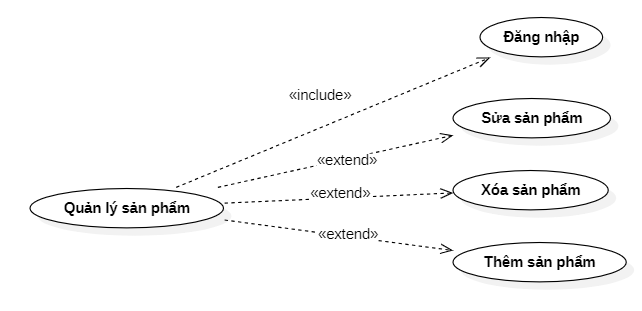


Hình 2.7. Usecase đánh giá sản phẩm

Bảng 2.6. Đặc tả usecase “Đánh sản phẩm”

|  |  |
| --- | --- |
| Tên usecase | Đánh giá sản phẩm |
| Mô tả | Usecase cho phép khách hàng đánh giá sản phẩm sau khi mua hoặc sử dụng. |
| Actor | Khách hàng. |
| Điều kiện kích hoạt | Khách hàng đã mua sản phẩm và muốn để lại đánh giá. |
| Tiền điều kiện | Khách hàng phải đăng nhập và có quyền truy cập vào sản phẩm đã mua. |
| Hậu điều kiện | Sản phẩm sẽ nhận được một đánh giá mới, bao gồm điểm số và nhận xét. |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Hệ thống | Hiển thị danh sách các sản phẩm đã mua của khách hàng trong trang "Lịch sử mua hàng". | | 2 | Khách hàng | Chọn sản phẩm cần đánh giá. | | 3 | Hệ thống | Hiển thị form đánh giá sản phẩm (điểm số, nhận xét). | | 4 | Khách hàng | Chọn điểm đánh giá và nhập thông tin đánh giá | | 5 | Hệ thống | Xác minh và lưu đánh giá của khách hàng. | | 6 | Hệ thống | Hiển thị thông báo xác nhận rằng đánh giá đã được lưu thành công. | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 5a | Hệ thống | Nếu thông tin đánh giá không hợp lệ ,hệ thống thông báo lỗi và yêu cầu khách hàng chỉnh sửa. | |

* Use case Quản lý sản phẩm:

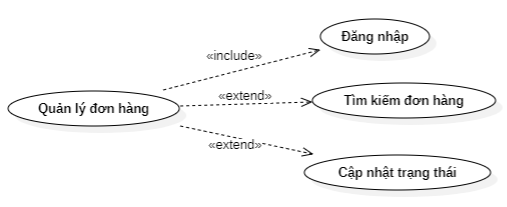


Hình 2.8. Usecase quản lý sản phẩm

Bảng 2.7. Đặc tả usecase “Quản lý sản phẩm”

|  |  |
| --- | --- |
| Tên usecase | Quản lý sản phẩm |
| Mô tả | Usecase cho phép người quản trị thêm, sửa, xóa sản phẩm trong kho hàng. |
| Actor | Người quản trị |
| Điều kiện kích hoạt | Người quản trị đăng nhập vào hệ thống và truy cập vào phần quản lý sản phẩm. |
| Tiền điều kiện | Người quản trị đã có quyền truy cập vào chức năng quản lý sản phẩm. |
| Hậu điều kiện | Sản phẩm được thêm, sửa, xóa thành công và hệ thống cập nhật danh sách sản phẩm. |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Người quản trị | Đăng nhập vào hệ thống và truy cập vào chức năng "Quản lý sản phẩm". | | 2 | Hệ thống | Hiển thị danh sách các sản phẩm hiện có trong kho hàng. | | 3 | Người quản trị | Chọn thao tác cần thực hiện: thêm sản phẩm mới, sửa thông tin sản phẩm, hoặc xóa sản phẩm. | | 4 | Hệ thống | Nếu thêm hoặc sửa sản phẩm, hệ thống yêu cầu quản trị viên nhập thông tin sản phẩm | | 5 | Người quản trị | Nhập thông tin cần thiết cho sản phẩm mới hoặc sản phẩm cần sửa. | | 6 | Hệ thống | Cập nhật thông tin sản phẩm vào cơ sở dữ liệu hoặc xóa sản phẩm khỏi hệ thống. | | 7 | Hệ thống | Hiển thị thông báo xác nhận hành động (thêm, sửa, xóa thành công). | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 4a | Hệ thống | Nếu thông tin nhập vào không hợp lệ, hệ thống thông báo lỗi và yêu cầu nhập lại. | | 6a | Hệ thống | Nếu có lỗi trong quá trình lưu hoặc xóa sản phẩm, hệ thống thông báo lỗi và yêu cầu thử lại. | |

* Use case Quản lý đơn hàng

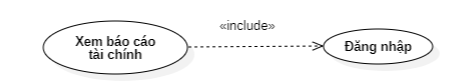


Hình 2.9. Usecase quản lý đơn hàng

Bảng 2.8. Đặc tả usecase “Quản lý đơn hàng”

|  |  |
| --- | --- |
| Tên usecase | Quản lý đơn hàng |
| Mô tả | Use case cho phép quản lý cửa hàng và nhân viên quản lý các đơn hàng, bao gồm xem, thay đổi trạng thái và xử lý các yêu cầu liên quan đến đơn hàng. |
| Actor | Quản lý cửa hàng, Nhân viên |
| Điều kiện kích hoạt | Đăng nhập vào hệ thống và truy cập vào phần quản lý đơn hàng. |
| Tiền điều kiện | Người dùng đã có quyền truy cập vào chức năng quản lý đơn hàng. |
| Hậu điều kiện | Trạng thái đơn hàng được cập nhật và người dùng xử lý đơn hàng thành công. |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Quản trị viên | Đăng nhập vào hệ thống và truy cập vào chức năng "Quản lý đơn hàng". | | 2 | Hệ thống | Hiển thị danh sách các đơn hàng, bao gồm trạng thái của mỗi đơn hàng. | | 3 | Quản trị viên | Chọn đơn hàng cần xem hoặc thay đổi trạng thái. | | 4 | Quản trị viên | Thay đổi trạng thái đơn hàng . | | 5 | Hệ thống | |  | | --- | | Cập nhật trạng thái đơn hàng trong cơ sở dữ liệu và thông báo cho khách hàng nếu cần. | | | 6 | Hệ thống | |  | | --- | | Cập nhật trạng thái đơn hàng trong cơ sở dữ liệu và thông báo cho khách hàng nếu cần. | | | 7 | Hệ thống | Hiển thị thông báo xác nhận thay đổi trạng thái thành công. | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 5a | Hệ thống | Nếu có lỗi trong quá trình thay đổi trạng thái, hệ thống thông báo lỗi và yêu cầu thử lại. | | 6a | Hệ thống | Nếu đơn hàng bị hủy, hệ thống thông báo cho quản lý cửa hàng, nhân viên và khách hàng về lý do hủy bỏ hoặc vấn đề phát sinh. | |

* Use case Xem báo cáo tài chính:

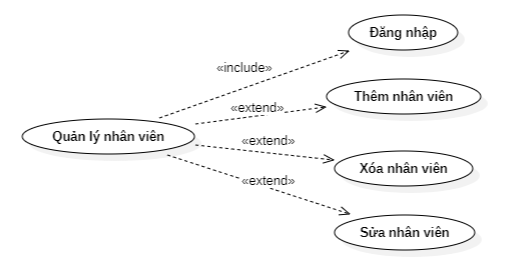
**

Hình 2.10. Usecase xem báo cáo tài chính

Bảng 2.9. Đặc tả usecase “Xem báo cáo tài chính”

|  |  |
| --- | --- |
| Tên usecase | Xem báo cáo tài chính |
| Mô tả | Use case cho phép quản lý cửa hàng xem các báo cáo tài chính của cửa hàng, bao gồm doanh thu, chi phí, lợi nhuận, và các báo cáo tài chính khác. |
| Actor | Quản lý cửa hàng |
| Điều kiện kích hoạt | Quản lý cửa hàng đăng nhập vào hệ thống và truy cập vào phần báo cáo tài chính. |
| Tiền điều kiện | Quản lý cửa hàng có quyền truy cập vào báo cáo tài chính. |
| Hậu điều kiện | Quản lý cửa hàng xem được báo cáo tài chính thành công. |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Quản lý cửa hàng | Truy cập vào phần "Báo cáo tài chính". | | 2 | Hệ thống | Hiển thị giao diện báo cáo tài chính với các tùy chọn thời gian | | 3 | Quản lý cửa hàng | Chọn loại báo cáo tài chính cần xem (doanh thu, chi phí, lợi nhuận, v.v.) và chọn khoảng thời gian (tháng, quý, năm). | | 4 | Hệ thống | Hiển thị báo cáo tài chính với dữ liệu được tổng hợp và tính toán theo yêu cầu. | | 5 | Quản lý cửa hàng | Xem chi tiết báo cáo tài chính và các số liệu liên quan. | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 3a | Hệ thống | Nếu dữ liệu không có sẵn hoặc có lỗi trong quá trình tính toán, hệ thống thông báo lỗi và yêu cầu thử lại. | | 5a | Hệ thống | Quản lý cửa hàng có thể tải báo cáo dưới dạng file (PDF, Excel) nếu cần. | |

* Use case Quản lý nhân viên

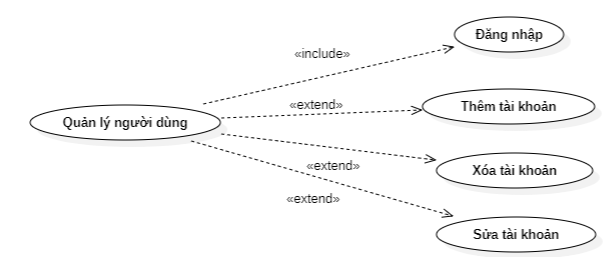
**

Hình 2.11. Usecase quản lý nhân viên

Bảng 2.10. Đặc tả usecase “Quản lý nhân viên”

|  |  |
| --- | --- |
| Tên usecase | Quản lý nhân viên |
| Mô tả | Use case cho phép quản lý cửa hàng quản lý thông tin nhân viên, bao gồm thêm mới, chỉnh sửa thông tin, xóa nhân viên. |
| Actor | Quản lý cửa hàng |
| Điều kiện kích hoạt | Quản lý cửa hàng đăng nhập vào hệ thống và truy cập vào phần quản lý nhân viên. |
| Tiền điều kiện | Quản lý cửa hàng có quyền truy cập và thao tác trên phần quản lý nhân viên. |
| Hậu điều kiện | Quản lý cửa hàng có thể xem và cập nhật thông tin nhân viên thành công. |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Quản lý cửa hàng | Chọn chức năng thêm mới hoặc cập nhật | | 2 | Hệ thống | Hiển thị danh sách các nhân viên hiện tại trong hệ thống. | | 3 | Quản lý cửa hàng | Chọn thao tác với một nhân viên (xem chi tiết, thêm mới, chỉnh sửa, xóa). | | 4 | Hệ thống | |  | | --- | | Cung cấp giao diện để thực hiện thao tác (thêm mới, chỉnh sửa thông tin nhân viên, hoặc xóa nhân viên). | | | 5 | Quản lý cửa hàng | Cập nhật thông tin hoặc thực hiện hành động xóa nhân viên (nếu có). | | 6 | Hệ thống | Cập nhật thông tin nhân viên hoặc xóa nhân viên khỏi hệ thống và hiển thị thông báo thành công. | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 4a | Hệ thống | Nếu có lỗi trong quá trình cập nhật hoặc xóa nhân viên (ví dụ: thông tin không hợp lệ), hệ thống thông báo lỗi và yêu cầu thử lại. | |

* Use case Quản lý người dùng

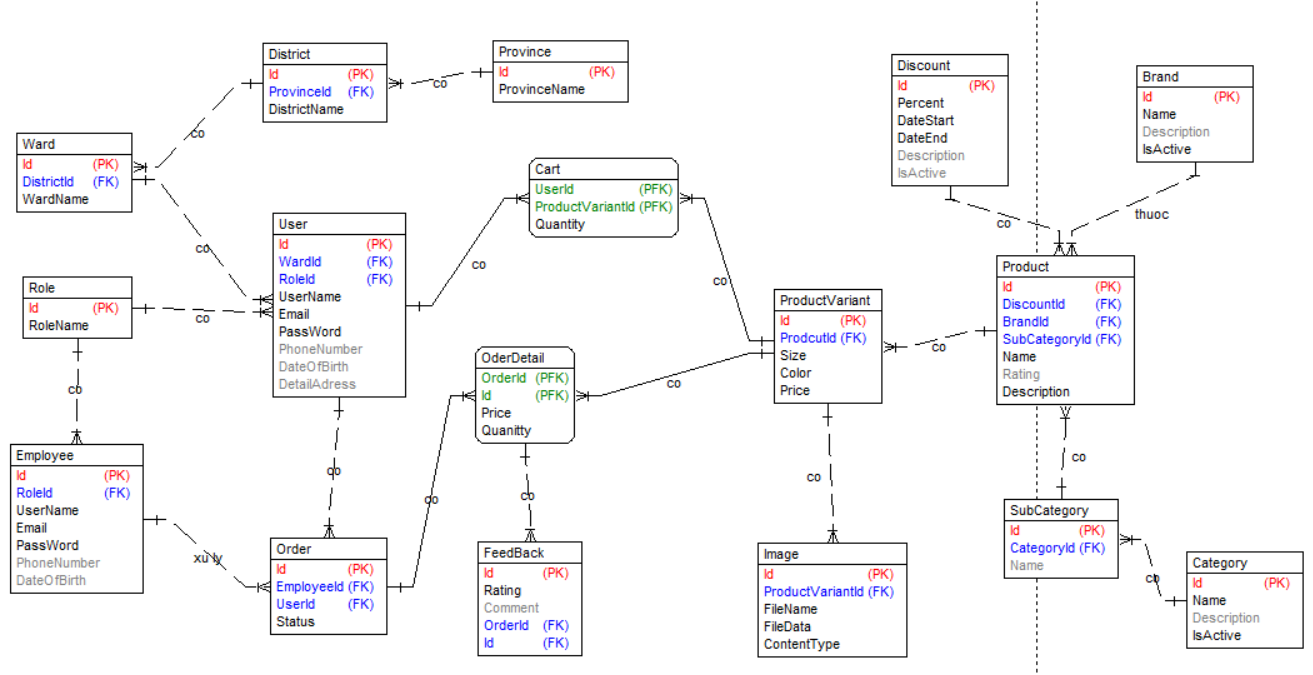
**

Hình 2.12. Usecase quản lý người dùng

Bảng 2.11. Đặc tả usecase “Quản lý người dùng”

|  |  |  |
| --- | --- | --- |
| Tên usecase | |  | | --- | | Quản lý người dùng | |
| Mô tả | Use case cho phép quản trị viên quản lý thông tin người dùng trong hệ thống, bao gồm thêm mới, chỉnh sửa, xóa tài khoản, cấp quyền truy cập và thay đổi mật khẩu. |
| Actor | Quản trị viên |
| Điều kiện kích hoạt | Quản trị viên đăng nhập vào hệ thống và truy cập vào phần quản lý người dùng. |
| Tiền điều kiện | Quản trị viên có quyền truy cập và thao tác trên phần quản lý người dùng. |
| Hậu điều kiện | Quản trị viên có thể quản lý thông tin người dùng thành công. |
| Luồng cơ bản | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 1 | Người quản trị | Đăng nhập vào hệ thống và truy cập vào phần "Quản lý người dùng". | | 2 | Hệ thống | Hiển thị danh sách các người dùng trong hệ thống. | | 3 | Người quản trị | Chọn thao tác với một người dùng (xem chi tiết, thêm mới, chỉnh sửa, xóa, thay đổi mật khẩu). | | 4 | Người quản trị | Cung cấp giao diện để thực hiện thao tác (thêm mới, chỉnh sửa thông tin người dùng, thay đổi mật khẩu, xóa người dùng). | | 5 | |  | | --- | | Người quản trị | | Cập nhật thông tin hoặc thực hiện hành động xóa người dùng (nếu có). | | 6 | Hệ thống | Cập nhật thông tin người dùng, thay đổi mật khẩu hoặc xóa người dùng khỏi hệ thống và hiển thị thông báo thành công. | |
| Luồng rẽ nhánh | |  |  |  | | --- | --- | --- | | STT | Tác nhân | Hành động | | 5a | Hệ thống | Nếu có lỗi trong quá trình cập nhật hoặc xóa người dùng (ví dụ: thông tin không hợp lệ), hệ thống thông báo lỗi và yêu cầu thử lại. | |

### Mô hình dữ liệu



Hình 2.13. Mô hình dữ liệu

### Phân tích use case

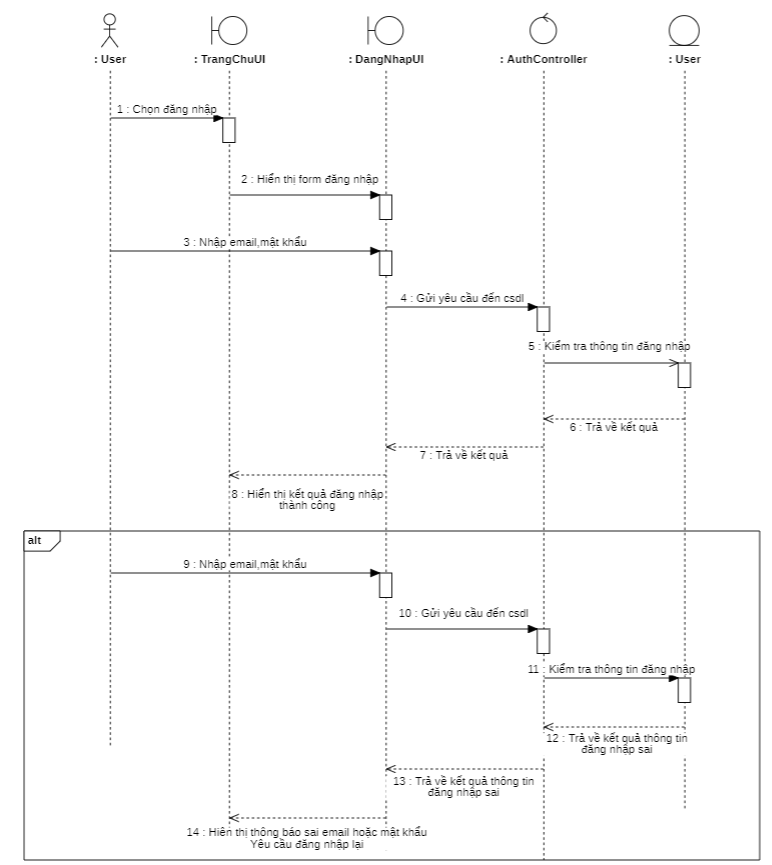
#### Đăng nhập

Sử dụng cho ca đăng nhập: Người dùng sử dụng tải khoản và mật khẩu để đăng nhập vào hệ thống

* VOPC của use case use case:

Hình 2.14. Sơ đồ VOPC use case đăng nhập

* Sơ đồ trình tự:



Hình 2.15. Sơ đồ trình tự use case “Đăng nhập”

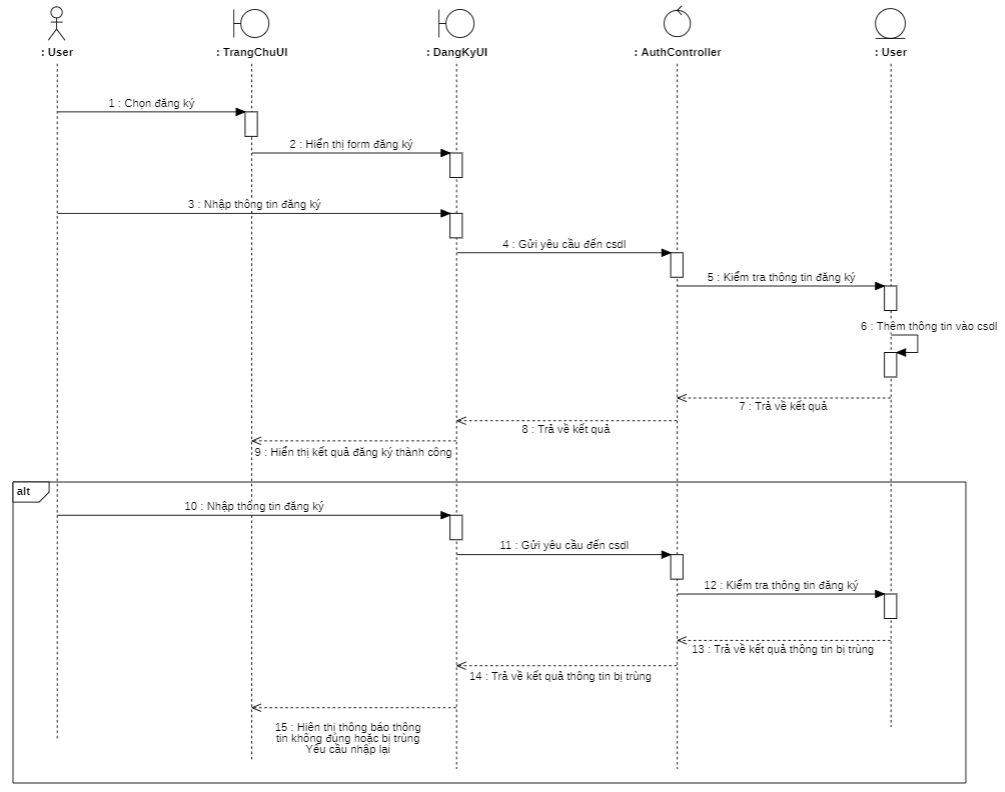
#### Đăng ký

Sử dụng cho ca đăng ký: Người dùng mới tạo tài khoản để có quyền truy cập vào hệ thống.

* Sơ đồ VOPC tham gia use case:

Hình 2.16. Sơ đồ VOPC use case đăng ký

* Sơ đồ trình tự của use case:



Hình 2.17. Sơ đồ trình tự use case “Đăng ký”

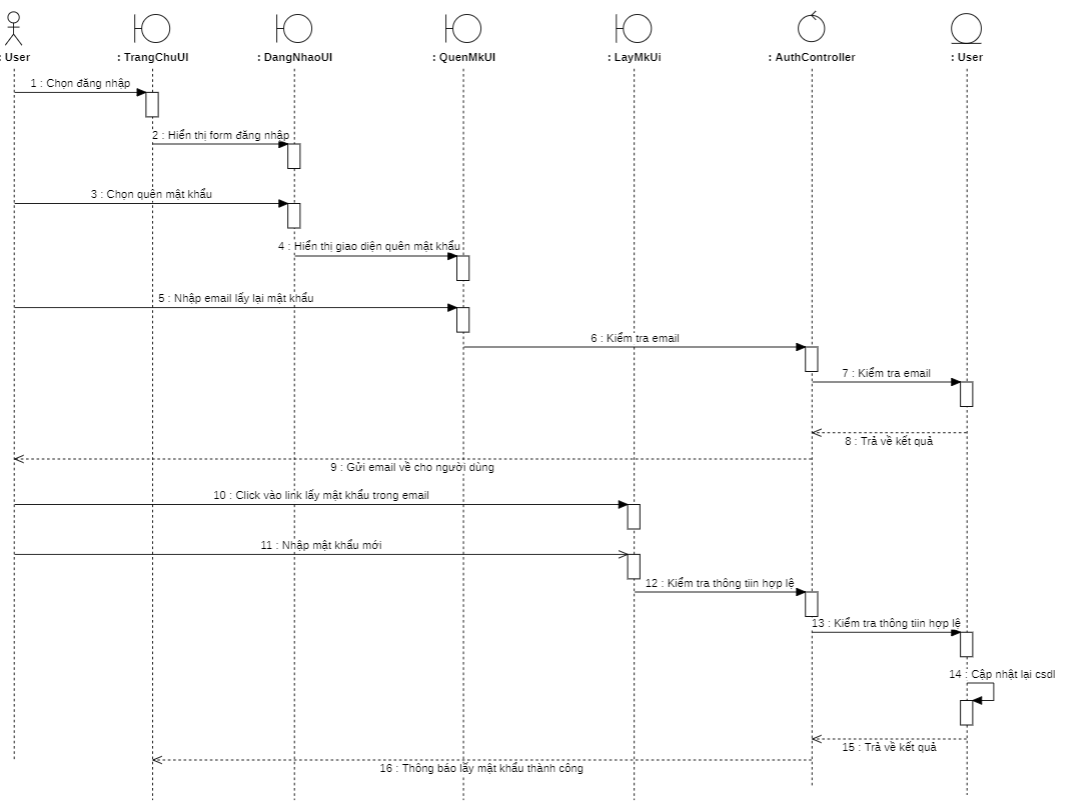
#### Quên mật khẩu

Sử dụng cho ca quản lý người dùng: Người dùng quên mật khẩu và yêu cầu hệ thống gửi một liên kết hoặc mã xác nhận để đặt lại mật khẩu.

* Sơ đồ VOPC tham gia use case:

Hình 2.18. Sơ đồ VOPC use case quên mật khẩu

* Sơ đồ trình tự của use case



Hình 2.19. Sơ đồ tuần tự use case “Quên mật khẩu”

#### Tìm kiếm sản phẩm

Sử dụng cho ca đặt phòng: Người dùng tìm kiếm sản phẩm

* Sơ đồ VOPC tham gia use case:

Hình 2.20. Sơ đồ VOPC use case Tìm kiếm sản phẩm

* Sơ đồ trình tự của use case:

Hình 2.21. Biểu đồ trình tự use case “Tìm kiếm sản phẩm”

#### Thanh toán

Sử dụng cho ca thanh toán: Người dùng thanh toán sau khi mua sản phẩm. Sơ đồ VOPC tham gia use case:

Hình 2.22. Sơ đồ VOPC use case “Thanh toán”

* Sơ đồ trình tự của use case:

Hình 2.23. Sơ đồ trình tự use case “Thanh toán”

#### Thêm sản phẩm

Sử dụng cho thêm khách sạn: Người dùng thêm sản phẩm vào hệ thống

* Sơ đồ VOPC tham gia use case:

Hình 2.24. Sơ đồ VOPC use case thêm sản phẩm

* Sơ đồ trình tự của use case:

Hình 2.25. Sơ đồ trình tự use case “Thêm sản phẩm”

#### Cập nhật sản phẩm

Sử dụng cho ca quản lý khách sạn: Người dùng cập nhật sản phẩm

* Sơ đồ VOPC tham gia use case:

Hình 2.26. Sơ đồ VOPC tham gia use case cập nhật sản phẩm

* Sơ đồ trình tự của use case:

Hình 2.27. Sơ đồ trình tự use case “Cập nhật sản phẩm”

#### Xóa sản phẩm

Sử dụng cho ca quản lý khách sạn: Ngưởi quản trị xóa sản phẩm

* Sơ đồ VOPC tham gia use case:

Hình 2.28. Sơ đồ VOPC tham gia use case xóa sản phẩm

* Sơ đồ trình tự của use case:

Hình 2.29. Sơ đồ trình tự use case “Xóa sản phẩm”

#### Đánh giá sản phẩm

Sử dụng cho ca quản lý khách sạn: Người dùng đánh giá sản phẩm

Sơ đồ VOPC tham gia use case:

Hình 2.30. Sơ đồ VOPC use case đánh giá sản phẩm

* Sơ đồ trình tự của use case:

Hình 2.31. Sơ đồ trình tự use case “Đánh giá sản phẩm”

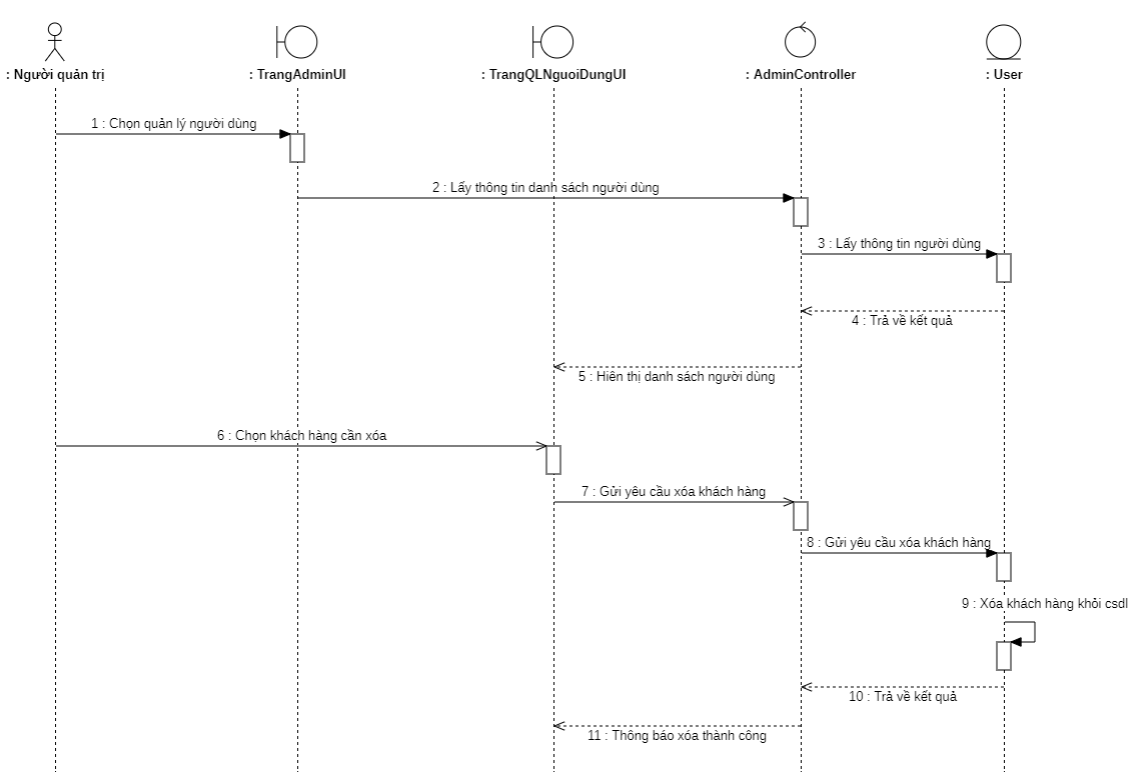
#### Xóa người dùng

Sử dụng cho ca quản lý người dùng: Quản trị viên xóa thông tin tài khoản của người dùng ra khỏi hệ thống.

* Sơ đồ lớp tham gia hệ thống

Hình 2.32. Sơ đồ VOPC use case xóa người dùng

* Sơ đồ trình tự của use case:



Hình 2.33. Sơ đồ trình tự use case “Xóa người dùng”

#### Cập nhật thuộc tính sản phẩm

Sử dụng cho ca quản lý hóa đơn: Quản trị viên cập nhật thuộc tính sản phẩm

* Sơ đồ lớp tham gia use case:

Hình 2.34. Sơ đồ VOPC use case cập nhật thuộc tính sản phẩm

* Sơ đồ trình tự của use case:

Hình 2.35. Sơ đồ trình tự use case “Cập nhật thuộc tính sản phẩm”

## Thiết kế hệ thống

### Thiết kế cơ sở dữ liệu



Hình 2.36. Thiết kế cơ sở dữ liệu

**Role**: Lưu thông tin về các vai trò trong hệ thống, ví dụ như "Admin", "Nhân viên", "Khách hàng". Mỗi vai trò có một mã và tên phân quyền riêng biệt.

**User**: Lưu thông tin về người dùng của hệ thống, bao gồm tên đăng nhập, email, mật khẩu, số điện thoại, ngày sinh, và địa chỉ chi tiết. Mỗi người dùng có một vai trò được xác định qua RoleId.

**Employee**: Lưu thông tin về nhân viên của cửa hàng, bao gồm tên, email, mật khẩu, số điện thoại, ngày sinh. Mỗi nhân viên có một vai trò và thuộc một nhóm công việc nhất định.

**Category**: Lưu thông tin về các danh mục sản phẩm, giúp phân loại các sản phẩm trong cửa hàng.

**Subcategory**: Lưu thông tin về các danh mục con sản phẩm, giúp phân loại các sản phẩm trong cửa hàng.

**Brand**: Lưu thông tin về các thương hiệu sản phẩm, giúp xác định xuất xứ hoặc thương hiệu của các sản phẩm.

**Product**: Lưu thông tin về sản phẩm, bao gồm tên sản phẩm, giá cả, mô tả, và hình ảnh. Mỗi sản phẩm thuộc một thương hiệu và một danh mục cụ thể.

**ProductVariant**: Lưu thông tin về các biến thể của sản phẩm (ví dụ: màu sắc, kích thước), cùng với số lượng hiện có của từng biến thể.

**FeedBack**: Lưu thông tin phản hồi từ khách hàng về sản phẩm, bao gồm đánh giá sao, bình luận, và liên kết với đơn hàng và biến thể sản phẩm.

**Cart**: Lưu thông tin giỏ hàng của người dùng, bao gồm sản phẩm đã chọn, số lượng và người sở hữu giỏ hàng.

**Image**: Lưu trữ các hình ảnh bổ sung cho sản phẩm, có thể là các góc chụp khác nhau hoặc ảnh chi tiết của sản phẩm.

**Order**: Lưu thông tin về đơn hàng của người dùng, bao gồm người dùng, nhân viên xử lý đơn hàng và trạng thái đơn hàng.

**OrderDetail**: Lưu chi tiết của từng đơn hàng, bao gồm sản phẩm và biến thể sản phẩm, giá và số lượng trong đơn hàng.

**Discount**: Lưu thông tin về các chương trình khuyến mãi, bao gồm tỷ lệ giảm giá và thời gian áp dụng.

**Ward**: Lưu thông tin về các phường/xã, dùng để phân loại địa chỉ của người dùng.

**District**: Lưu thông tin về các quận/huyện, giúp phân loại địa chỉ của các phường/xã.

**Province**: Lưu thông tin về các tỉnh/thành phố, là cấp cao nhất trong phân loại địa chỉ.

**Mối quan hệ giữa các bảng:**

Mỗi vai trò có thể thuộc về một hoặc nhiều nhân viên, nhưng mỗi nhân viên chỉ có một vai trò duy nhất.

Mỗi vai trò có thể thuộc về một hoặc nhiều người dùng, nhưng mỗi người dùng chỉ có một vai trò duy nhất.

Mỗi người dùng sẽ thuộc về một phường/xã. Mối quan hệ giữa User và Ward là "nhiều-một".

Mỗi quận/huyện có thể có nhiều phường/xã, nhưng mỗi phường/xã chỉ thuộc về một quận/huyện.

Mỗi tỉnh/thành phố có thể có nhiều quận/huyện, nhưng mỗi quận/huyện chỉ thuộc về một tỉnh/thành phố.

Mỗi sản phẩm có thể có một chương trình giảm giá duy nhất. Một chương trình giảm giá có thể áp dụng cho nhiều sản phẩm.

Mỗi sản phẩm có thể có nhiều biến thể (màu sắc, kích thước, số lượng), nhưng mỗi biến thể chỉ thuộc về một sản phẩm.

Mỗi sản phẩm có thể có nhiều hình ảnh trong thư viện. Một hình ảnh chỉ thuộc về một sản phẩm.

Mỗi giỏ hàng sẽ thuộc về một người dùng, nhưng mỗi người dùng có thể có nhiều giỏ hàng.

Mỗi đơn hàng chỉ thuộc về một người dùng, nhưng một người dùng có thể có nhiều đơn hàng.

Mỗi đơn hàng được xử lý bởi một nhân viên. Một nhân viên có thể xử lý nhiều đơn hàng.

Một chi tiết đơn hàng sẽ liên kết với một biến thể sản phẩm. Một biến thể sản phẩm có thể có trong nhiều chi tiết đơn hàng.

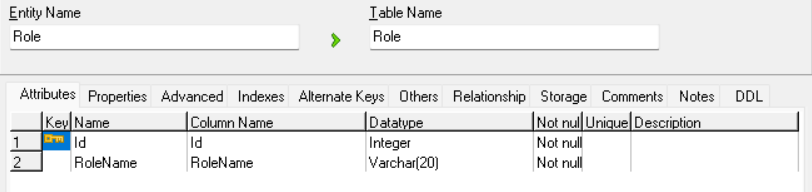
Mỗi chi tiết đơn hàng chỉ thuộc về một đơn hàng, nhưng mỗi đơn hàng có thể có nhiều chi tiết đơn hàng.

Mỗi phản hồi sẽ liên kết với một đơn hàng. Một đơn hàng có thể có nhiều phản hồi.

Mỗi phản hồi sẽ liên kết với một biến thể sản phẩm. Một biến thể sản phẩm có thể nhận nhiều phản hồi.

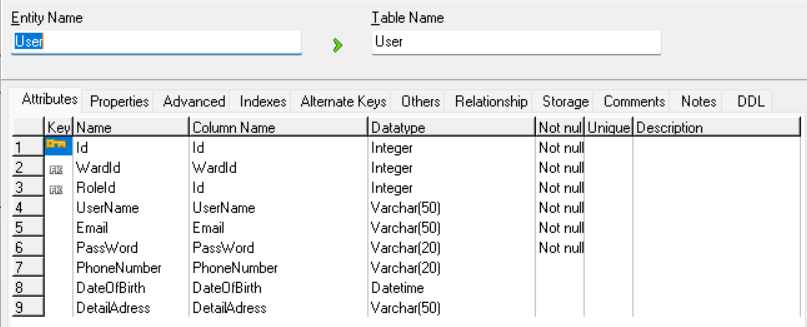
Mỗi sản phẩm có một hoặc nhiều ảnh.Mỗi ảnh chỉ thuộc về một sản phẩm.

#### Bảng dữ liệu “Role”



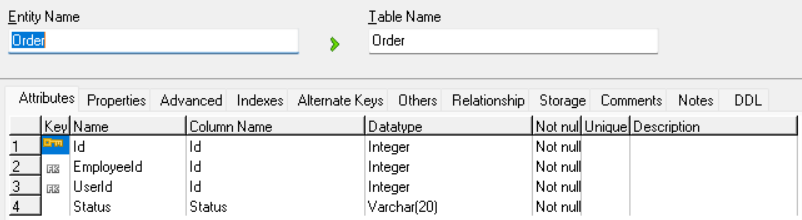
Bảng 2.12. Mô tả bảng “Role”

#### Bảng dữ liệu “User”



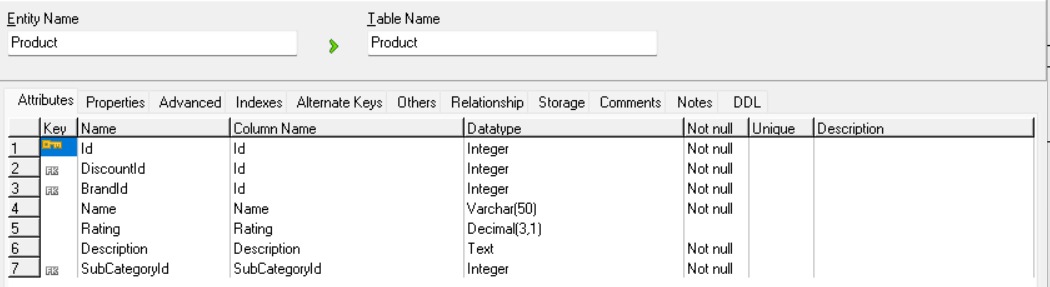
Bảng 2.13. Mô tả bảng “User”

#### Bảng dữ liệu “Order”



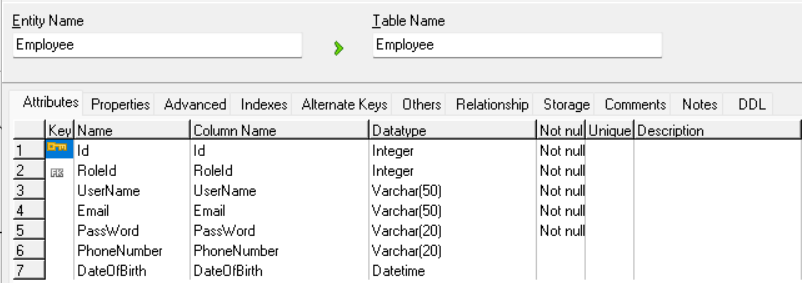
Bảng 2.14. Mô tả bảng “Order”

#### Bảng dữ liệu “Product”



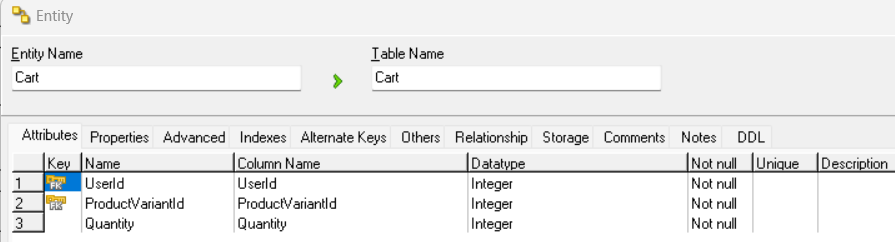
Bảng 2.15. Mô tả bảng “Product”

#### Bảng dữ liệu “Employee”



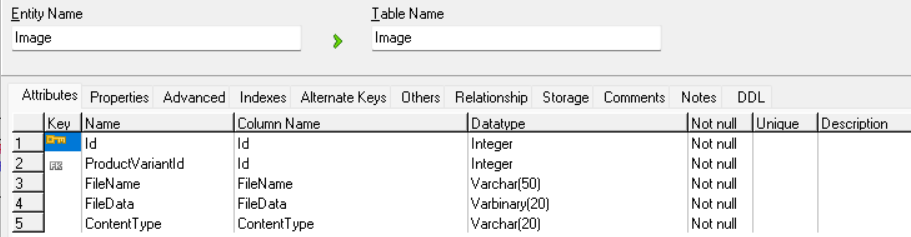
Bảng 2.16. Mô tả bảng “Employee”

#### Bảng dữ liệu “Cart”



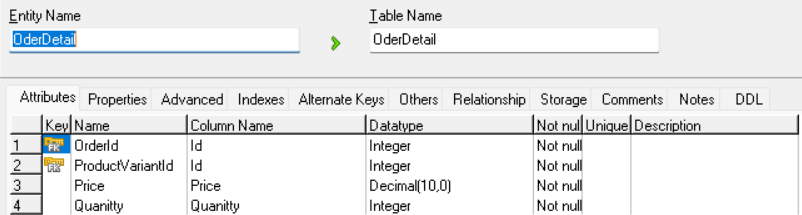
Bảng 2.17. Mô tả bảng “Cart”

#### Bảng dữ liệu “Image”



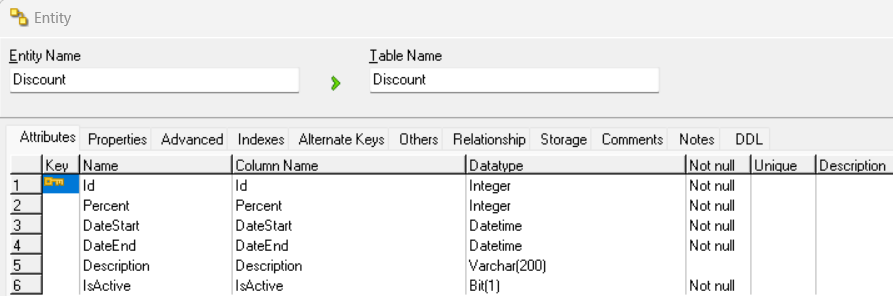
Bảng 2.18. Mô tả bảng “Image”

#### Bảng dữ liệu “OrderDetail”

****

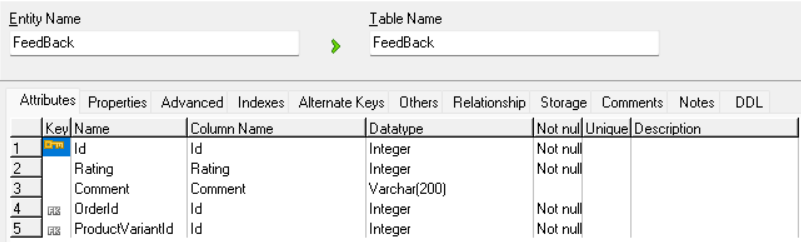
Bảng 2.19. Mô tả bảng “OrderDetail”

#### Bảng dữ liệu “Discount”



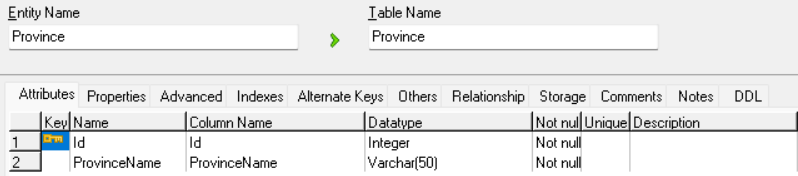
Bảng 2.20. Mô tả bảng “Discount”

#### Bảng dữ liệu “Feedback”



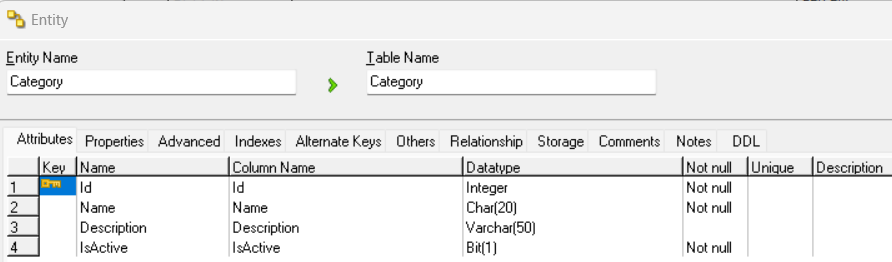
Bảng 2.21. Mô tả bảng “Feedback”

#### Bảng dữ liệu “Province”



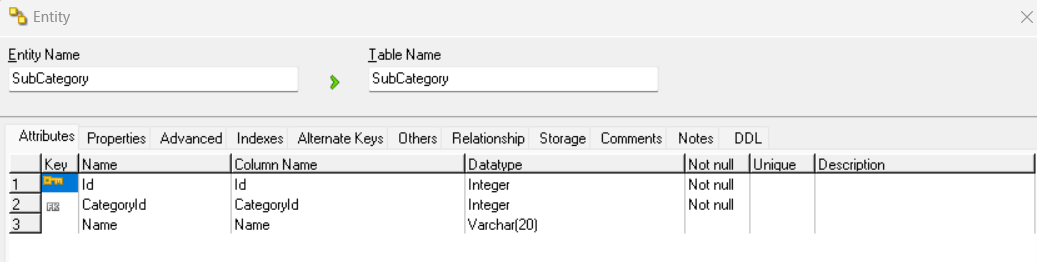
Bảng 2.22. Mô tả bảng “Province”

#### Bảng dữ liệu “Category”



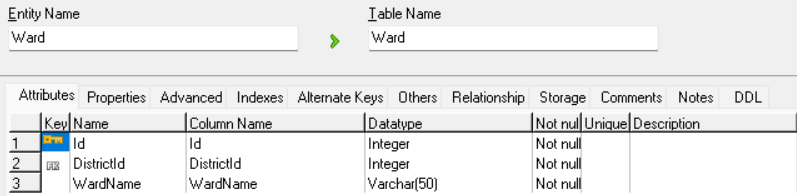
Bảng 2.23. Mô tả bảng “Category”

#### Bảng dữ liệu “SubCategory”



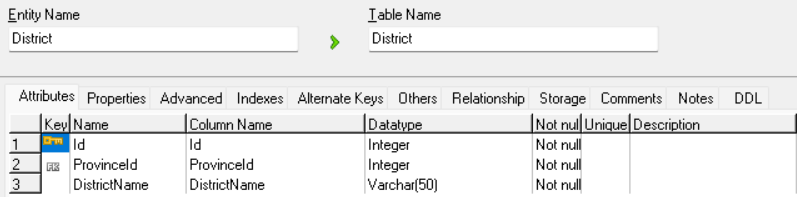
Bảng 2.24. Mô tả bảng “SubCategory”

#### Bảng dữ liệu “Ward”



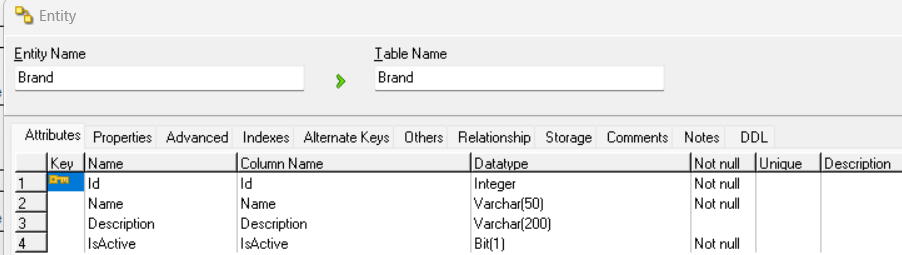
Bảng 2.25. Mô tả bảng “Ward”

#### Bảng dữ liệu “District”



Bảng 2.26. Mô tả bảng “District”

#### Bảng dữ liệu “Brand”



Bảng 2.27. Mô tả bảng “Brand”

### Thiết kế giao diện

# CHƯƠNG 3. CÀI ĐẶT VÀ KẾT QUẢ

## Cài đặt hệ thống

### Yêu cầu cấu hình

* Hệ điều hành (OS):
* Windows 10 (Phiên bản 1909 trở lên) hoặc Windows 11.
* Windows Server 2016, 2019, hoặc 2022.
* CPU:
* Tối thiểu: Bộ vi xử lý lõi kép 1.8 GHz hoặc nhanh hơn.
* Khuyến nghị: Bộ vi xử lý lõi tứ hoặc cao hơn (tốt nhất là từ Intel Core i5, i7 hoặc tương đương).
* RAM:
* Tối thiểu: 4 GB RAM (ít nhất).
* Khuyến nghị: 8 GB RAM trở lên để chạy mượt mà hơn, đặc biệt nếu làm việc với các dự án lớn, nhiều ngôn ngữ lập trình hoặc chạy các máy ảo.
* Ổ cứng:
* Tối thiểu: 850 MB dung lượng ổ cứng để cài đặt Visual Studio tối thiểu.
* Khuyến nghị: 20-50 GB dung lượng trống nếu cài đặt đầy đủ các thành phần (các bộ SDK, framework).
* Sử dụng ổ SSD để cải thiện tốc độ tải và chạy ứng dụng.
* Trình duyệt: Visual Studio yêu cầu một trình duyệt hiện đại, chẳng hạn như Microsoft Edge hoặc Google Chrome, để sử dụng các tính năng web và công cụ liên quan đến phát triển ứng dụng web.
* Kết nối Internet
* .NET Framework 4.8 hoặc cao hơn (Visual Studio sẽ tự động cài đặt nếu chưa có).

### Hướng dẫn cài đặt

Bước 1: Khởi động Visual Studio

Bước 2: Mở project có sẵn

* Từ giao diện Visual Studio, chọn File → Open → Project/Solution.

Duyệt đến thư mục chứa project web của bạn.

* Chọn file .sln (solution) hoặc file .csproj nếu không có file solution, rồi nhấn Open.

Bước 3: Khôi phục các gói NuGet (nếu cần)

Sau khi mở project, bạn cần khôi phục các gói NuGet nếu chúng chưa được cài đặt.

Click chuột phải vào solution trong Solution Explorer.

Chọn Restore NuGet Packages.

Visual Studio sẽ tự động tải về và cài đặt các gói cần thiết.

Bước 4: Kiểm tra cấu hình môi trường

Kiểm tra file launchSettings.json (nằm trong thư mục Properties).

Đảm bảo cấu hình đúng URL để chạy trên IIS Express hoặc Kestrel.

Bước 6: Chạy project

Nhấn nút IIS Express hoặc nút Run (biểu tượng hình tam giác màu xanh) trên thanh công cụ của Visual Studio.

Nếu bạn chọn IIS Express, địa chỉ sẽ có dạng <http://localhost:xxxx> (với xxxx là một port số ngẫu nhiên).

Trình duyệt sẽ mở ra và chạy ứng dụng web của bạn.

* Tài khoản Admin:

Tài khoản: [admin@gmail.com](mailto:admin@gmail.com)

Mật khẩu: admin

## Kết quả

### Giao diện người dùng

### Giao diện quản trị

## Kiểm thử

### Kế hoạch kiểm thử

Mục đích: Quá trình kiểm thử nhằm mang lại mục đích sau:

* Xác định những thông tin cơ bản về dự án và các thành phần, chức năng được kiểm thử.
* Liệt kê những yêu cầu cho kiểm thử.
* Những chiến lược kiểm thử nên được sử dụng
* Ước lượng những yêu cầu về tài nguyên và chi phí cho việc kiểm thử

### Lịch trình kiểm thử

# KẾT LUẬN

Đề tài **“Xây dựng website bán hàng thể thao SuperSports”** xuất phát từ những nhu cầu thực tế mà ngày nay từ các du khách trong và ngoài nước. Những gợi ý về khách sạn, giá phòng đã được đề xuất đến khách hàng nhằm giúp các du khách rút ngắn thời gian trong việc tìm hiểu, lên kế hoạch và chuẩn bị cho những chuyến du lịch, nghỉ dưỡng của mình.

Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và học hỏi thêm được nhiều kinh nghiệm cũng như kiến thức công nghệ để áp dụng vào thực tế.

**1. Đánh giá kết quả**

**Về công nghệ:**

* Xây dựng thành công hệ thống quản lý đặt phòng khách sạn cơ bản
* Nắm được các kiến thức xây dựng giao diện website: HTML, CSS, JavaScript, Boostrap.
* Nắm vững về cách xây dựng ứng dụng web thông qua thư viện, framework: Asp.Net MVC, Entity Framework trong lập trình.
* Sử dụng thành thạo các công cụ hỗ trợ lập trình: Visual Studio, SQL Server

**Về cài đặt chương trình:**

* Khách hàng xem thông tin về giá cả, hình ảnh phòng, đặt phòng
* Tìm kiếm phòng khách sạn, xem blog về trải nghiệm, đánh giá

**Tính năng khác:**

* Giao diện đẹp gây ấn tượng thân thiện với người dùng, dễ sử dụng
* Áp dụng mô hình MVC để xây dựng hệ thống.

**2. Hướng phát triển**

* Tiếp tục hoàn thiện các tính năng mở rộng phần mềm nhằm cải tiến và nâng cấp chương trình.
* Phát triển thêm tính năng đặt vé máy bay, tích hợp ChatBot để chăm sóc khách hàng
* Tìm hiểu sâu hơn về xu hướng phát triển của các website thương mại điện tử.

# TÀI LIỆU THAM KHẢO

[1] Đinh Văn Hòa (2021) - *Lập Trình C# Nâng Cao,* Nhà Xuất Bản Bách Khoa.

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

[3] **Nguyễn Ngọc Bình Phương, Trần Thanh Phong – *Các giải pháp lập trình C#* (10/2006), NXB Giao thông vận tải.**

[4] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng (2011), *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.

[5] Adam Freeman - *Pro ASP.NET MVC 5* (2013), NXB Apress.

[6] “HTML Tutorial”

Nguồn: <https://www.w3schools.com/html/default.asp>

Truy cập gần nhất: 19/11/2024

[7] “CSS Tutorial”

Nguồn: <https://www.w3schools.com/css/default.asp>

Truy cập gần nhất: 19/11/2024

[8] “Javascript Tutorial”

Nguồn: <https://www.w3schools.com/js/default.asp>

Truy cập gần nhất: 29/11/2024

[9] “Bootstrap Tutorial”

Nguồn: <https://www.w3schools.com/bootstrap4/default.asp>

Truy cập gần nhất: 18/11/2024