BÔ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NGUYỄN TRÃI**

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



**THỰC TẬP CHUYÊN ĐỀ 2**

**Đề tài: Bài toán quản lý cửa hàng thời trang Adidas sử dụng NetCore**

Họ tên sinh viên: Nguyễn Ngọc Việt

Lớp: K21CNT1

Giảng viên hướng dẫn: Th.S Trịnh Văn Chung

Năm thực hiện: 2024

Nội dung

[1. Tổng quan về đề tài 2](#_Toc177579865)

[1.1 Mô tả về hệ thống 2](#_Toc177579866)

[1.2 Các yêu cầu chức năng 3](#_Toc177579867)

[1.3 Công nghệ sử dụng 4](#_Toc177579868)

[1.3.1 ASP.NET Core MVC 4](#_Toc177579869)

[1.3.2 Entity Framework Core (EF Core) 4](#_Toc177579870)

[1.3.3 SQL Server 4](#_Toc177579871)

[1.3.4 Bootstrap & jQuery 4](#_Toc177579872)

[1.3.5 Identity & Role Management 5](#_Toc177579873)

[1.3.6 Restful API 5](#_Toc177579874)

[1.3.7 JSON và AJAX 5](#_Toc177579875)

[1.3.8 SignalR 5](#_Toc177579876)

[1.3.8 Cloud Deployment (Azure hoặc AWS) 6](#_Toc177579877)

[1.3.9 Docker 6](#_Toc177579878)

[2. Phân tích và thiết kế bài toán 6](#_Toc177579879)

# Tổng quan về đề tài

## 1.1 Mô tả về hệ thống

Hệ thống sẽ gồm các nghiệp vụ cơ bản sau:

Quản lý sản phẩm:

- Quản lý danh mục sản phẩm Adidas như: quần áo, giày dép, phụ kiện, v.v.

- Cho phép thêm, chỉnh sửa, xóa sản phẩm và cập nhật thông tin chi tiết (tên, mô tả, hình ảnh, giá cả, kích cỡ, màu sắc, số lượng tồn kho).

- Tích hợp với hệ thống mã vạch để dễ dàng theo dõi và quản lý sản phẩm.

Quản lý kho:

- Kiểm soát số lượng hàng tồn kho, thông báo khi số lượng sản phẩm xuống thấp hoặc đến thời gian nhập thêm hàng.

- Quản lý việc nhập, xuất kho, theo dõi quá trình luân chuyển hàng hóa giữa các cửa hàng hoặc kho tổng.

- Báo cáo chi tiết về tình hình kho hàng, bao gồm tồn kho theo từng sản phẩm, cửa hàng hoặc theo thời gian.

Quản lý bán hàng:

- Hỗ trợ giao dịch bán hàng tại cửa hàng, bao gồm xử lý đơn hàng, thanh toán qua nhiều phương thức (tiền mặt, thẻ tín dụng, ví điện tử).

- Tích hợp hệ thống giảm giá, khuyến mãi và thẻ khách hàng thân thiết.

- Theo dõi lịch sử mua hàng của khách hàng, hỗ trợ quản lý đơn hàng online và tại cửa hàng.

Quản lý khách hàng:

- Quản lý thông tin khách hàng (tên, địa chỉ, số điện thoại, email).

- Theo dõi lịch sử mua sắm của khách hàng, tích lũy điểm thưởng và quản lý chương trình khuyến mãi cá nhân.

- Hỗ trợ hệ thống chăm sóc khách hàng với khả năng gửi thông báo, email marketing, tin nhắn khuyến mãi.

Quản lý nhân viên:

- Phân quyền quản lý nhân viên dựa trên vai trò (quản lý, nhân viên bán hàng, nhân viên kho, v.v.).

- Theo dõi hoạt động của nhân viên, quản lý hiệu suất làm việc và giờ làm việc.

- Hỗ trợ đánh giá hiệu quả công việc và tính lương dựa trên doanh thu bán hàng hoặc số lượng sản phẩm bán ra.

Báo cáo và phân tích:

- Cung cấp các báo cáo doanh thu, lợi nhuận, tình hình kinh doanh theo ngày, tháng, năm hoặc theo từng cửa hàng.

- Phân tích xu hướng mua hàng của khách hàng, hiệu suất bán hàng của từng mặt hàng hoặc nhân viên.

- Giao diện trực quan, cho phép hiển thị số liệu dưới dạng biểu đồ và bảng biểu, hỗ trợ người quản lý đưa ra các quyết định kinh doanh nhanh chóng.

## 1.2 Các yêu cầu chức năng

Chức năng của quản trị viên:

* Quản lý sản phẩm
* Quản lý danh mục
* Quản lý kho
* Quản lý bán hàng
* Quản lý chương trình khuyến mãi
* Quản lý nhân viên
* Quản lý khách hàng
* Báo cáo và phân tích

Chức năng của nhân viên:

* Quản lý bán hàng tại cửa hàng
* Quản lý đơn hàng online
* Quản lý hàng tồn kho
* Hỗ trợ khách hàng

Chức năng của khách hàng:

* Đăng ký/Đăng nhập tài khoản
* Tìm kiếm sản phẩm
* Đặt hàng online
* Quản lý tài khoản cá nhân
* Đổi trả hàng
* Đánh giá và phản hồi

## 1.3 Công nghệ sử dụng

### 1.3.1 ASP.NET Core MVC

* ASP.NET Core là một framework mã nguồn mở do Microsoft phát triển, sử dụng để xây dựng các ứng dụng web hiện đại, đa nền tảng và có khả năng mở rộng cao. Với kiến trúc MVC (Model-View-Controller), hệ thống được tách ra làm ba thành phần chính:
* Model: Quản lý dữ liệu và logic nghiệp vụ.
* View: Quản lý giao diện người dùng và cách hiển thị dữ liệu.
* Controller: Điều khiển luồng hoạt động, tương tác giữa View và Model.
* Ưu điểm:
* Khả năng phát triển nhanh chóng và dễ bảo trì.
* Hỗ trợ tốt việc phân tách các thành phần trong ứng dụng, dễ dàng phát triển và kiểm thử.

### 1.3.2 Entity Framework Core (EF Core)

* Entity Framework Core là ORM (Object-Relational Mapping) của Microsoft, giúp phát triển và tương tác với cơ sở dữ liệu một cách dễ dàng. EF Core cho phép nhà phát triển làm việc với cơ sở dữ liệu bằng các đối tượng .NET thay vì sử dụng SQL thuần túy.
* Các tính năng chính:
* Hỗ trợ truy vấn LINQ (Language Integrated Query), giúp truy vấn cơ sở dữ liệu linh hoạt và mạnh mẽ.
* Quản lý di trú (migrations) để đồng bộ hóa mô hình dữ liệu với cơ sở dữ liệu.
* Hỗ trợ nhiều loại cơ sở dữ liệu khác nhau như SQL Server, MySQL, PostgreSQL, v.v.

### 1.3.3 SQL Server

* Microsoft SQL Server là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) do Microsoft phát triển. Đây là một lựa chọn phổ biến cho các ứng dụng doanh nghiệp nhờ tính bảo mật, hiệu suất cao và tích hợp tốt với các công nghệ của Microsoft như ASP.NET.
* Ưu điểm:
* Khả năng xử lý các giao dịch nhanh chóng và an toàn.
* Hỗ trợ các tính năng tiên tiến như replication, high availability, và data encryption.
* Tích hợp mạnh mẽ với các công cụ phân tích và báo cáo dữ liệu.

### 1.3.4 Bootstrap & jQuery

* Bootstrap là một framework CSS mã nguồn mở, giúp xây dựng các giao diện người dùng thân thiện và tương thích trên nhiều thiết bị (responsive). Nó cung cấp sẵn các thành phần giao diện như nút, bảng, biểu mẫu, v.v., giúp tăng tốc quá trình phát triển giao diện người dùng.
* jQuery là một thư viện JavaScript nhẹ và nhanh, hỗ trợ thao tác với DOM, xử lý sự kiện và AJAX một cách đơn giản. Kết hợp với Bootstrap, jQuery giúp tăng cường khả năng tương tác và trải nghiệm người dùng.
* Ưu điểm:
* Bootstrap cung cấp giao diện hiện đại, dễ tùy chỉnh và đảm bảo tính đáp ứng.
* jQuery giúp giảm thiểu việc viết mã JavaScript phức tạp, cải thiện hiệu suất trang web.

### 1.3.5 Identity & Role Management

* ASP.NET Core Identity là hệ thống quản lý người dùng tích hợp sẵn trong ASP.NET Core, cung cấp các tính năng như đăng nhập, đăng ký, quản lý tài khoản, và phân quyền.
* Role Management: Cho phép phân quyền dựa trên vai trò của người dùng (quản trị viên, nhân viên, khách hàng), đảm bảo chỉ những người có quyền mới có thể truy cập các tính năng cụ thể của hệ thống.
* Các tính năng bảo mật bao gồm:
* Mã hóa mật khẩu mạnh.
* Hỗ trợ xác thực hai yếu tố (2FA) để tăng cường bảo mật.
* Quản lý tài khoản, khôi phục mật khẩu và khóa tài khoản sau nhiều lần đăng nhập thất bại.

### 1.3.6 Restful API

* RESTful API là giao diện lập trình ứng dụng sử dụng các phương thức HTTP để tương tác với dữ liệu (GET, POST, PUT, DELETE). Đây là chuẩn API phổ biến cho phép hệ thống tương tác dễ dàng với các dịch vụ khác, ví dụ như ứng dụng di động hoặc hệ thống quản lý kho bên ngoài.
* Hỗ trợ tích hợp và phát triển các tính năng đa nền tảng, dễ dàng kết nối với các hệ thống khác qua API.

### 1.3.7 JSON và AJAX

* JSON (JavaScript Object Notation) là định dạng dữ liệu nhẹ, dễ dàng truyền tải giữa máy khách và máy chủ. Hệ thống sử dụng JSON để trao đổi dữ liệu, giúp tăng tốc độ xử lý và tương tác trong ứng dụng.
* AJAX (Asynchronous JavaScript and XML) là kỹ thuật tải dữ liệu không đồng bộ giữa máy khách và máy chủ mà không cần tải lại toàn bộ trang. Điều này giúp tăng tốc độ và trải nghiệm người dùng mượt mà hơn.

### 1.3.8 SignalR

* SignalR là một thư viện trong ASP.NET Core giúp hỗ trợ giao tiếp thời gian thực (real-time communication). SignalR được sử dụng để xây dựng các tính năng như cập nhật trạng thái đơn hàng theo thời gian thực, hoặc thông báo ngay lập tức khi có giao dịch mới.
* Tính năng chính:
* Hỗ trợ việc truyền dữ liệu hai chiều giữa máy khách và máy chủ.
* Khả năng tương thích với nhiều nền tảng khác nhau, bao gồm web, di động và desktop.

### 1.3.8 Cloud Deployment (Azure hoặc AWS)

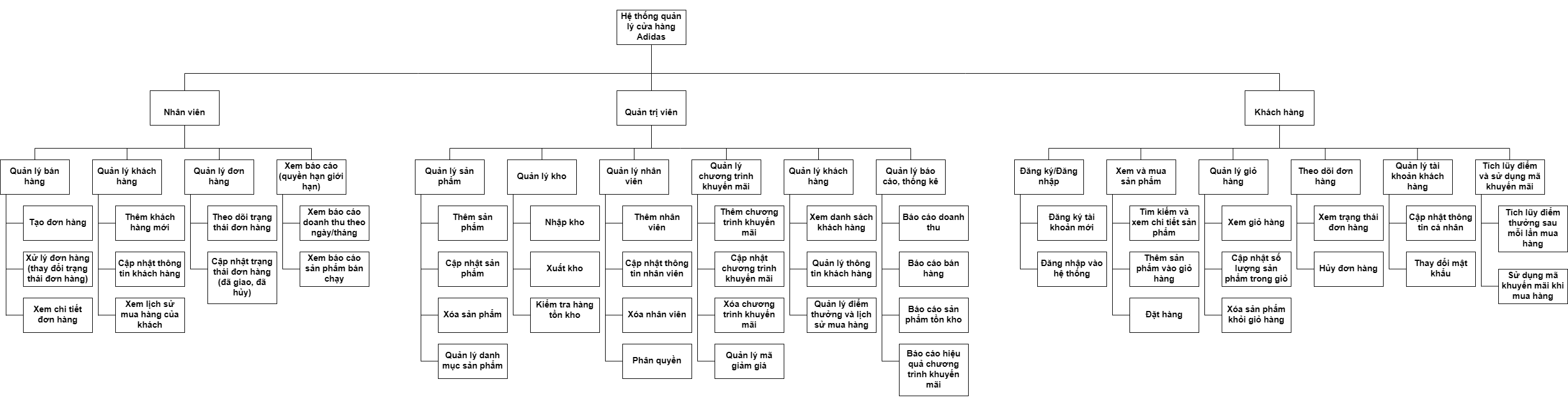
* Hệ thống có thể được triển khai trên các nền tảng đám mây như Microsoft Azure hoặc Amazon Web Services (AWS), cung cấp khả năng mở rộng linh hoạt, bảo mật cao và khả năng sao lưu dữ liệu an toàn.
* Lợi ích:
* Hỗ trợ triển khai nhanh chóng và dễ dàng trên môi trường sản xuất.
* Khả năng mở rộng tài nguyên theo nhu cầu, đảm bảo hiệu suất cao cho hệ thống.
* Hỗ trợ backup dữ liệu, giúp đảm bảo an toàn thông tin khi có sự cố.

### 1.3.9 Docker

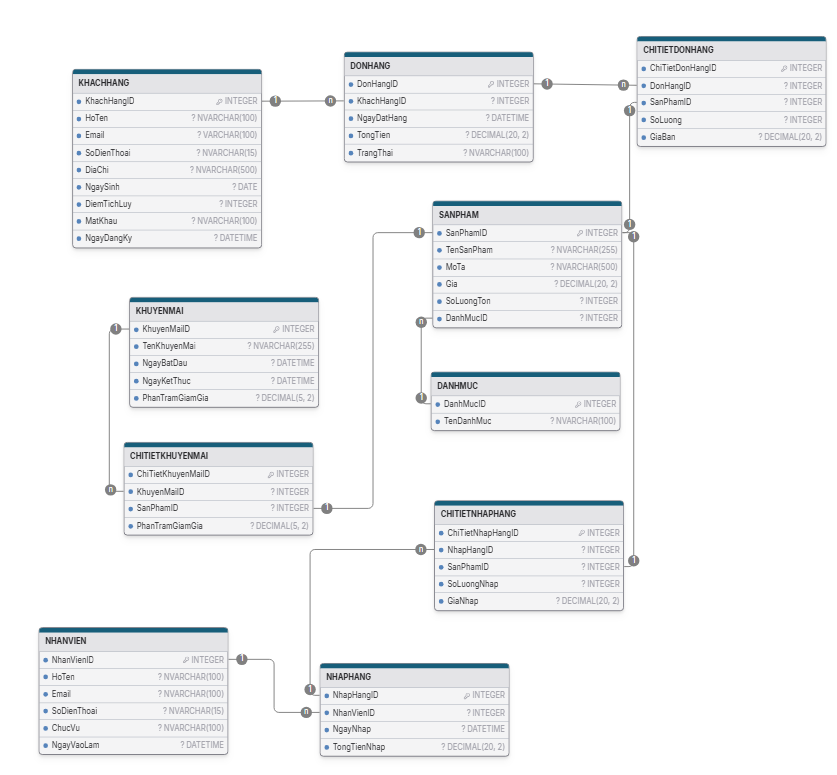
* Docker là một công nghệ containerization, cho phép đóng gói ứng dụng và các thành phần phụ thuộc vào một container duy nhất. Điều này giúp việc triển khai và di chuyển hệ thống trở nên đơn giản và hiệu quả.
* Ưu điểm:
* Khả năng nhất quán về môi trường phát triển và triển khai.
* Giảm thiểu rủi ro lỗi do môi trường khác nhau trong quá trình phát triển và triển khai.

# Phân tích và thiết kế bài toán

## Biểu đồ phân cấp chức năng

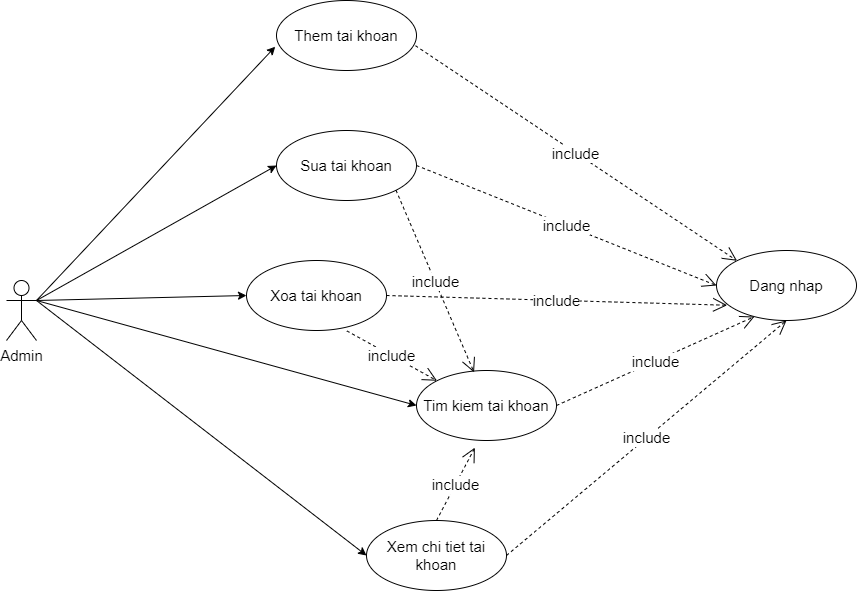


## 2.2 Biểu đồ thực thể quan hệ

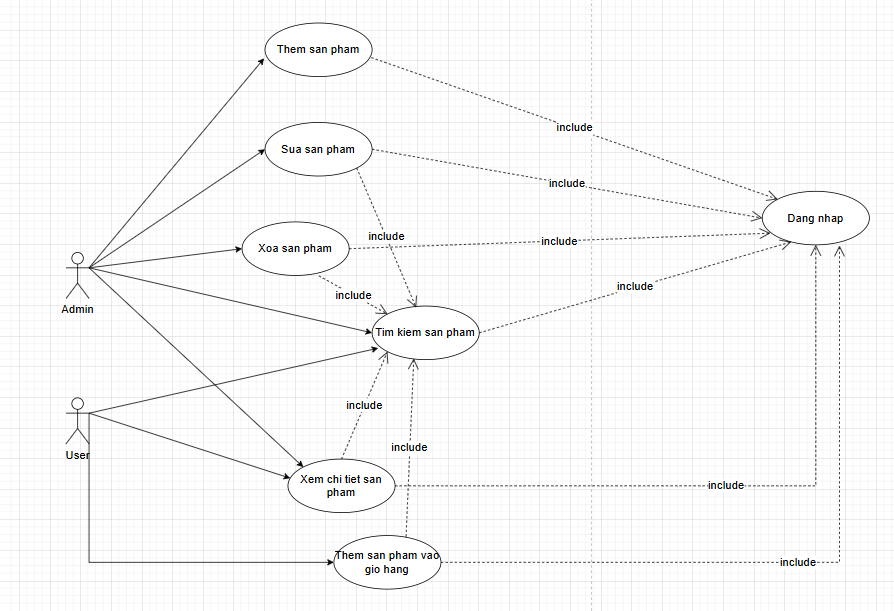


## 2.3 Biểu đồ use case

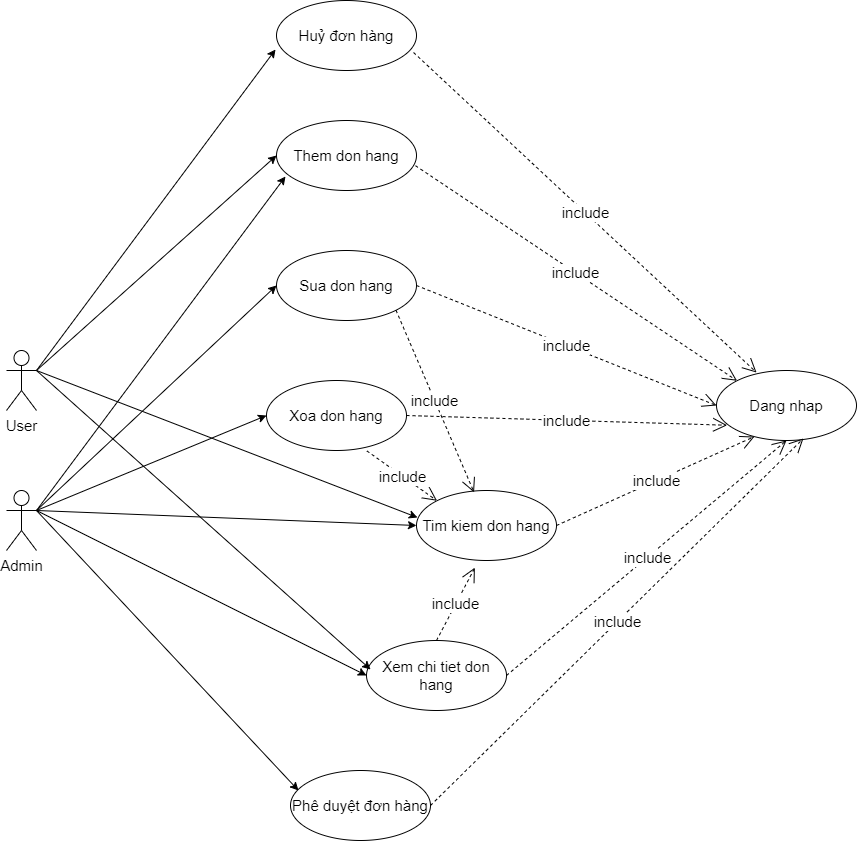
### 2.2.1 Biểu đồ use case Quản lý tài khoản



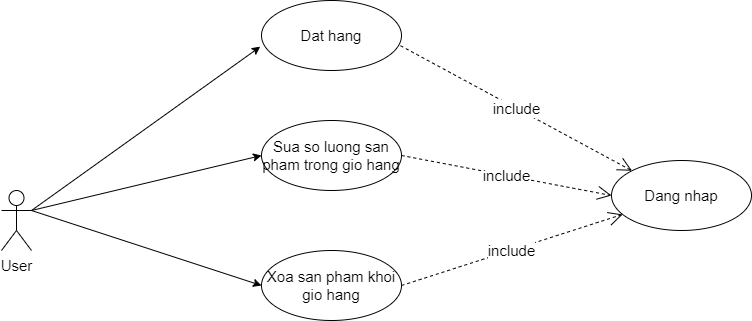
### 2.2.2 Biểu đồ use case Quản lý sản phẩm



### 2.2.3 Biểu đồ use case Quản lý đơn hàng



### 2.2.4 Biểu đồ use case Quản lý giỏ hàng



## 2.3 Mô tả cơ sở dữ liệu

### 2.3.1 Bảng KHACHHANG

Bảng 2.3.1: Bảng KHACHHANG

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| KhachHangID | INT (PK) | Mã khách hàng (Primary Key) | KhachHangID |
| HoTen | NVARCHAR(100) | Họ và tên | HoTen |
| Email | NVARCHAR(100) | Email khách hàng | Email |
| SoDienThoai | NVARCHAR(15) | Số điện thoại | SoDienThoai |
| DiaChi | NVARCHAR(200) | Địa chỉ | DiaChi |
| NgaySinh | DATE | Ngày sinh | NgaySinh |
| DiemTichLuy | INT | Điểm tích lũy | DiemTichLuy |
| MatKhau | NVARCHAR(100) | Mật khẩu được mã hóa | MatKhau |
| NgayDangKy | DATETIME | Ngày đăng ký tài khoản | NgayDangKy |

### 2.3.2 Bảng SANPHAM

Bảng 2.3.1: Bảng SANPHAM

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| SanPhamID | INT (PK) | Mã sản phẩm (Primary Key) | SanPhamID |
| TenSanPham | NVARCHAR(100) | Tên sản phẩm | TenSanPham |
| MoTa | NVARCHAR(500) | Mô tả sản phẩm | MoTa |
| Gia | DECIMAL(18, 2) | Giá bán | Gia |
| SoLuongTon | INT | Số lượng tồn kho | SoLuongTon |
| DanhMucID | INT (FK) | Khóa ngoại liên kết với bảng DanhMuc | DanhMucID |

### 2.3.3 Bảng DANHMUC

Bảng 2.3.1: Bảng DANHMUC

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| DanhMucID | INT (PK) | Mã danh mục (Primary Key) | DanhMucID |
| TenDanhMuc | NVARCHAR(100) | Tên danh mục sản phẩm | TenDanhMuc |

### 2.3.4 Bảng DONHANG

Bảng 2.3.1: Bảng SANPHAM

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| DonHangID | INT (PK) | Mã đơn hàng (Primary Key) | DonHangID |
| KhachHangID | INT (FK) | Khóa ngoại liên kết với bảng KhachHang | KhachHangID |
| NgayDatHang | DATETIME | Ngày đặt hàng | NgayDatHang |
| TongTien | DECIMAL(18, 2) | Tổng tiền | TongTien |
| TrangThai | NVARCHAR(50) | Trạng thái đơn hàng (Đang xử lý, Đã giao, Hủy) | TrangThai |

### 2.3.5 Bảng CHITIETDONHANG

Bảng 2.3.1: Bảng CHITIETDONHANG

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| ChiTietDonHangID | INT (PK) | Mã chi tiết đơn hàng (Primary Key) | ChiTietDonHangID |
| DonHangID | INT (FK) | Khóa ngoại liên kết với bảng DonHang | DonHangID |
| SanPhamID | INT (FK) | Khóa ngoại liên kết với bảng SanPham | SanPhamID |
| SoLuong | INT | Số lượng sản phẩm | SoLuong |
| GiaBan | DECIMAL(18, 2) | Giá bán sản phẩm tại thời điểm mua | GiaBan |

### 2.3.6 Bảng NHANVIEN

Bảng 2.3.1: Bảng NHANVIEN

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| NhanVienID | INT (PK) | Mã nhân viên (Primary Key) | NhanVienID |
| HoTen | NVARCHAR(100) | Họ và tên nhân viên | HoTen |
| Email | NVARCHAR(100) | Email nhân viên | Email |
| SoDienThoai | NVARCHAR(15) | Số điện thoại | SoDienThoai |
| ChucVu | NVARCHAR(50) | Chức vụ (Quản lý, Nhân viên bán hàng, v.v.) | ChucVu |
| NgayVaoLam | DATETIME | Ngày bắt đầu làm việc | NgayVaoLam |

### 2.3.7 Bảng KHUYENMAI

Bảng 2.3.1: Bảng KHUYENMAI

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| KhuyenMaiID | INT (PK) | Mã khuyến mãi (Primary Key) | KhuyenMaiID |
| TenKhuyenMai | NVARCHAR(100) | Tên chương trình khuyến mãi | TenKhuyenMai |
| NgayBatDau | DATETIME | Ngày bắt đầu khuyến mãi | NgayBatDau |
| NgayKetThuc | DATETIME | Ngày kết thúc khuyến mãi | NgayKetThuc |
| PhanTramGiamGia | DECIMAL(5, 2) | Phần trăm giảm giá | PhanTramGiamGia |

### 2.3.8 Bảng CHITIETKHUYENMAI

Bảng 2.3.1: Bảng CHITIETKHUYENMAI

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| ChiTietKhuyenMaiID | INT (PK) | Mã chi tiết khuyến mãi (Primary Key) | ChiTietKhuyenMaiID |
| KhuyenMaiID | INT (FK) | Khóa ngoại liên kết với bảng KhuyenMai | KhuyenMaiID |
| SanPhamID | INT (FK) | Khóa ngoại liên kết với bảng SanPham | SanPhamID |
| PhanTramGiamGia | DECIMAL(5, 2) | Phần trăm giảm giá cho sản phẩm | PhanTramGiamGia |

### 2.3.9 Bảng NHAPHANG

Bảng 2.3.1: Bảng NHAPHANG

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| NhapHangID | INT (PK) | Mã nhập hàng (Primary Key) | NhapHangID |
| NhanVienID | INT (FK) | Khóa ngoại liên kết với bảng NhanVien | NhanVienID |
| NgayNhap | DATETIME | Ngày nhập hàng | NgayNhap |
| TongTienNhap | DECIMAL(18, 2) | Tổng tiền nhập hàng | TongTienNhap |

### 2.3.10 Bảng CHITIETNHAPHANG

Bảng 2.3.1: Bảng CHITIETNHAPHANG

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Tên cột |
| ChiTietNhapHangID | INT (PK) | Mã chi tiết nhập hàng (Primary Key) | ChiTietNhapHangID |
| NhapHangID | INT (FK) | Khóa ngoại liên kết với bảng NhapHang | NhapHangID |
| SanPhamID | INT (FK) | Khóa ngoại liên kết với bảng SanPham | SanPhamID |
| SoLuongNhap | INT | Số lượng sản phẩm nhập vào | SoLuongNhap |
| GiaNhap | DECIMAL(18, 2) | Giá nhập sản phẩm | GiaNhap |

### 2.3.11 Quan hệ giữa các bảng

* Khách hàng có thể tạo nhiều Đơn hàng (1-N).
* Đơn hàng có nhiều Chi tiết đơn hàng, và mỗi Chi tiết đơn hàng thuộc về một Sản phẩm (N-N qua bảng ChiTietDonHang).
* Sản phẩm thuộc về một Danh mục (N-1).
* Nhân viên quản lý các hoạt động nhập hàng thông qua Phiếu nhập hàng (1-N).
* Sản phẩm có thể tham gia nhiều chương trình Khuyến mãi (N-N qua bảng ChiTietKhuyenMai).

# Kiểm thử

## 3.1 Phương pháp kiểm thử

Trong quá trình phát triển hệ thống đặt đồ ăn nhanh cho cửa hàng thời trang Adidas, dự án đã áp dụng nhiều phương pháp kiểm thử khác nhau để đảm bảo tính chính xác và hiệu quả của hệ thống. Cụ thể, các phương pháp kiểm thử được sử dụng bao gồm:

* Kiểm thử đơn vị (Unit Testing): Sử dụng kiểm thử đơn vị để kiểm tra tính năng của các thành phần nhỏ trong hệ thống, chẳng hạn như:
* Kiểm tra tính hợp lệ của dữ liệu đầu vào khi khách hàng chọn quần áo, giày(ví dụ: số lượng đồ trong giỏ phải là số nguyên dương).
* Xác nhận quá trình tính toán tổng tiền đơn hàng dựa trên giá đồ và số lượng.
* Ví dụ, tôi đã viết một trường hợp kiểm thử để đảm bảo rằng khi khách hàng đặt 2 cái quần và 1 cái áo, tổng tiền phải được tính đúng dựa trên giá mỗi đồ.
* Kiểm thử tích hợp (Integration Testing): Sau khi kiểm thử từng thành phần riêng lẻ, tôi đã tiến hành kiểm thử tích hợp để đảm bảo sự kết nối và hoạt động đúng đắn giữa các mô-đun. Ví dụ:
* Kiểm tra quá trình đặt hàng từ khi khách hàng chọn món, xác nhận đơn hàng, đến khi thông tin đơn hàng được chuyển tới hệ thống thanh toán và lưu trữ vào cơ sở dữ liệu.
* Đảm bảo rằng khi đơn hàng được thanh toán thành công, hệ thống tự động cập nhật trạng thái đơn hàng và giảm số lượng hàng tồn kho.
* Kiểm thử hệ thống (System Testing): Tôi đã thực hiện kiểm thử toàn bộ hệ thống để đảm bảo rằng tất cả các chức năng hoạt động đúng như mong đợi. Các kịch bản kiểm thử hệ thống bao gồm:
* Đặt hàng từ trang web, xác nhận đơn hàng và nhận email xác nhận.
* Hủy đơn hàng trước khi thanh toán và kiểm tra xem đơn hàng có được hủy thành công hay không.
* Kiểm tra khả năng hiển thị của các loại đồ và giá cả trên giao diện người dùng.
* Kiểm thử chấp nhận người dùng (User Acceptance Testing - UAT): Để đảm bảo hệ thống phù hợp với yêu cầu thực tế của cửa hàng thời trang Adidas, tôi đã tiến hành kiểm thử với sự tham gia của nhân viên tại cửa hàng. Họ đã thử nghiệm đặt hàng trực tiếp trên hệ thống, đánh giá tính dễ sử dụng của giao diện và sự nhanh chóng trong quy trình đặt hàng.

## 3.2 Công cụ kiểm thử

Sử dụng JUnit để viết các trường hợp kiểm thử đơn vị cho từng thành phần nhỏ như tính toán tổng tiền đơn hàng, kiểm tra trạng thái đơn hàng sau khi thanh toán

Postman: Tôi đã sử dụng Postman để kiểm thử các API của hệ thống. Ví dụ:

Kiểm thử API tạo đơn hàng với các thông tin như danh mục quần áo giày dép, số lượng, và phương thức thanh toán.

Kiểm thử API cập nhật trạng thái đơn hàng sau khi thanh toán thành công.

Selenium: Để kiểm thử giao diện web, tôi sử dụng Selenium để tự động hóa các bước kiểm thử như đặt đồ, xác nhận đơn hàng, và kiểm tra phản hồi từ hệ thống.