**TÊN SINH VIÊN**

**LÝ CHÍNH ĐẠT**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI**

**XÂY DỰNG WEBSITE CHO CỬA HÀNG BÁN ĐIỆN THOẠI**

**HÀ NỘI - 2022**

****BỘ GIÁO DỤC VÀ ĐÀO TẠO****

****TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT****



**ĐỒ ÁN TỐT NGHIỆP**

**CHUYÊN NGÀNH MẠNG MÁY TÍNH**

**Hà Nội - 2022**

****BỘ GIÁO DỤC VÀ ĐÀO TẠO****

****TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT****

**ĐỀ TÀI**

**XÂY DỰNG WEBSITE CHO CỬA HÀNG BÁN ĐIỆN THOẠI**

**Sinh viên thực hiện**

(Chữ ký)

**Họ tên sv**

**Cán bộ hướng dẫn**

**Họ tên giảng viên hướng dẫn**

**TÊN SINH VIÊN**

**LÝ CHÍNH ĐẠT**



# MỤC LỤC

[MỤC LỤC 1](#_Toc107792226)

[DANH MỤC CÁC HÌNH VẼ 5](#_Toc107792227)

[DANH MỤC CÁC BẢNG 7](#_Toc107792228)

[THÔNG TIN KẾT QUẢ ĐỒ ÁN 8](#_Toc107792229)

[MỞ ĐẦU 9](#_Toc107792230)

[CHƯƠNG 1 : TỔNG QUAN VỀ THIẾT KẾ WEBSITE 10](#_Toc107792231)

[1. Giới thiệu về Front end và Back end – 2 thành phần tạo nên trang web 10](#_Toc107792232)

[1.1. Front end 10](#_Toc107792233)

[1.2. Back end 12](#_Toc107792234)

[2. Những thành phần chính của Front end 17](#_Toc107792235)

[2.1. HTML – “xương sống” của trang web 17](#_Toc107792236)

[2.2. CSS – kiểu dáng của trang web 19](#_Toc107792237)

[2.3. JavaScript – yếu tố động và tương tác của trang web 20](#_Toc107792238)

[2.4. TypeScript – ngôn ngữ nâng cấp của JavaScript 22](#_Toc107792239)

[3. Những thành phần chính của Back end 23](#_Toc107792240)

[3.1. T-SQL – ngôn ngữ truy vấn database 23](#_Toc107792241)

[3.2. Java – ngôn ngữ mạnh mẽ và đa nền tảng 24](#_Toc107792242)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT LIÊN QUAN 27](#_Toc107792243)

[1. Visual Studio Code (VS Code) 27](#_Toc107792244)

[1.1. VS Code là gì? 27](#_Toc107792245)

[1.2. Những tính năng nổi bật của VS Code 28](#_Toc107792246)

[1.3. Ưu điểm của VS Code 30](#_Toc107792247)

[1.4. Nhược điểm của VS Code 32](#_Toc107792248)

[2. Angular 32](#_Toc107792249)

[2.1. Angular là gì? 32](#_Toc107792250)

[2.2. Các đặc trưng cơ bản của Angular 33](#_Toc107792251)

[2.3. Ứng dụng của Angular 34](#_Toc107792252)

[2.4. Ưu điểm của Angular 34](#_Toc107792253)

[2.5. Nhược điểm của Angular 35](#_Toc107792254)

[3. SQL Server 35](#_Toc107792255)

[3.1. SQL Server là gì ? 35](#_Toc107792256)

[3.2. Ứng dụng của SQL Server 36](#_Toc107792257)

[3.3. Ưu điểm của SQL Server 36](#_Toc107792258)

[3.4. Nhược điểm của SQL Server 36](#_Toc107792259)

[3.5. Phân loại sql server 36](#_Toc107792260)

[4. Eclipse 37](#_Toc107792261)

[4.1. Eclipse là gì? 37](#_Toc107792262)

[4.2. Ưu nhược điểm của Eclipse 38](#_Toc107792263)

[4.3. Chức năng nổi bật của Eclipse là gì? 39](#_Toc107792264)

[4.4. Công nghệ Servlet 39](#_Toc107792265)

[4.5. Spring 40](#_Toc107792266)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 44](#_Toc107792267)

[1. Phân tích hệ thống về chức năng 44](#_Toc107792268)

[1.1. Biểu đồ phân cấp chức năng 44](#_Toc107792269)

[1.2. Đặc tả chức năng của hệ thống 44](#_Toc107792270)

[1.3. Use-case-diagram 46](#_Toc107792271)

[1.4. Mô tả chi tiết các chức năng của hệ thống 50](#_Toc107792272)

[2. Phân tích hệ thống về dữ liệu 69](#_Toc107792273)

[2.1. Xác định các thực thể và các thuộc tính cho thực thể 69](#_Toc107792274)

[2.2. Xác định các liên kết và các mối quan hệ của thực thể 70](#_Toc107792275)

[3. Mô hình thực thể/ liên kết (Mô hình ER) 72](#_Toc107792276)

[CHƯƠNG 4 : THIẾT KẾ CƠ SỞ DỮ LIỆU 73](#_Toc107792277)

[1. Mô tả thực thể kết hợp 73](#_Toc107792278)

[1.1. Danh mục 73](#_Toc107792279)

[1.2. Sản phẩm 73](#_Toc107792280)

[1.3. Ảnh của sản phẩm 73](#_Toc107792281)

[1.4. Khách hàng 74](#_Toc107792282)

[1.5. Giỏ hàng 74](#_Toc107792283)

[1.6. Voucher giảm giá 74](#_Toc107792284)

[1.7. Đơn hàng 74](#_Toc107792285)

[1.8. Chi tiết đơn hàng 75](#_Toc107792286)

[1.9. Phản hồi của khách hàng 75](#_Toc107792287)

[1.10. Tài khoản 75](#_Toc107792288)

[1.11. Quyền trong trang web 75](#_Toc107792289)

[1.12. Phân quyền cho người dùng 76](#_Toc107792290)

[2. Mô tả các mối kết hợp 76](#_Toc107792291)

[2.1. Mối kết hợp: Category và Product 76](#_Toc107792292)

[2.2. Mối kết hợp: Tài khoản và Quyền gán cho tài khoản 76](#_Toc107792293)

[2.3. Mối kết hợp: Quyền trong website và Quyền gán cho tài khoản 77](#_Toc107792294)

[2.4. Mối kết hợp: Khách hàng và Đơn hàng 77](#_Toc107792295)

[2.5. Mối kết hợp: Khách hàng và Phản hồi của khách hàng 77](#_Toc107792296)

[2.6. Mối kết hợp: Đơn hàng và Chi tiết đơn hàng 78](#_Toc107792297)

[2.7. Mối kết hợp: Sản phẩm và Chi tiết đơn hàng 78](#_Toc107792298)

[2.8. Mối kết hợp: Khách hàng và Giỏ hàng 78](#_Toc107792299)

[2.9. Mối kết hợp: Voucher và Đơn hàng 78](#_Toc107792300)

[2.10. Mối kết hợp: Sản phẩm và Ảnh sản phẩm 79](#_Toc107792301)

[2.11. Mối kết hợp: Sản phẩm và Phản hồi của khách hàng 79](#_Toc107792302)

[2.12. Mối kết hợp: Tài khoản và Khách hàng 79](#_Toc107792303)

[3. Mô tả chi tiết các bảng 79](#_Toc107792304)

[3.1. Category 79](#_Toc107792305)

[3.2. Product 80](#_Toc107792306)

[3.3. ProductImage 81](#_Toc107792307)

[3.4. Customer 81](#_Toc107792308)

[3.5. Cart 82](#_Toc107792309)

[3.6. Promotions 82](#_Toc107792310)

[3.7. Orders 83](#_Toc107792311)

[3.8. OrderDetail 83](#_Toc107792312)

[3.9. FeedBack 84](#_Toc107792313)

[3.10. Users 85](#_Toc107792314)

[3.11. Roles 85](#_Toc107792315)

[3.12. UserRoles 85](#_Toc107792316)

[4. Sơ đồ quan hệ dữ liệu giữa các bảng 86](#_Toc107792317)

[CHƯƠNG 5 : DEMO SẢN PHẨM 87](#_Toc107792318)

[1. Giao diện người dùng 87](#_Toc107792319)

# DANH MỤC CÁC HÌNH VẼ

[Hình 1‑1: Front-end gồm các yếu tố tạo nên giao diện của các trang web 11](#_Toc107792320)

[Hình 1‑2: Front-end giúp giao diện web trực quan và hấp dẫn hơn 12](#_Toc107792321)

[Hình 1‑3: Back end là gì? 13](#_Toc107792322)

[Hình 1‑4: Backend là cầu nối thông tin giữa website và client 14](#_Toc107792323)

[Hình 1‑5: Backend giúp các chức năng của website hoạt động mượt mà 15](#_Toc107792324)

[Hình 1‑6: Server chính là trung tâm điều hành của backend 16](#_Toc107792325)

[Hình 1‑7: API là giao thức giúp kết nối nhiều ứng dụng website với nhau khi lập trình backend 17](#_Toc107792326)

[Hình 1‑8: Logo HTML 5 19](#_Toc107792327)

[Hình 1‑9: Logo CSS 3 21](#_Toc107792328)

[Hình 1‑10: HTML, CSS và JS là bộ 3 không thể thiếu trong mảng Front end 22](#_Toc107792329)

[Hình 1‑11: Logo Java 26](#_Toc107792330)

[Hình 1‑12: Thông dịch trong Java 26](#_Toc107792331)

[Hình 2‑1: Logo VS Code 29](#_Toc107792332)

[Hình 2‑2: Logo Angular 34](#_Toc107792333)

[Hình 2‑3: Logo SQL Server Management Studio 37](#_Toc107792334)

[Hình 2‑4: Logo Eclipse 39](#_Toc107792335)

[Hình 2‑5: Logo Spring 41](#_Toc107792336)

[Hình 2‑6: Cấu trúc Spring MVC và các thành phần chứa bên trong Model – View – Controller. 42](#_Toc107792337)

[Hình 3‑1: Biểu đồ phân cấp chức năng 45](#_Toc107792338)

[Hình 3‑2: Use-case-diagram cho khách hàng 47](#_Toc107792339)

[Hình 3‑3: Use-case-diagram cho khách hàng vãng lai 48](#_Toc107792340)

[Hình 3‑4: Use-case-diagram cho bộ phận quản trị hệ thống 48](#_Toc107792341)

[Hình 3‑5: Use-case-diagram admin quản lý sản phẩm 49](#_Toc107792342)

[Hình 3‑6: Use-case-diagram admin quản lý danh mục 49](#_Toc107792343)

[Hình 3‑7: Use-case-diagram admin quản lý đơn đặt hàng 50](#_Toc107792344)

[Hình 3‑8: Use-case-diagram admin quản lý tin tức 50](#_Toc107792345)

[Hình 3‑9: Use-case-diagram admin quản lý phản hồi của khách hàng 51](#_Toc107792346)

[Hình 3‑10: Liên kết giữa các thực thể 72](#_Toc107792347)

[Hình 3‑11: Mô hình thực thể liên kết ER 73](#_Toc107792348)

[Hình 4‑1: Mối kết hợp giữa Danh mục và Sản phẩm 77](#_Toc107792349)

[Hình 4‑2: Mối kết hợp giữa Tài khoản và Quyền gán cho tài khoản 77](#_Toc107792350)

[Hình 4‑3: Mối kết hợp giữa Quyền trong website và Quyền gán cho tài khoản 78](#_Toc107792351)

[Hình 4‑4: Mối kết hợp giữa Khách hàng và Đơn hàng 78](#_Toc107792352)

[Hình 4‑5: Mối kết hợp giữa Khách hàng và Phản hồi của khách hàng 78](#_Toc107792353)

[Hình 4‑6: Mối kết hợp giữa Đơn hàng và Chi tiết đơn hàng 79](#_Toc107792354)

[Hình 4‑7: Mối kết hợp giữa Sản phẩm và Chi tiết đơn hàng 79](#_Toc107792355)

[Hình 4‑8: Mối kết hợp giữa Khách hàng và Giỏ hàng 79](#_Toc107792356)

[Hình 4‑9: Mối kết hợp giữa Voucher và Đơn hàng 79](#_Toc107792357)

[Hình 4‑10: Mối kết hợp giữa Sản phẩm và Ảnh sản phẩm 80](#_Toc107792358)

[Hình 4‑11: Mối kết hợp giữa Sản phẩm và Phản hồi của khách hàng 80](#_Toc107792359)

[Hình 4‑12: Mối kết hợp giữa Tài khoản và Khách hàng 80](#_Toc107792360)

[Hình 4‑13: Sơ đồ quan hệ dữ liệu giữa các bảng 87](#_Toc107792361)

[Hình 5‑1: Giao diện trang chủ 88](#_Toc107792362)

[Hình 5‑2: Thanh tìm kiếm sản phẩm 89](#_Toc107792363)

[Hình 5‑3: Giao diện Cửa hàng 90](#_Toc107792364)

[Hình 5‑4: Giao diện chio tiết sản phẩm 91](#_Toc107792365)

[Hình 5‑5: Giao diện xem trước giỏ hàng khi thêm sản phẩm vào giỏ 92](#_Toc107792366)

[Hình 5‑6: Giao diện giỏ hàng 93](#_Toc107792367)

[Hình 5‑7: Giao diện Đăng nhập/Đăng ký (bắt buộc phải đăng nhập để tới trang thanh toán) 93](#_Toc107792368)

[Hình 5‑8: Giao diện thanh toán sau khi đăng nhập xong 94](#_Toc107792369)

[Hình 5‑9: Sau khi nhập thông tin đầy đủ, nút Đặt hàng sẽ hiện ra 94](#_Toc107792370)

[Hình 5‑10: Sau khi nhấn nút Đặt hàng sẽ chuyển tới trang xác nhận đặt hàng thành công 95](#_Toc107792371)

[Hình 5‑11: Ngay lập tức thông tin sẽ được lưu vào cơ sở dữ liệu và hiển thị trong trang quản trị đơn hàng của admin 95](#_Toc107792372)

[Hình 5‑12: Người quản trị có thể chọn nút Detail để xem đơn hàng chi tiết 95](#_Toc107792373)

# DANH MỤC CÁC BẢNG

[Bảng 4‑1: Bảng chứa thông tin các danh mục 81](#_Toc102048783)

[Bảng 4‑2: Bảng chứa thông tin sản phẩm 82](#_Toc102048784)

[Bảng 4‑3: Bảng chứa thông tin ảnh sản phẩm 82](#_Toc102048785)

[Bảng 4‑4: Bảng chứa thông tin ảnh khách hàng 83](#_Toc102048786)

[Bảng 4‑5: Bảng chứa thông tin ảnh giỏ hàng 83](#_Toc102048787)

[Bảng 4‑6: Bảng chứa thông tin voucher giảm giá 84](#_Toc102048788)

[Bảng 4‑7: Bảng chứa thông tin đơn hàng 84](#_Toc102048789)

[Bảng 4‑8: Bảng chứa thông tin chi tiết đơn hàng. 85](#_Toc102048790)

[Bảng 4‑9: Bảng chứa thông tin phản hồi của khách hàng 85](#_Toc102048791)

[Bảng 4‑10: Bảng chứa thông tin tài khoản 86](#_Toc102048792)

[Bảng 4‑11: Bảng chứa thông tin quyền 86](#_Toc102048793)

[Bảng 4‑12: Bảng chứa thông tin phân quyền 87](#_Toc102048794)

# THÔNG TIN KẾT QUẢ ĐỒ ÁN

1. Thông tin chung

|  |  |
| --- | --- |
| Tên báo cáo: | Xây dựng website bán điện thoại trực tuyến |
| Sinh viên thực hiện: Lý Chính Đạt | |
| Mã sinh viên: 1721050116 Lớp: DCCTMM62C | |
| Hệ đào tạo: Chính quy | |
| Điện thoại: 0344541807 | |
| Email: lychinhdat1999000@gmail.com | |
| Thời gian thực hiện: 2022 | |

2. Mục tiêu

Xây dựng được một website bán điện thoại với đầy đủ các tính năng cơ bản mà một website thương mại điện tử cần có.

3. Nội dung chính

Gồm 2 chương:

- CHƯƠNG I: Tổng quan về thiết kế website

Giới thiệu về những ngôn ngữ cấu thành nên website thuộc 2 mảng chính là Front end và Back end: HTML, CSS, JavaScript, TypeScript, T-SQL, Java.

- CHƯƠNG II: Cơ sở lý thuyết liên quan

Giới thiệu những framework và ứng dụng được sử dụng để xây dựng website: VS Code, Angular, SQL Server, Eclipse.

4. Kết quả chính đạt được

Tự xây dựng được một trang web phục vụ cho công việc bán điện thoại. Nắm được và vận dụng được các công nghệ, các ngôn ngữ phục vụ cho quá trình thiết kế và lập trình chức năng cho website.

# MỞ ĐẦU

Chúng ta đang sống trong thời kỳ của cuộc Cách mạng công nghiệp lần thứ 4. Đây là thời kỳ gắn với những đột phá về công nghệ. Công nghệ thông tin được coi là nền tảng cho các ngành khác phát triển. Sự thay đổi công nghệ gây tác động lớn tới thị trường lao động. Theo đó, nhiều việc làm truyền thống sẽ mất đi, nhiều công việc mới, cơ hội mới sẽ xuất hiện.

Có rất nhiều loại hình công việc trong ngành công nghệ thông tin từ nhân viên kinh doanh đến thiết kế web, tùy thuộc vào các điểm mạnh và sự yêu thích mà chúng ta có thể tìm được cho mình vai trò phù hợp. Sau đây là một vài lựa chọn tiêu biểu:

* Nhân viên phân tích dữ liệu
* Quản trị hệ thống
* Lập trình viên
* Kỹ sư phần mềm
* Nhân viên phân tích hệ thống
* Chuyên viên hỗ trợ kỹ thuật/ người sử dụng cuối cùng
* Thiết kế web/ dịch vụ Internet

Trong số đó, lập trình web là công việc của lập trình viên. Các lập trình viên sẽ có nhiệm vụ nhận toàn bộ dữ liệu từ bộ phận thiết kế web để chuyển thành một hệ thống website hoàn chỉnh. Website này có tương tác với cơ sở dữ liệu và tương tác với người dùng dựa trên ngôn ngữ máy tính.

Trong bài luận này, em sẽ trình bày về những lý thuyết về các ngôn ngữ mà em sử dụng trong việc tạo ra một trang web thương mại điện tử và những công cụ, công nghệ cần thiết áp dụng vào để tạo ra một trang web bán hàng hoàn chỉnh nhất trong khả năng của mình.

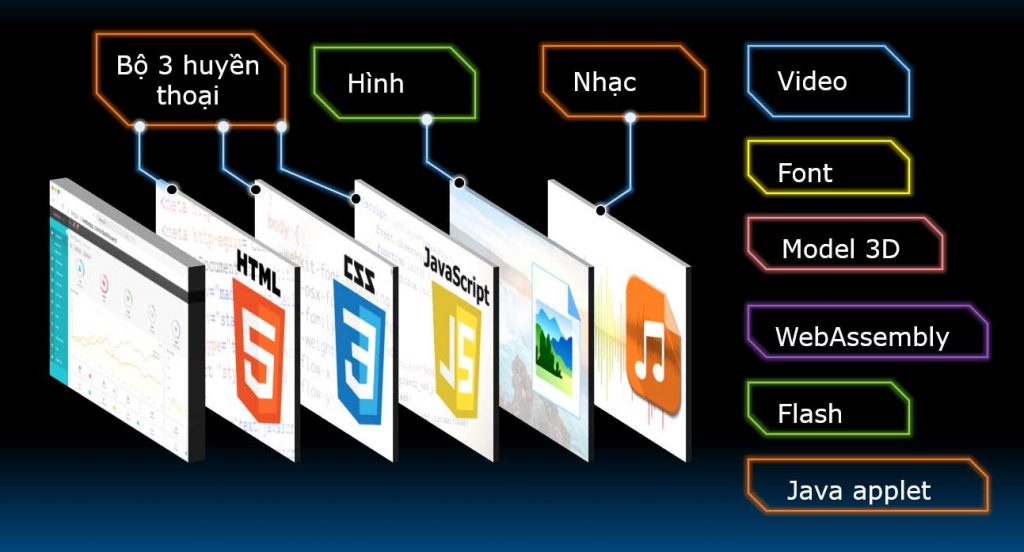
# : TỔNG QUAN VỀ THIẾT KẾ WEBSITE

## Giới thiệu về Front end và Back end – 2 thành phần tạo nên trang web

### Front end

#### Front end là gì?

Front-end là những gì mà người dùng nhìn thấy khi truy cập vào các trang web hay ứng dụng web. Hiểu một cách đơn giản, Front-end chính là giao diện của một website. Nó mang lại cho người sử dụng các trải nghiệm tương tác (nghe, nhìn) trên trang web đó. Vì thế, Front-end còn được gọi là “Client-side”.



Hình ‑: Front-end gồm các yếu tố tạo nên giao diện của các trang web

Trong thiết kế web, Front-end được tạo ra bởi 3 loại ngôn ngữ cơ bản là HTML, CSS và JavaScript. Các lập trình viên Front-end sẽ cần đảm bảo nội dung hiển thị tốt trên mọi nền tảng khác nhau.

Thông thường, Front-end của một trang web sẽ được người dùng tương tác trực tiếp theo các khía cạnh như:

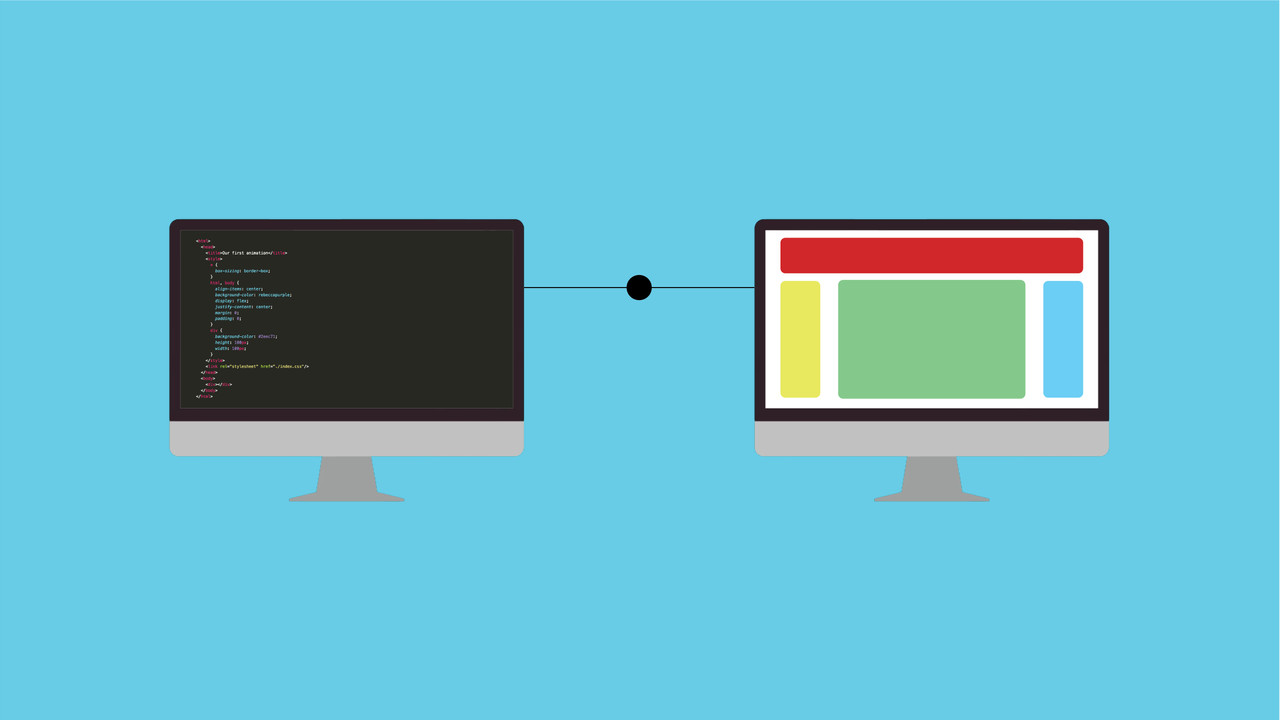
* Nhận biết màu sắc chủ đạo, logo
* Khai thác thông tin trên web (văn bản, hình ảnh, âm thanh…)
* Sử dụng các button, toolbars…

#### Vai trò của Front end trong lập trình web

* **Xây dựng giao diện trực quan**

Một trong những mục tiêu chính của việc phát triển Front-end là nâng cao trải nghiệm người dùng. Nhờ có Front-end, giao diện của các trang web trở nên trực quan và dễ sử dụng hơn. Nếu ví trang web là một bài báo, thì Front-end đóng vai trò như một cái “Tít” hấp dẫn thu hút độc giả truy cập vào.

Đối với các website của doanh nghiệp, Front-end giữ một vai trò cực kỳ quan trọng. Nếu trang web có giao diện trực quan, “bắt mắt” sẽ thu hút được nhiều khách hàng. Thông qua trang web, công ty sẽ đưa được các thông tin về sản phẩm đến người tiêu dùng và trực tiếp tạo ra lợi nhuận.



Hình ‑: Front-end giúp giao diện web trực quan và hấp dẫn hơn

* **Giúp nội dung hiển thị tốt trên mọi nền tảng**

Hiện nay, nhờ sự hỗ trợ của các framework, lập trình viên Front-end có thể đảm bảo nội dung hiển thị tốt trên các nền tảng khác nhau.

Các công ty công nghệ đã cho ra đời rất nhiều framework hỗ trợ phát triển Front-end như:

Google: Material, AngularJS…

Facebook: React native, React,…

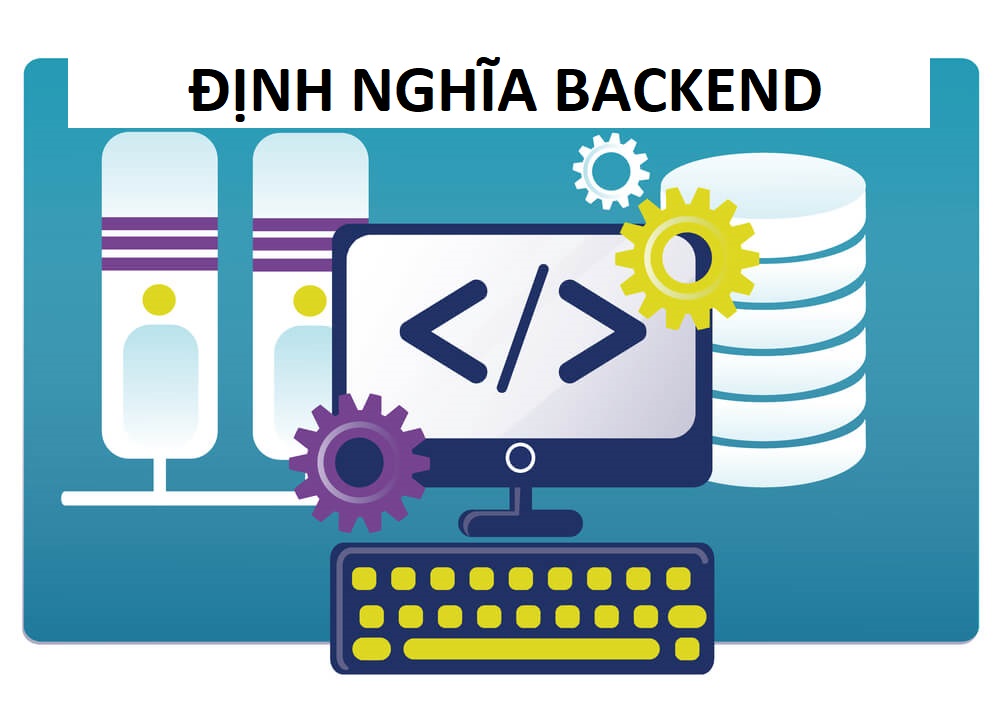
Twitter: Bootstrap, hogan.js…

Quả thực, Front-end là yếu tố không thể thiếu đối với bất kỳ trang web hay ứng dụng web nào. Nhờ có Front-end, người dùng có những trải nghiệm trực quan hơn trên các thiết bị khác nhau.

### Back end

#### Back end là gì?

Backend là những tương tác ngầm giữa ba phần chính gồm server, application và database. Dù không được thấy hay tương tác bởi người dùng, phần hậu trường này lại rất quan trọng với website. [Backend](https://vn.got-it.ai/blog/backend-developer-la-gi-backend-developer-can-hoc-nhung-gi(opens%20in%20a%20new%20tab)) cung cấp các chức năng mượt mà cho website và là cây cầu bắc ngang giữa client và database. Phải có [backend](https://vn.got-it.ai/blog/top-7-ngon-ngu-lap-trinh-backend-pho-bien-nhat) thì website mới hoạt động trơn tru, hiệu quả để trải nghiệm người dùng được như ý.

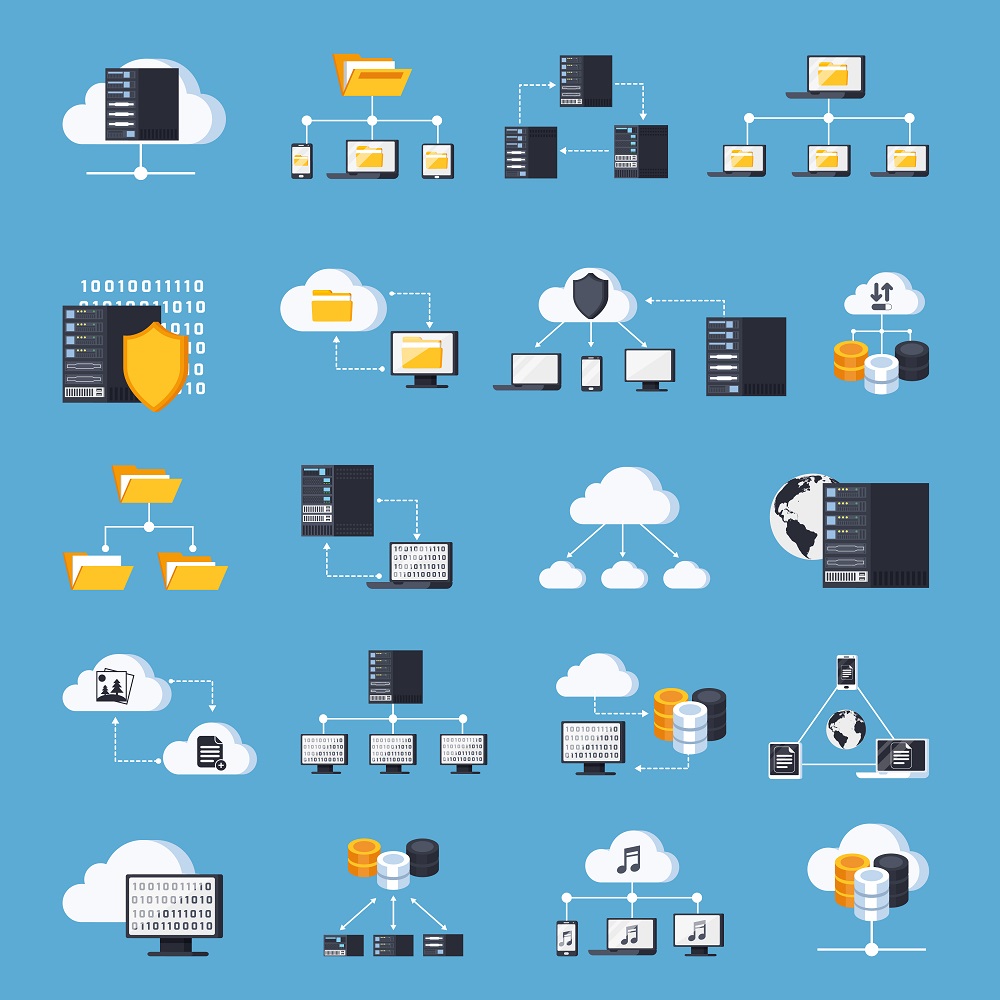


Hình ‑: Back end là gì?

#### Tại sao ta cần Back end?

**Backend là cầu nối thông tin giữa website và client**

Mỗi tương tác trên giao diện Google sẽ tạo ra những luồng data giữa trình duyệt và database của Google. Nhờ đó, bạn mới có thể tìm vị trí khách sạn, sản phẩm hay đăng ký tài khoản trên Google. Backend là nơi xử lý những luồng data này, giúp bạn tương tác với database của website.



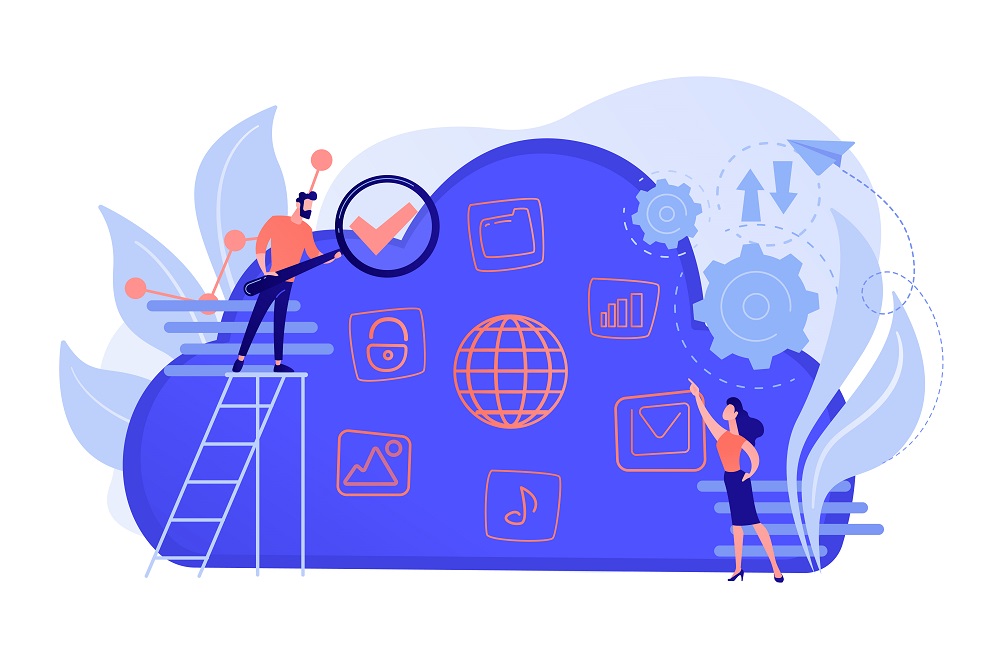
Hình ‑: Backend là cầu nối thông tin giữa website và client

**Backend giúp website chạy trên nhiều nền tảng lưu trữ**

Ngày nay, bạn có thể triển khai website trên những nền tảng lưu trữ (hosting) khác nhau. Ví dụ như máy chủ web (web server), máy chủ đám mây (cloud server) hoặc dùng cả hai. Vì mỗi hosting có những yêu cầu thiết lập riêng nên bạn phải tùy chỉnh backend tương ứng với chúng. Chưa kể bạn còn phải điều chỉnh khối lượng công việc backend khi dùng nhiều server khác nhau.

**Backend cung cấp các chức năng mượt mà cho website**

Trong kiến trúc website, những đoạn code về backend cung cấp các chức năng trên website cho người dùng. Kể cả thứ đơn giản như chức năng đăng nhập cũng cần có backend để hoạt động bình thường.



Hình ‑: Backend giúp các chức năng của website hoạt động mượt mà

Khi bạn điền thông tin và nhấn nút đăng nhập, backend sẽ lấy thông tin và kiểm tra với database. Chỉ cần backend lấy sai thông tin mà bạn điền thì coi như chức năng đăng nhập đã bị lỗi. Tương tự như vậy với các chức năng khác như giỏ hàng (shopping cart) hay tìm sản phẩm trên Shopee. Do đó, không có website nào có thể hoạt động trơn tru mà không có backend.

**Tổng quan về kiến trúc backend**

Như đã nói ở trên thì kiến trúc backend có ba phần chính là server, application và database. Không chỉ vậy, backend còn có một thứ không thể thiếu là giao diện lập trình API (Application Programming Interface).

**Server: cỗ máy xử lý của backend**

Trong ba phần chính, có thể coi server chính là trung tâm điều hành của backend. Khi bạn truy cập website, server sẽ gửi lại bạn với data về giao diện của website đó. Khi bạn tương tác trên website, server sẽ nhận request, thực hiện request và gửi lại response cho bạn. Server đơn giản là một hoặc nhiều máy tính được nối mạng, nhận request và gửi lại response tương ứng.



Hình ‑: Server chính là trung tâm điều hành của backend

Những máy tính này không cần chuột, bàn phím mà đã được chế tạo, tối ưu chỉ để làm server. Dù vậy, bất kỳ máy tính nào có kết nối mạng cũng đều có thể trở thành server. Trên thực tế, vì server khá đắt đỏ nên bạn thường phải dùng chính máy tính của mình làm server.

**Database: đầu não lưu trữ của backend**

Trong kiến trúc website, database là thứ tạo nên sự khác biệt giữa website động (dynamic) và tĩnh (static). Database lưu trữ tất cả data của website động để server có thể truy vấn lúc cần thiết. Việc này vừa giảm tải bộ nhớ server, vừa giảm thiểu rủi ro mất data khi server gặp sự cố.

Nói đơn giản thì database là nơi tập hợp và tổ chức data theo một mô hình nhất định. Để quản trị database, bạn cần phải dùng phần mềm quản trị cơ sở dữ liệu DBMS (Database Management System). Có nhiều database khác nhau và DBMS tương ứng nên hãy chọn lựa dựa theo nhu cầu của bạn nhé.

**Application: trợ lý công việc của backend**

Hãy tưởng tượng application như một hệ thống ống nước truyền dẫn request, data và response giữa website và database. Khi truy cập Google thì server sẽ chạy application có chứa logic về cách phản ứng với request của client. Đó là truy vấn và gửi đi data giao diện Google đến website để hiển thị website Google. Ngoài ra, application còn có thể chỉnh sửa database hoặc xử lý những request đặc biệt khác của client.

Thực ra, server có nhiều application và mỗi application có một tác vụ khác nhau. Trong quá trình xử lý request, các application sẽ chuyển quyền chạy cho nhau mỗi khi hoàn thành tác vụ. Cuối cùng, server sẽ chạy một application để kết thúc chu trình request-response bằng một response gửi lại client.

**API: mảnh ghép không thể thiếu của backend**

Đã lập trình backend thì phải làm việc với các giao diện lập trình API. Chúng là các phương thức, giao thức giúp kết nối nhiều ứng dụng website với nhau.



Hình ‑: API là giao thức giúp kết nối nhiều ứng dụng website với nhau khi lập trình backend

Với API, có thể dễ dàng kết nối phần mềm, CSDL và dịch vụ của mình hoặc của người khác lại với nhau mà không cần đến những đoạn code phức tạp. Ngoài ra, data và ứng dụng website của cũng có giá trị hơn khi được người khác tận dụng khi phát triển sản phẩm.

## Những thành phần chính của Front end

### HTML – “xương sống” của trang web

#### HTML là gì?

HTML, viết tắt từ Hyper Text Markup Language, là ngôn ngữ chính thức dùng trong protocol World Wide Web, được tạo ra vào năm 1990. HTML kết hợp các đoạn văn bản (text) với cấu trúc code để từ đó quy định cách các text đó thể hiện.

Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet). Các thiết lập và cấu trúc HTML được vận hành và phát triển bởi World Wide Web Consortium (W3C). Bạn có thể kiểm tra tình trạng mới nhất của ngôn ngữ này bất kỳ lúc nào trên trang [W3C’s website](https://www.w3.org/).

Cùng với sự phát triển nhanh chóng của World Wide Web, HTML dần trở thành một công cụ vô cùng hữu ích đối với các nhà phát triển browser. Nhiều HTML độc quyền được đăng ký bản quyền tạo ra, như tag <blink> hay <marquee> (proprietary tag) của Microsoft trong cuộc chiến đầu tiên giữa các trình duyệt.  Điều này tạo nên nhiều rắc rối khi người viết web nhận ra rằng web của mình thể hiện khác nhau ở các browser khác nhau do ảnh hưởng của các tag độc quyền.

Dần nhận thấy sự phát triển tràn lan của ngôn ngữ HTML và những rắc rối mà nó mang lại, tổ chức World Wide Web Consortium (W3C) đã đi đến thống nhất và phê duyệt bộ ngôn ngữ HTML chuẩn.  Tháng 9 năm 1995, bộ HTML 2.0 được ra đời với một số tính năng mới như tag hình ảnh <img>, hay sau đó là tag bảng biểu <table> ở version 3.0.

**HTML** được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes… và **HTML** không phải là ngôn ngữ lập trình.



Hình ‑: Logo HTML 5

#### Ưu điểm của HTML

* Ngôn ngữ được sử dụng rộng lớn này có rất nhiều nguồn tài nguyên hỗ trợ và cộng đồng sử dụng cực lớn.
* Sử dụng mượt mà trên hầu hết mọi trình duyệt.
* Có quá trình học đơn giản và trực tiếp.
* Mã nguồn mở và hoàn toàn miễn phí.
* Markup gọn gàng và đồng nhất.
* Chuẩn chính của web được vận hành bởi World Wide Web Consortium (W3C).
* Dễ dàng tích hợp với các ngôn ngữ backend như PHP và Node.js.

#### Nhược điểm của HTML

* Được dùng chủ yếu cho web tĩnh. Đối với các tính năng động, bạn cần sử dụng JavaScript hoặc ngôn ngữ backend bên thứ 3 như PHP.
* Nó có thể thực thi một số logic nhất định cho người dùng. Vì vậy, hầu hết các trang đều cần được tạo riêng biệt, kể cả khi nó sử dụng cùng các yếu tố, như là headers hay footers.
* Một số trình duyệt chậm hỗ trợ tính năng mới.
* Khó kiểm soát cảnh thực thi của trình duyệt (ví dụ: những trình duyệt cũ không render được tag mới).

### CSS – kiểu dáng của trang web

#### CSS là gì?

CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu ([HTML](https://topdev.vn/blog/html-la-gi/)). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

CSS được phát triển bởi W3C ([World Wide Web Consortium](https://www.w3.org/)) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

#### Ưu điểm của CSS

* Tiết kiệm thời gian, công sức trong việc code web/ xây dựng giao diện; kiểm soát dễ dàng các lỗi không đáng có trên web
* CSS cung cấp nhiều thuộc tính cụ thể hơn HTML giúp tạo ra nhiều style khác nhau và có thể áp dụng được cho nhiều web.
* CSS giúp tốc độ load trang web nhanh chóng hơn, dễ dàng bảo trì, tương thích với các thiết bị di động.
* CSS thiết lập nhiều kiểu chữ khác nhau giúp website của bạn trở nên khác biệt hơn; tự động sắp xếp chữ khi chèn hình ảnh (thuộc tính shape-outsite); đặt nội dung theo các chiều khác nhau/ thay đổi chế độ viết theo mong muốn của mình (thuộc tính write-mode); có thể tạo chữ lớn ở đầu dòng (thuộc tính first-letter)…



Hình ‑: Logo CSS 3

#### Nhược điểm của CSS

* CSS có nhiều cấp độ, bởi vậy dễ gây nhầm lẫn cho người dùng mới, ít am hiểu về lập trình
* Với mỗi trình duyệt khác nhau, CSS sẽ hoạt động khác nhau.

### JavaScript – yếu tố động và tương tác của trang web

#### JavaScript là gì?

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa. Javascript được sử dụng rộng rãi trong các ứng dụng Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... trên máy tính lẫn điện thoại.

Nhiệm vụ của Javascript là xử lý những đối tượng HTML trên trình duyệt. Nó có thể can thiệp với các hành động như thêm / xóa / sửa các thuộc tính CSS và các thẻ HTML một cách dễ dàng. Hay nói cách khác, Javascript là một ngôn ngữ lập trình trên trình duyệt ở phía client. Tuy nhiên, hiện nay với sự xuất hiện của NodeJS đã giúp cho Javascript có thể làm việc ở backend.

ExpressJS (NodeJS framework), và nhiều thư viện frontend khác như Angular, jQuery, RactJS ra đời, giúp tạo ra một cơn sốt với từ khóa Javascript Fullstack.

Sức mạnh của Javascript là không thể chối cãi, bằng chứng là hiện nay có rất nhiều libraries và framework được viết bằng Javascript ra đời như:

* **Angular**: Một thư viện dùng để xây dựng ứng dụng Single Page.
* **NodeJS**: Một thư viện được phát triển phía Server dùng để xây dựng ứng dụng realtime.
* **Sencha Touch**: Một Framework dùng để xây dựng ứng dụng Mobile.
* **ExtJS**: Một Framework dùng xây dựng ứng dụng quản lý (Web Applications).
* **jQuery**: Một thư viện rất mạnh về hiểu ứng.
* **ReactJS**: Một thư viện viết ứng dụng mobie.
* Và còn nhiều thư viện khác.



Hình ‑: HTML, CSS và JS là bộ 3 không thể thiếu trong mảng Front end

#### Ưu điểm của JavaScript

* Bạn không cần compiler vì web browser có thể biên dịch nó bằng HTML
* Nó dễ dàng học hơn các ngôn ngữ lập trình khác
* Lỗi dễ phát hiện hơn nên dễ sửa hơn;
* Nó có thể được gắn trên một số element hay event của trang như là thông qua click chuột hoặc di chuột tới;
* JavaScript hoạt động trên nhiều trình duyệt, nền tảng khác nhau
* Bạn cũng có thể dùng JavaScript để kiểm tra input làm giảm thiểu việc kiểm tra thủ công khi truy xuất qua database
* Nó giúp website tương tác tốt hơn với khách hàng truy cập;
* Nó nhanh hơn, nhẹ hơn so với các ngôn ngữ lập trình khác.

#### Nhược điểm của JavaScript

* Dễ bị khai thác;
* Có thể dùng để thực thi mã độc trên máy tính của người dùng;
* Nhiều khi không được hỗ trợ trên mọi trình duyệt;
* JavaScript code snippets lớn;
* Có thể triển khai khác nhau tùy vào từng thiết bị dẫn đến việc không đồng nhất;

### TypeScript – ngôn ngữ nâng cấp của JavaScript

#### TypeScript là gì?

Typescript hiểu đơn giản chính là một ngôn ngữ được Microsoft cung cấp hoàn toàn miễn phí cho chúng ta. Nghe đến Typescript thì bạn đang mường tượng đến Javascript phải không? Nền tảng của TypeScript ít nhiều cũng có sự liên quan đến JavaScript vì nó chính là một ngôn ngữ mã nguồn mở của JavaScript. Type script đóng vai trò là dùng để thiết kế và xây dựng các dự án ứng dụng quy mô lớn mang tính chất phức tạp.

Nhiều bạn hiểu theo “Typescript là phiên bản cao cấp của Javascript” thì cũng đúng, vì nó được kế thừa hầu như tất cả những gì của Javascript mà “bonus” thêm một số chức năng tiện lợi hơn, cải tiến hơn từ những yếu điểm của javascript như: các lớp hướng đối tượng và Static Structural typing, bên cạnh đó TypeScript còn có thể hoạt động rộng rãi cho các ứng dụng của ngôn ngữ Angular2 và Nodejs,…

#### Ưu điểm của TypeScript

* TypeScript thật sự thuận tiện và hoàn toàn miễn phí
* Thao tác nhanh chóng và đơn giản hơn
* Tái cấu trúc
* Giảm tỷ lệ mắc lỗi trong hệ thống
* Hạn chế thử nghiệm Boilerplate
* Hợp nhất mã đơn giản
* Hỗ trợ tối ưu hóa quy trình làm việc

#### Nhược điểm của TypeScript

* Bắt buộc dùng biên dịch
* Bước thiết lập cồng kềnh
* Chỉ là phần ngôn ngữ mở rộng hỗ trợ
* TypeScript không chắc sẽ là một sản phẩm nguồn mở vĩnh cửu, điều đó tùy thuộc vào Microsoft – cha đẻ của nó

## Những thành phần chính của Back end

### T-SQL – ngôn ngữ truy vấn database

Transact-SQL (còn gọi là T-SQL) là một ngôn ngữ lập trình database hướng thủ tục độc quyền của Microsoft sử dụng trong SQL Server chuẩn của ISO (International Organization for Standardization) và ANSI (American National Standards Institute) được sử dụng trong SQL Server khác với P-SQL (Procedural-SQL) dùng trong Oracle.

SQL chuẩn bao gồm khoảng 40 câu lệnh. Trong các hệ quản trị cơ sở dữ liệu khác nhau, mặc dù các câu lệnh đều có cùng dạng và cùng mục đích sử dụng song mỗi một hệ quản trị cơ sở dữ liệu có thể có một số thay đổi nào đó. Điều này đôi khi dẫn đến cú pháp chi tiết của các câu lệnh có thể sẽ khác nhau trong các hệ quản trị cơ sở dữ liệu khác nhau.

Ngôn ngữ thủ tục được thiết kế để mở rộng khả năng của SQL trong khi có khả năng tích hợp tốt với SQL. Một số tính năng như các biến địa phương và xử lý chuỗi/dữ liệu được thêm vào. Các tính năng này làm cho ngôn ngữ Transact-SQL là Turing-complete (\*\*).

Chúng cũng được sử dụng để viết các thủ tục lưu trữ: Một đoạn code nằm trên máy chủ để quản lý các quy tắc kinh doanh phức tạp mà khó hoặc không thể làm nổi với các thao tác tập hợp thông thường (pure set-based operations).

T-SQL được chia làm ba nhóm chính như sau:

* Nhóm ngôn ngữ định nghĩa dữ liệu: Là tập hợp những lệnh dùng để xử lý dữ liệu như các lệnh như create, alter và drop table, các lệnh này dùng để định nghĩa cấu trúc của một table.
* Nhóm điều khiển dữ liệu: Dùng để phân quyền user truy cập vào các đối tượng table, view, function ... Nhóm này là các câu truy vấn như Grant, Deny, Revoke.
* Nhóm thao tác dữ liệu: là tập hợp các câu lệnh dùng để thao tác sàn lọc dữ liệu như: Select, Insert, Update, Delêt. Những lệnh này giúp chúng ta truy  vấn tìm kiếm dữ liệu một cách chính xác nhất.

### Java – ngôn ngữ mạnh mẽ và đa nền tảng

#### Java là gì?

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

Java được tạo ra với tiêu chí “*Viết (code) một lần, thực thi khắp nơi*” (Write Once, Run Anywhere  – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

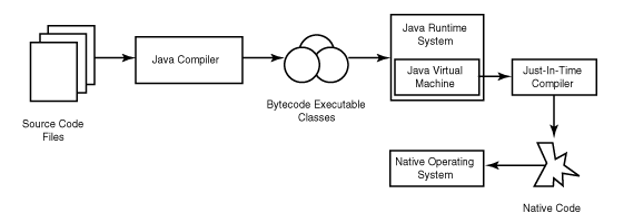


Hình ‑: Logo Java

#### Đặc điểm của ngôn ngữ Java

* Tương tự như C++, hướng đối tượng hoàn toàn(Object Oriented).
* Độc lập phần cứng và hệ điều hành(Encapsulation).
* Độc lập phần cứng và hệ điều hành(Platform Independent).
* Ngôn ngữ thông dịch(Interpreted Language).

Ngôn ngữ lập trình thường được chia ra làm 2 loại (tùy theo các hiện thực hóa ngôn ngữ đó) là ngôn ngữ thông dịch và ngôn ngữ biên dịch.



Hình ‑: Thông dịch trong Java

* Thông dịch (Interpreter) : Nó dịch từng lệnh rồi chạy từng lệnh, lần sau muốn chạy lại thì phải dịch lại.
* Biên dịch (Compiler): Code sau khi được biên dịch sẽ tạo ra 1 file thường là .exe, và file .exe này có thể đem sử dụng lại không cần biên dịch nữa.

Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ vừa biên dịch vừa thông dịch. Cụ thể như sau:

* Cơ chế thu gom rác tự động(Garbage collection automatically).
* Đa luồng(Multithreaded).
* Tính an toàn và bảo mật(Backward Compatibility).
* Tính an toàn và bảo mật(Secure).

#### Ưu điểm của Java

* Java là nền tảng độc lập vì chúng ta có thể chạy mã Java trên bất kỳ máy nào mà không cần cài đặt bất kỳ phần mềm đặc biệt nào, JVM thực hiện điều đó.
* Java là hướng đối tượng vì các lớp và đối tượng của nó.
* Lý do chính để Java được bảo mật là con trỏ, Java không sử dụng con trỏ.
* Trong Java, chúng ta có thể thực thi nhiều chương trình đồng thời, do đó, có thể đạt được đa luồng.
* Java mạnh mẽ vì nó có nhiều tính năng như thu gom rác, không sử dụng con trỏ rõ ràng, xử lý ngoại lệ.
* Java là một ngôn ngữ cấp cao giúp nó dễ hiểu.
* Quản lý bộ nhớ hiệu quả được thực hiện bằng Ngôn ngữ lập trình Java.

#### Nhược điểm của Java

* Là một ngôn ngữ cấp cao, nó phải xử lý các cấp độ biên dịch và trừu tượng của một máy ảo. Java thể hiện hiệu suất kém, nguyên nhân chính là do bộ thu gom rác, cấu hình bộ nhớ đệm không hợp lệ và bế tắc giữa các quy trình.
* Java có rất ít trình tạo GUI – Swing, SWT, JSF và JavaFX trong số những trình xây dựng phổ biến hơn.
* Để viết mã để thực hiện một tập hợp các hoạt động đơn giản, bạn có thể phải viết những đoạn mã dài và phức tạp. Điều này có thể ảnh hưởng đến khả năng đọc nhưng đảm bảo rằng các lập trình viên nhập chính xác những gì cần phải làm. Khi chúng tôi so sánh Java với Python , chúng tôi nhận thấy Python không cần dấu chấm phẩy, dấu ngoặc đơn hoặc dấu ngoặc nhọn và có mã ngắn hơn rõ ràng.

#### Ứng dụng của ngôn ngữ lập trình Java

Java phổ biến rộng rãi, sau đây là một số lĩnh vực mà java có thể sử dụng được:

* Ứng dụng máy tính để bàn
* Ứng dụng web
* Ứng dụng di động (Android)
* Điện toán đám mây
* Ứng dụng doanh nghiệp
* Ứng dụng khoa học
* Các hệ điều hành
* Những hệ thống nhúng
* Phần mềm thời gian thực
* Mật mã học
* Nhưng chiêc thẻ thông minh
* Trò chơi máy tính
* Máy chủ web và máy chủ ứng dụng

# : CƠ SỞ LÝ THUYẾT LIÊN QUAN

## Visual Studio Code (VS Code)

### VS Code là gì?

Visual Studio Code là công cụ soạn thảo mã nguồn do Microsoft phát triển. Nó được giới thiệu lần đầu tiên vào năm 2015 và chính thức được phát hành vào năm 2016.



Hình ‑: Logo VS Code

Visual Studio Code (VS Code) có dung lượng khá nhẹ và khả năng vận hành mạnh mẽ trên cả 3 nền tảng được phát triển bởi Microsoft là: macOS, Linux và Windows. Đặc biệt, VS Code là một mã nguồn mở và hoàn toàn miễn phí. Vì vậy, nó trở thành cái tên quen thuộc của nhiều lập trình viên.

Có thể nói, Visual Studio Code là sự kết hợp cực kỳ hoàn hảo giữa IDE và Code Editor. Nó hỗ trợ cho người dùng rất nhiều tiện ích như: đổi theme, hỗ trợ Git, cải tiến mã nguồn, có syntax highlighting, hỗ trợ cho quá trình gõ code, sử dụng các phím tắt và nhiều tùy chọn khác nhau,…

Hiện nay, Visual Studio Code chiếm ưu thế ở hầu hết các môi trường phát triển dành cho lập trình viên. Trong một khảo sát của Stack Overflow (năm 2019), Visual Studio Code được đánh giá là môi trường phát triển được dùng phổ biến nhất với hơn 50% lượt bình chọn trong tổng số hơn 90 nghìn người dùng tham gia khảo sát. Trong khi đó, con số này của năm 2018 chỉ có 35%. Điều này cho thấy độ phủ sóng ngày càng rộng rãi và sự hữu ích mà VSCode mang lại cho lập trình viên.

### Những tính năng nổi bật của VS Code

* **Intellisense**

Intellisense là sự kết hợp giữa trí tuệ nhân tạo và code auto-complete. Nó sẽ cung cấp hàng loạt các đề nghị, gợi ý và mô tả ngắn khi chúng ta đang viết code. Những gợi ý sẽ được tính toán dựa trên nhiều nhân tố như: cú pháp, ngôn ngữ lập trình, hàm, biến,…

* **Tích hợp sẵn Git**

Git trên Visual Studio Code sẽ cung cấp cho bạn tất cả các git action cơ bản như: push, commit code,… Đặc biệt, qua từng phiên bản khác nhau thì tính năng hỗ trợ Git ngày càng đầy đủ và tiện ích.

* **Debugger**

Hỗ trợ debug là một tính năng hữu ích của VSCode. Theo mặc định, VSCode chỉ đính kèm chương trình Debug hỗ trợ riêng cho Node JS. Tuy nhiên, bạn hoàn toàn có thể cài đặt thêm extension để có thể debug cho những ngôn ngữ khác theo nhu cầu sử dụng.

* **Tích hợp Terminal**

Terminal là chương trình giao diện cửa sổ dòng lệnh. Đây là một tính năng vô cùng quan trọng với lập trình viên. Khi sử dụng VSCode, bạn có thể mở một hoặc có thể mở nhiều tab terminal tại thư mục bạn đang làm việc, giúp nâng cao hiệu quả công việc.

* **Khả năng mở rộng và tùy chỉnh**

Visual Studio Code cung cấp cho người dùng khả năng tùy chỉnh tuyệt vời như: tùy chỉnh theme, kích thước, font chữ, keyboard shortcut, coding style, tùy chỉnh tính năng,… một cách linh hoạt. Bên cạnh đó, bạn cũng có thể tùy chỉnh trên từng workshop vô cùng tiện lợi.

* **Code Spell Checker**

Tính năng này cho phép người dùng kiểm tra tên hàm, các comment, tên biến sai chính tả một cách nhanh chóng. Từ đó giúp bạn có thể chỉnh sửa dễ dàng và nhanh chóng hơn.

* **Live Server**

Xây dựng server chạy CSS/HTML/JS thông qua tính năng Live Server. Tính năng này sẽ tự động refresh lại trình duyệt mỗi khi bạn chỉnh sửa code, giúp tiết kiệm rất nhiều thời gian.

Ngoài ra, VSCode còn cung cấp rất nhiều tính năng hữu ích khác.

### Ưu điểm của VS Code

* **Nhỏ gọn và tiện lợi**

VScode có dung lượng nhỏ nên không chiếm nhiều RAM. Đặc biệt, sự tích hợp giữa Code Editor và DIE giúp bạn có thể linh hoạt sử dụng mà không phải tốn quá nhiều thời gian như khi sử dụng các phần mềm hỗ trợ khác.

* **Tốc độ nhanh**

VSCode có tốc độ làm việc nhanh, cho phép người dùng có thể sử dụng nhiều tính năng cùng lúc như: đối sánh dấu ngoặc, đánh dấu cú pháp,… Đồng thời, bạn cũng có thể xây dựng, chỉnh sửa và gỡ lỗi một cách đơn giản, dễ dàng.

* **Hỗ trợ đa ngôn ngữ lập trình**

VSCode hỗ trợ nhiều ngôn ngữ lập trình khác nhau như: C#, C/C+, HTML, Visual Basic, F#, JavaScript, CSS,… Do vậy, nó có thể dễ dàng phát hiện và đưa ra thông báo khi chương trình xuất hiện lỗi.

* **Kho lưu trữ an toàn**

Đây là một trong những ưu điểm nổi bật nhất của VSCode. Nó có thể dễ dàng kết nối với Git hay bất kỳ kho lưu trữ nào hiện có. Vậy nên, bạn hoàn toàn có thể yên tâm khi sử dụng.

* **Hỗ trợ web**

VSCode hỗ trợ nhiều ứng dụng web khác nhau. Đồng thời, nó cũng có 1 trình soạn thảo và thiết kế website.

* **Hỗ trợ đa nền tảng**

Nếu bạn lo lắng các chương trình viết code thông thường chỉ có thể sử dụng cho hoặc Linux hoặc Windows hoặc Systems thì vấn đề này sẽ được giải quyết nhanh chóng bằng VSCode. Bởi mã nguồn mở này có thể hoạt động tốt trên cả 3 nền tảng đó.

* **Lưu trữ dữ liệu theo dạng phân cấp**

Những tệp lưu trữ đoạn mã đều được đặt trong những thư mục tương tự nhau. Bên cạnh đó, VSCode còn cung cấp các thư mục dành riêng cho một số tệp quan trọng khác.

* **Hỗ trợ viết code**

Một số đoạn code có thể sẽ được thay đổi nhằm mang lại sự thuận tiện cho người dùng. Lúc đó, Visual Studio Code sẽ đề xuất các tùy chọn thay thế (nếu có) cho bạn.

* **Hỗ trợ thiết bị đầu cuối**

VSCode tích hợp các thiết bị đầu cuối. Điều này giúp người dùng sẽ không phải chuyển đổi giữa hai màn hình hay trở về thư mục gốc để thực hiện các thao tác khác.

* **Giao diện dễ sử dụng, thân thiện với người dùng**

Hệ thống các phím tắt đa dạng giúp bạn dễ dàng quan sát cũng như thao tác làm việc nhanh chóng hơn. Từ đó giúp nâng cao hiệu suất và tăng hiệu quả làm việc.

* **Màn hình đa nhiệm**

Visual Studio Code cho phép người dùng có thể mở nhiều tệp và thư mục cùng lúc. Mặc dù giữ chúng không có sự liên quan với nhau.

* **Sự hỗ trợ từ cộng đồng GitHub**

VSCode là một mã nguồn mở. Do vậy, bạn có thể tham gia đóng góp cũng như tìm kiếm sự trợ giúp từ tất cả các thành viên trong cộng đồng mỗi khi có câu hỏi thắc mắc hay gặp khó khăn khi sử dụng.

### Nhược điểm của VS Code

* Tương tự như tất cả các ứng dụng electron khác, bộ nhớ và mức sử dụng pin của VSCode khá tệ.
* Không có git merge, điều mà nhiều người mong đợi, vì ST3 và Atom có ​​khả năng làm điều đó.
* Các phím tắt mặc định không có ý nghĩa và người dùng phải cấu hình lại gần như tất cả chúng.

## Angular

### Angular là gì?

Angular được xe là một open source (mã nguồn mở) hay frameworks miễn phí chuyên dụng cho công việc thiết kế web. Angular được phát triển từ những năm 2009 và được duy trì bởi Google. Frameworks này được xem là frameworks front end mạnh mẽ nhất chuyên dụng bởi các lập trình viên cắt HTML cao cấp.

Angular được ứng dụng rộng rãi với mục đích xây dựng project Single Page Application (SPA). Hiện tại, Version stable của Angular là Angular 9 (released on February 7, 2020) với TypeScript 3.6 và 3.7.

Trên thực tế, Angular sở hữu hai phiên bản hoàn toàn khác nhau như sau:

Phiên bản 1: Phiên bản này có tên chính thức là AngularJS đến hiện nay nó vẫn đang được phát triển hoàn thiện bởi các chuyên gia. Vào năm 2018, thì phiên bản 1.7.3 đã được nhà sản xuất phát hành.

Đây là phiên bản được hoàn thành thực sự dựa trên Javascript bởi những lập trình viên viết mã theo ngôn ngữ Javascript thì nó thường dễ làm và dễ tìm kiếm Tuy nhiên, hiệu năng khi so sánh với phiên bản ReactJS thì vẫn còn nhiều công ty xây dựng phải cân nhắc sử dụng Angular với phiên bản khác 1.

Phiên bản 2: Phiên bản này được gọi với cái tên chính thức là Angular. Nó sở hữu tư duy thay đổi hoàn toàn với phiên bản thứ 1 và mang lại tham vọng giúp cho Angular có thể đánh bại ReactJS. AngularJS được xem là một trong những framework sở hữu cấu  trúc ứng dụng dạng động và cho phép bạn có thể sử dụng HTML như là ngôn ngữ mẫu cũng như mở việc mở rộng cú pháp của HTML.

Nhờ vậy, việc diễn đạt các thành phần ứng dụng của người lập trình sẽ trở nên súc tích và rõ ràng hơn rất nhiều.Bạn có thể tham khảo thêm: 2 tính năng nổi bật là Data binding và Dependency injection của AngularJS có thể loại bỏ phần lớn code mà người lập trình thường xuyên phải viết. Điều này xảy ra trong mọi trình duyệt và giúp nó trở thành đối tác lý tưởng của mọi công nghệ Server.



Hình ‑: Logo Angular

### Các đặc trưng cơ bản của Angular

* Được sử dụng để có thể phát triển dựa trên JavaScript.
* Angular có khả năng tạo các ứng dụng client-side dựa trên mô hình MVC.
* Angular sở hữu khả năng tương thích cao có thể tự động xử lý dễ dàng các mã Javascript sao cho phù hợp với các trình duyệt nhất.
* Khi có mã nguồn mở và miễn phí thì nó sẽ được sử dụng rộng rãi hơn.

Thông thường, kiến trúc của một ứng dụng Angular là dựa trên những ý tưởng liên quan đến Components. Mỗi một ứng dụng Angular thường bắt đầu với những  level trên cùng tên gọi là Root Component.

### Ứng dụng của Angular

* CRUD Web Apps: đây là một trong những tính năng mà chính người sử dụng Angular ưu tiên sử dụng nhiều nhất.
* Mobile Apps: Lập trình viên có thể sử dụng Angular kết hợp cùng Phonegap để tạo ra một loại Mobile App  bạn có thể tạo một Mobile Web App.
* CSS3 Animations.
* Chrome Extensions: Đây là cách đơn giản nhất giúp lập trình viên tạo ra Chrome Extension lả việc sử dụng Yeoman- một Chrome Extension Generator.
* Testable JS Apps.
* Firebase Powered Apps.

### Ưu điểm của Angular

* AngularJS được đánh giá là giải pháp dành cho các Single Page Application làm việc dễ dàng.
* Code Front end thường rất thân thiện nhờ khả năng Binding data lên trên các nền tảng HTML để có thể tạo ra mọi thao tác rất tuyệt.
* Bạn có thể dễ dàng Unit test
* Có thể tái sử dụng component dễ dàng hơn
* Hỗ trợ cho các lập trình viên có thể viết code được ít hơn với nhiều chức năng hơn.
* Bạn có thể chạy AngularJS trên nhiều loại trình duyệt khác nhau lẫn PC hoặc mobile.

### Nhược điểm của Angular

* Không an toàn: Thông thường, bản chất của AngularJS là một trong những các Front End, mà front end này thường vốn không thể bảo mật bằng [Back End](https://itnavi.com.vn/blog/nhung-ky-nang-quan-trong-de-tro-thanh-mot-backend-developer/). Chính vì vậy, khi sử dụng API thì bạn cần xây dựng cho một hệ thống kiểm tra dữ liệu sao cho việc trả về được tốt nhất.
* Với một số trình duyệt sở hữu tính năng Disable Javascript nên có nghĩa là website sẽ không hoàn toàn có thể sử dụng được dựa trên những trình duyệt đó nữa.

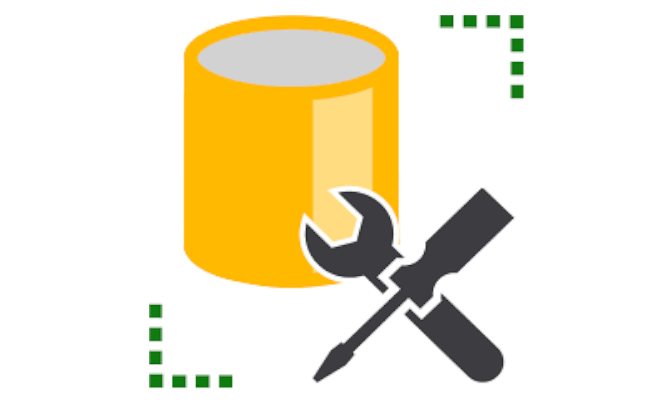
## SQL Server

### SQL Server là gì ?

SQL Server hay còn gọi là Microsoft SQL Server, viết tắt là MS SQL Server. Đây là một phần mềm được phát triển bởi Microsoft dùng để lưu trữ dữ liệu dựa trên chuẩn RDBMS, và nó cũng là một hệ quản trị [cơ sở](https://www.semtek.com.vn/co-so-du-lieu/) dữ liệu quan hệ đối tượng (ORDBMS).

SQL Server cung cấp đầy đủ công cụ để quản lý, từ [giao diện](https://www.semtek.com.vn/giao-dien-website/) GUI cho đến việc sử dụng ngôn ngữ truy vấn SQL. Ngoài ra điểm mạnh của nó là Microsoft có khá nhiền nền tảng kết hợp hoàn hảo với SQL Server như ASP.NET, C# xây dựng Winform, bởi vì nó hoạt động hoàn toàn độc lập.

Microsoft SQL Server Management Studio là một ứng dụng phần mềm thiết kế bởi Microsoft, ra mắt lần đầu năm 2005. Ứng dụng này cho phép lập trình viên cấu hình, quản lý và quản trị bộ máy cơ sở dữ liệu (database engine) SQL Server. SSMS phổ biến và được sử dụng rộng rãi trong cộng đồng lập trình viên và quản trị viên cơ sở dữ liệu. Ứng dụng này giúp ta kiểm tra hoặc tạo các thủ tục truy vấn nhằm áp dụng vào trong việc lập trình các chức năng của trang web.



Hình ‑: Logo SQL Server Management Studio

### Ứng dụng của SQL Server

* Tạo cơ sở dữ liệu.
* Duy trì cơ sở dữ liệu.
* Phân tích dữ liệu bằng SSAS – SQL Server Analysis Services.
* Tạo báo cáo bằng SSRS – SQL Server Reporting Services.
* Thực hiện quá trình ETL (Extract-Transform-Load) bằng SSIS – SQL Server Integration Services.

### Ưu điểm của SQL Server

* Có thể cài nhiều phiên bản MS SQL khác nhau trên cùng một máy tính.
* Duy trì riêng biệt các môi trường sản xuất, phát triển, thử nghiệm.
* Giảm thiểu các vấn đề tạm thời trên cơ sở dữ liệu.
* Tách biệt các đặc quyền bảo mật.
* Duy trì máy chủ dự phòng.

### Nhược điểm của SQL Server

* SQL Server chỉ chạy trên hệ điều hành Windows.
* Cần thanh toán phí license để chạy nhiều CSDL (database).

### Phân loại sql server

#### SQL DDL (Data Definition Language)

DDL là ngôn ngữ định nghĩa dữ liệu. Các lệnh DDL bao gồm:

* Lệnh CREATE: Tạo một bảng, một View của bảng, hoặc đối tượng khác trong Database.
* Lệnh ALTER: Sửa đổi một đối tượng Database đang tồn tại, ví dụ như một bảng.
* Lệnh DROP: Xóa toàn bộ một bảng, một View của bảng hoặc đối tượng khác trong một Database.

#### SQL DML (Data Manipulation Language)

DML là ngôn ngữ thao tác dữ liệu. Các lệnh DML bao gồm:

* Lệnh SELECT: Lấy các bản ghi cụ thể từ một hoặc nhiều bảng.
* Lệnh INSERT: Tạo một bản ghi.
* Lệnh UPDATE: Sửa đổi các bản ghi.
* Lệnh DELETE: Xóa các bản ghi.

#### SQL DCL (Data Control Language)

DCL là ngôn ngữ điều khiển dữ liệu. Các lệnh DCL bao gồm:

* Lệnh GRANT: Trao một quyền tới người dùng.
* Lệnh REVOKE: Thu hồi quyền đã trao cho người dùng.

## Eclipse

### Eclipse là gì?

Eclipse là 1 công cụ hỗ trợ lập trình mã nguồn mở được phát triển bởi IBM.

Eclipse như một môi trường phát triển Java tích hợp (IDE), với Eclipse chúng ta có thể mở rộng hơn mã nguồn bằng cách chèn thêm các plugins cho project (PDE- Plug-in Development Environment). Mặc dù Eclipse được viết bằng ngôn ngữ lập trình Java, nhưng việc sử dụng nó không hạn chế chỉ cho ngôn ngữ Java. Ví dụ, Eclipse hỗ trợ sẵn hoặc có thể cài thêm các plugins để hỗ trợ cho các ngôn ngữ lập trình như C/C + + và COBOL. Ngoài ra, còn rất nhiều ngôn ngữ khác như PHP, Groovy, ...

Eclipse còn hỗ trợ cho lập trình viên code theo các mô hình phát triển như MVC, tạo thêm các lib hỗ trợ phát triển phần mềm.



Hình ‑: Logo Eclipse

### Ưu nhược điểm của Eclipse

#### Ưu điểm của Eclipse

* Tạo thuận lợi cho tích hợp liền mạch các công cụ bên trong mỗi một và xuyên qua nhiều kiểu nội dung và các nhà cung cấp công cụ khác nhau.
* Hỗ trợ việc xây dựng nhiều công cụ.
* Hỗ trợ một tập hợp không hạn chế các nhà cung cấp công cụ, bao gồm cả các nhà cung cấp phần mềm độc lập (ISV).
* Hỗ trợ các công cụ thao tác các kiểu nội dung bất kỳ (bao gồm cả HTML, Java, C, JSP, EJB, XML, và GIF).
* Hỗ trợ cả môi trường phát triển ứng dụng GUI lẫn không dựa trên GUI.
* Chạy trên nhiều hệ điều hành, bao gồm cả Windows và Linux.
* Lợi dụng tính phổ biến của ngôn ngữ Java để viết các công cụ.
* Do sử dụng SWT/JFace nên có lẽ load nhanh hơn

#### Nhược điểm của Eclipse

* Cài đặt phức tạp
* Tốn phần cứng máy Eclipse cũng khá tốn bộ nhớ máy để cài đặt nên hiện nay eclipse đã khắc phục bằng phiên bản Eclipse Luna, không cần cái đặt
* Tốn bộ nhớ máy: chạy eclipse cũng khá tốn bộ nhớ máy tính nên để sử dụng nó máy của bạn cũng cần có cấu hình tương đối cao
* Thiếu tính nhất quán vì có nhiều plugins quá

### Chức năng nổi bật của Eclipse là gì?

Là một nền tảng mã nguồn mở, Eclipse chứa các các khung ứng dụng để xây dựng phần mềm và các dự án tập trung vào việc cung cấp một nền tảng phát triển mở rộng, Eclipse có những chức năng quan trọng và nổi bật mà bạn đọc cần nên biết như:

* Hỗ trợ Export và Import Project ngoài việc tạo mới Project
* Có hai chế độ Run Project: Debug và Run
* Chứa nhiều ngôn ngữ
* Có chứa thêm nhiều Plugins và thư viện mới

### Công nghệ Servlet

#### Servlet là gì?

Servlet có thể được mô tả bằng nhiều cách, tùy thuộc vào ngữ cảnh:

* Servlet là một công nghệ được sử dụng để tạo ra ứng dụng web.
* Servlet là một API cung cấp các interface và lớp bao gồm các tài liệu.
* Servlet là một thành phần web được triển khai trên máy chủ để tạo ra trang web động.

Có nhiều interface và các lớp trong API servlet như Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, ...

#### Nhiệm vụ của Servlet

* Đọc dữ liệu rõ ràng do khách hàng (trình duyệt) gửi. Điều này bao gồm một mẫu HTML trên một trang Web hoặc nó cũng có thể đến từ một applet hoặc một chương trình khách hàng HTTP tùy chỉnh.
* Đọc dữ liệu yêu cầu HTTP ẩn được gửi bởi khách hàng (trình duyệt). Điều này bao gồm các cookie, loại phương tiện truyền thông và các chương trình nén mà trình duyệt hiểu được, v.v.
* Xử lý dữ liệu và tạo ra các kết quả. Quá trình này có thể yêu cầu nói chuyện với một cơ sở dữ liệu, thực hiện một cuộc gọi RMI hoặc CORBA, gọi một dịch vụ Web, hoặc tính trực tiếp phản hồi.
* Gửi dữ liệu rõ ràng (tức là tài liệu) tới khách hàng (trình duyệt). Tài liệu này có thể được gửi bằng nhiều định dạng, bao gồm văn bản (HTML hoặc XML), nhị phân (hình ảnh GIF), Excel, v.v ...
* Gửi phản hồi HTTP ẩn cho khách hàng (trình duyệt). Điều này bao gồm nói với trình duyệt hoặc các trình khách khác loại tài liệu đang được trả về (ví dụ, HTML), thiết lập cookie và các tham số bộ nhớ đệm, và các tác vụ khác.

### Spring

#### Spring là gì?

Spring được xem là một Framework có vai trò phát triển cho các ứng dụng Java. Trong số đó, phổ biến nhất là Java Enterprise và nó được sử dụng bởi hàng triệu lập trình viên khác nhau. Framework này được phát triển đầu tiên bởi Rod Johnson và được ban hành giấy phép Apache 2.0.

Spring có kích thước nhẹ và trong suốt nên có thể hoạt động một cách trong suốt nhất với lập trình viên. Spring mặc dù nhẹ nhưng lại có khả năng hỗ trợ cho các ứng dụng tạo ra hiệu năng cao, sử dụng lại code dễ dàng và dễ kiểm thử,... Nhờ vậy mà các lập trình viên có thể dễ dàng thực hiện việc tối ưu hóa được hầu hết mọi công việc cũng như tiết kiệm được tối đa thời gian làm việc.



Hình ‑: Logo Spring

#### Tính năng của Spring

Các tính năng chính của Spring chủ yếu được sử dụng để có thể xây dựng được các ứng dụng bên trong Java Desktop, một số ứng dụng mobile, các Java Web. Một trong những mục tiêu chính nhất của Spring là đảm bảo cho việc phát triển những ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object).

#### Spring MVC

**Spring MVC là gì ?**

Spring MVC là một trong những mô hình ứng dụng mà ở đó các thành phần được phân tách thành nhiều lớp riêng biệt và mỗi cái có các nhiệm vụ đặc trưng. Do nhu cầu xây dựng các ứng dụng trên nền tảng Web được nhanh chóng và tiện lợi nên Spring MVC đã ra đời giúp việc triển khai các dự án nhanh hơn.Spring MVC được xây dựng theo cấu trúc MVC (Model-View-Controller) ở đó có các API cho phép việc xây dựng các ứng dụng trên web được dễ dàng và có quy tắc hơn.

**Cấu trúc của Spring MVC**



Hình 2‑6: Cấu trúc Spring MVC và các thành phần chứa bên trong Model – View – Controller.

Model là lớp chứa thông tin của các đối tượng (Dữ liệu), tương tác với Database. Trong Spring MVC nó gồm các file POJO, Service, DAO thực hiện truy cập database, xử lý business.

View là nơi hiển thị nội dung và tương tác trực tiếp với người dùng. Trên thực tế khi làm việc với Spring thì trong phần view người ra sẽ sử dụng các template engine mã nguồn mở cho phép chúng ta sử dụng lại các thành phần chung giữa các trang trong ứng dụng web và xử lý nhiều loại ngôn ngữ khác như HTML, XML, Javascript, [CSS](https://itnavi.com.vn/blog/css-la-gi/).

Controller là nơi tiếp nhận những request từ phía người dùng sau đó xử lý request để xây dựng dữ liệu cho view (model) và chọn view để trả lại kết quả của cho người dùng. Cụ thể ở đây là Dispatcher Controller, Handler Mapping, Controller.Thành phần chính của controller trong Spring MVC là:

* Một annotation @Controller được khai báo cùng với định nghĩa của lớp HomeController và một annotation @RequestMapping được khai báo cùng với định nghĩa của phương thức home().
* Phương thức home() trong lớp HomeController đang trả về một chuỗi “home”, chuỗi này định nghĩa tên view sẽ được dùng để hiển thị kết quả request cho người dùng. Biến model lưu trữ những dữ liệu cần thiết cho view “home” sử dụng và biến locale của đối tượng Locale để lấy dữ liệu về ngày giờ của server đang chạy web của chúng ta.

#### Spring Boot

**Spring Boot là gì?**

Spring Boot là một Spring framework hiện đang được nhiều lập trình viên lựa chọn vì nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng. Spring Boot được coi như là điểm khởi đầu để xây dựng tất cả các ứng dụng dựa trên nền tảng Spring.

**Ứng dụng của Spring Boot**

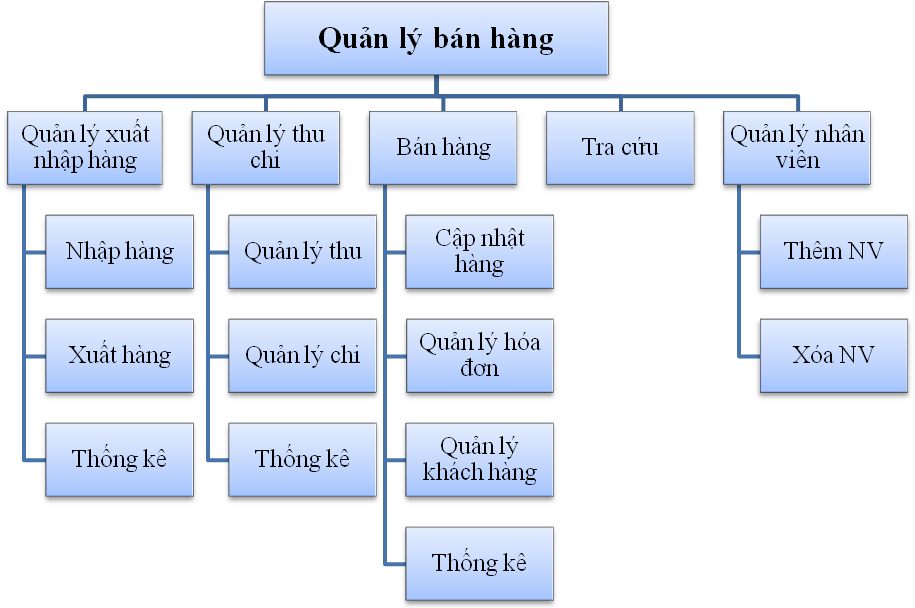
Bất cứ thứ gì sinh ra và ngày càng phát triển đều có lý do của nó, không phải tự nhiên mà Spring Boot lại được dùng để tạo các dự án Spring:

* Giúp xây dựng ứng dụng Spring độc lập (stand-alone) bao gồm gần như mọi thứ REST API, WebSocket, Web, Streaming …
* Giúp giảm tối ưu công đoạn cấu hình và không yêu cầu cấu hình thông qua XML.
* Có thể nhúng trực tiếp Tomcat, Jetty, Undertow… do đó không cần phải triển khai tập tin WAR.
* Tích hợp các loại mô-đun khác nhau trong các loaik Spring như Spring Data, Spring MVC, Spring Security, Spring JDBC, Spring ORM…
* Các starter dependency làm cho việc thêm các hỗ trợ phù hợp cho project giúp cấu hình ban đầu nhanh hơn.
* Bạn có thể dễ dàng phát triển các ứng dụng Spring dựa trên các loại ngôn ngữ khác như Java hoặc Groovy.
* Spring Boot có thể ự động cấu hình Spring mỗi khi cần thiết.
* Hỗ trợ nhiều IDE như Spring Tool Suite, IntelliJ IDEA, NetBeans hoặc cũng có thể cấu hình đơn giản và dùng Sublime Text để phát triển luôn.
* Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML …

# : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Phân tích hệ thống về chức năng

### Biểu đồ phân cấp chức năng



Hình ‑: Biểu đồ phân cấp chức năng

### Đặc tả chức năng của hệ thống

#### Chức năng đối với khách hàng:

* Chức năng đăng ký tài khoản khách hàng
* Chức năng đăng nhập hệ thống
* Chức năng thoát khỏi hệ thống
* Chức năng sửa thông tin thành viên
* Chức năng tìm kiếm sản phẩm
* Chức năng thêm hàng vào giỏ hàng
* Chức năng xem giỏ hàng

Chức năng xóa sản phẩm trong giỏ hàng

Chức năng cập nhật giỏ hàng

* Chức năng tạo đơn hàng
* Chức năng xem sản phẩm
* Chức năng xem chi tiết sản phẩm
* Chức năng xem tin tức

#### Chức năng đối với khách hàng vãng lai:

* Chức năng đăng ký tài khoản khách hàng
* Chức năng đăng nhập hệ thống
* Chức năng tìm kiếm sản phẩm
* Chức năng xem chi tiết sản phẩm
* Chức năng xem tin tức

#### Chức năng đối với người quản trị hệ thống:

* Chức năng đăng nhập hệ thống
* Chức năng thoát khỏi hệ thống
* Chức năng quản lý danh mục sản phẩm

Thêm danh mục

Xoá danh mục

Cập nhật danh mục

* Chức năng quản lý sản phẩm

Thêm sản phẩm

Xoá sản phẩm

Cập nhật sản phẩm

Tìm kiếm sản phẩm

* Chức năng quản lý đơn đặt hàng

Duyệt đơn đặt hàng

Xoá đơn đặt hàng

Xem chi tiết đơn đặt hàng

Tìm kiếm đơn đặt hàng

* Chức năng quản lý tin tức

Tạo mới tin tức

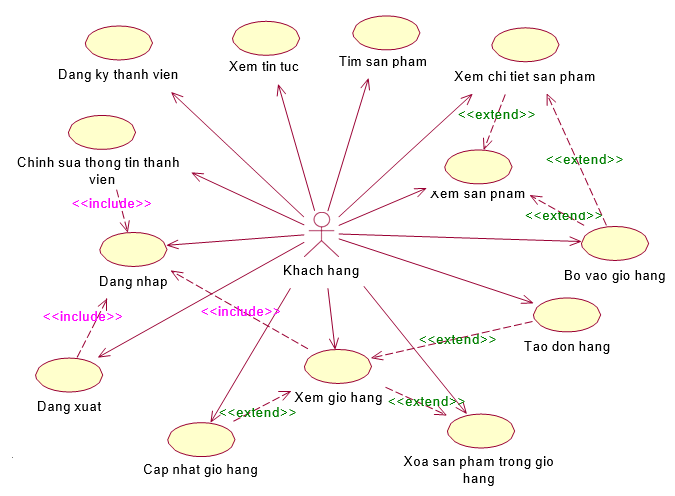
Xoá tin tức

Cập nhật tin tức

* Chức năng xem sản phẩm
* Chức năng xem chi tiết sản phẩm
* Chức năng xem tin tức

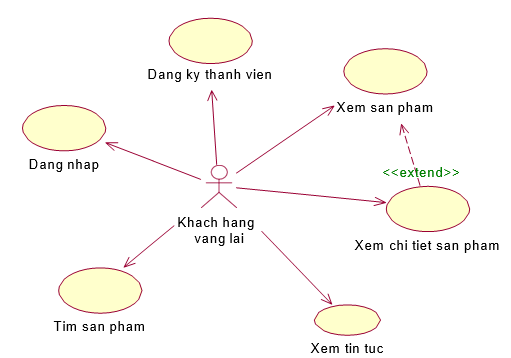
### Use-case-diagram

#### Use-case-diagram cho khách hàng



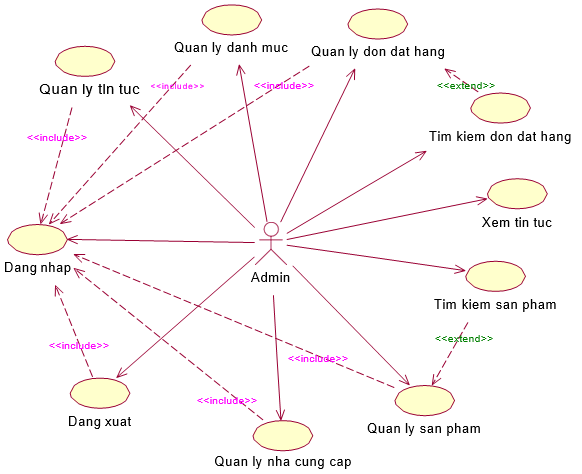
Hình ‑: Use-case-diagram cho khách hàng

#### Use-case-diagram cho khách hàng chưa đăng nhập



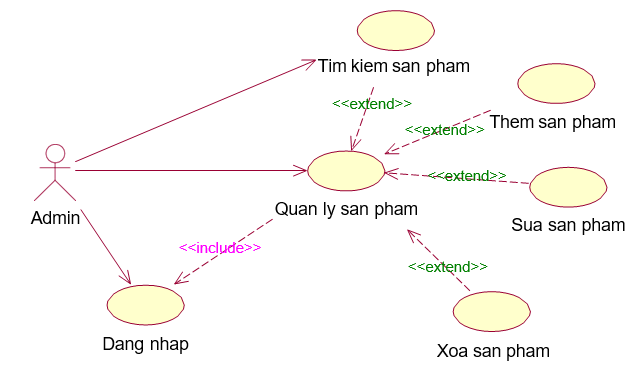
Hình ‑: Use-case-diagram cho khách hàng vãng lai

#### Use-Case Diagram cho bộ phận quản trị hệ thống



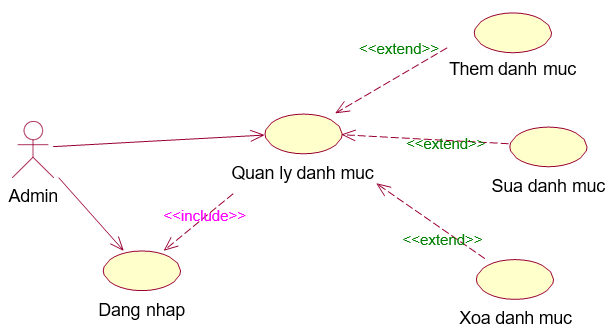
Hình ‑: Use-case-diagram cho bộ phận quản trị hệ thống

#### Quản lý sản phẩm



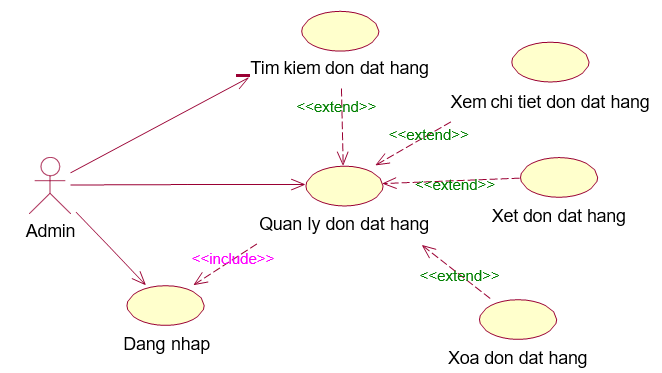
Hình ‑: Use-case-diagram admin quản lý sản phẩm

#### Quản lý danh mục



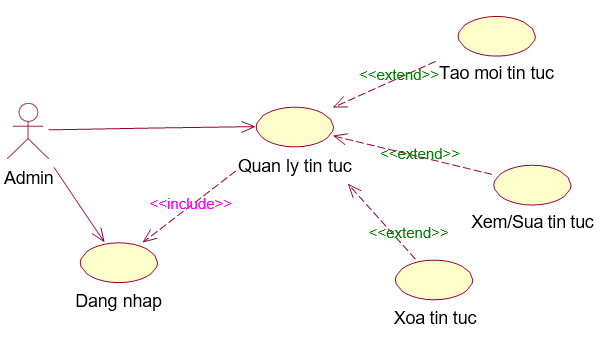
Hình ‑: Use-case-diagram admin quản lý danh mục

#### Quản lý đơn đặt hàng



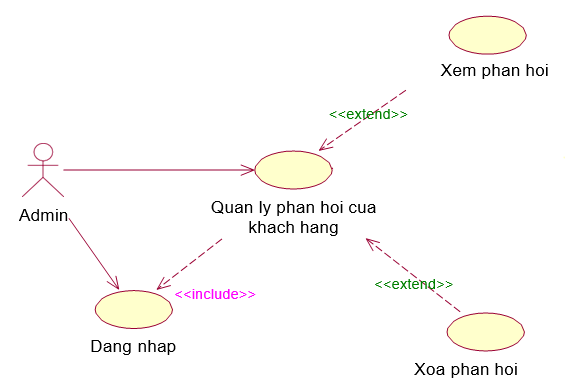
Hình ‑: Use-case-diagram admin quản lý đơn đặt hàng

#### Quản lý tin tức



Hình ‑: Use-case-diagram admin quản lý tin tức

#### Quản lý phản hồi của khách hàng



Hình ‑: Use-case-diagram admin quản lý phản hồi của khách hàng

### Mô tả chi tiết các chức năng của hệ thống

#### Thêm vào giỏ hàng

**Tóm tắt**

* Use-case này cho phép tạo ra giỏ hàng ảo cho khách hàng.
* Giỏ hàng ảo sẽ chứa sản phẩm mà khách hàng chọn mua.
* Giỏ hàng ảo này sẽ tồn tại từ khi khách hàng chọn “thêm vào” cho đến khi khách hàng rời khỏi cửa hàng.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi khách hàng chọn mua sản phẩm bằng cách click lên nút “thêm vào”.

**Dòng sự kiện chính**

* Actor chọn nút “thêm vào” tồn tại dưới từng sản phẩm.
* Hệ thống lấy “mã sản phẩm” vừa được actor chọn.
* Hệ thống lấy thông tin về khách hàng (mã khách hàng).
* Hệ thống tìm mã giỏ hàng đã tồn tại trong hệ thống chưa
* Nếu giỏ hàng chưa được tạo
* Hệ thống tạo ra một mã giỏ hàng.
* Hệ thống lưu những thông tin vào database: mã giỏ hàng, mã khách hàng, mã sản phẩm, số lượng sản phẩm, ngày thực hiện.
* Nếu giỏ hàng đã tồn tại

Hệ thống lưu những thông tin vào database: mã giỏ hàng, mã khách hàng, mã sản phẩm, số lượng sản phẩm, ngày thực hiện.

**Các dòng sự kiện khác**

Khách hàng chưa đăng nhập thành viên, khi khách hàng nhấn nút “Thêm vào” thì hệ thống yêu cầu khách hàng phải đăng ký thành viên hay đăng nhập thành viên. Khi khách hàng hoàn tất các thủ tục đó thì use-case xem như kết thúc.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Trước khi use-case này bắt đầu thì actor phải đăng nhập vào hệ thống với vai trò là khách hàng.

**Điều kiện hoàn tất**

Không có.

**Điểm mở rộng**

Không có.

#### Xem giỏ hàng

**Tóm tắt**

* Use-case này cho phép actor xem chi tiết giỏ hàng của mình.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn chức năng xem giỏ hàng bằng cách click vào biểu tượng giỏ hàng.

**Dòng sự kiện chính**

* Actor chọn chức năng xem giỏ hàng
* Hệ thống lấy mã giỏ hàng tồn tại trong hệ thống dưới dạng session hoặc cookies.
* Hệ thống tìm trong database các mẫu tin ứng với mã giỏ hàng vừa lấy được và hiển thị giỏ hàng với các thông tin: tên sản phẩm, giá sản phẩm, số lượng, số tiền ứng với một sản phẩm.

**Các dòng sự kiện khác**

* Nếu khách hàng chưa là thành viên thì hệ thống sẽ hiện thị phần đăng ký đòi actor phải đăng ký thành viên.
* Khi actor đăng ký hoàn tất thì use-case xem như kết thúc.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Trước khi use-case này bắt đầu thì actor phải đăng nhập vào hệ thống với vai trò là khách hàng.

**Điều kiện hoàn tất**

Không có.

**Điểm mở rộng**

Giảm số lượng sản phẩm trong giỏ hàng. Sửa số lượng sản phẩm trong giỏ hàng.

#### Giảm sản phẩm trong giỏ hàng

**Tóm tắt**

* Use-case cho phép actor xóa các sản phẩm chứa trong giỏ hàng của mình.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

* Use-case bắt đầu khi actor chọn chức năng “xóa sản phẩm” có trong giỏ hàng.

**Dòng sự kiện chính**

* Actor chọn chức năng “Xóa sản phẩm” cho một sản phẩm có trong giỏ hàng.
* Hệ thống lấy mã sản phẩm vừa được actor chọn trong giỏ hàng.
* Hệ thống lấy thông tin về actor (mã khách hàng).
* Hệ thống lấy mã giỏ hàng trong hệ thống tồn tại dưới dạng Session hoặc Cookie.
* Hệ thống tìm mẫu tin trong database ứng với mã giỏ hàng, mã sản phẩm, mã khách hàng và xóa mẫu tin đó đi.
* Hệ thống hiển thị giỏ hàng ở trạng thái mới.

**Các dòng sự kiện khác**

Không có.

**Các yêu cầu đặc biệt**

Không có

**Điều kiện tiên quyết**

Giỏ hàng phải tồn tại.

Có sản phẩm trong giỏ hàng.

**Điều kiện hoàn tất**

Giỏ hàng phải được hiển thị lại ở trạng thái vừa được actor cập nhật.

**Điểm mở rộng**

Xem giỏ hàng.

#### Sửa số lượng sản phẩm trong giỏ hàng

**Tóm tắt**

* Use-case này cho phép actor cập nhật lại số lượng mua của một sản phẩm trong giỏ hàng của mình.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

* Use-case bắt đầu khi actor chọn chức năng “cập nhật” có trong giỏ hàng ứng với từng sản phẩm.

**Dòng sự kiện chính**

* Actor chọn chức năng “cập nhật” cho một sản phẩm có trong giỏ hàng.
* Hệ thống lấy mã sản phẩm vừa được actor chọn trong giỏ hàng.
* Hệ thống lấy thông tin về actor.
* Hệ thống lấy mã giỏ hàng trong hệ thống tồn tại dưới dạng Session hoặc Cookies.
* Hệ thống tìm mẫu tin trong database ứng với mã giỏ hàng, mã sản phẩm, mã khách hàng.
* Hệ thống sẽ hiện giỏ hàng với mục số lượng ở dạng “sửa” ngay tại sản phẩm trong giỏ hàng mà actor chọn “cập nhật”
* Actor cập nhật lại mục số lượng.

**Các dòng sự kiện khác**

* Hệ thống sẽ cập nhật lại mẫu tin ứng với mã khách hàng, mã sản phẩm, mã giỏ hàng trong database.
* Use-case “xem giỏ hàng” được thực hiện.

**Các yêu cầu đặc biệt**

Không có

**Điều kiện tiên quyết**

Giỏ hàng phải tồn tại.

Có sản phẩm trong giỏ hàng.

**Điều kiện hoàn tất**

Giỏ hàng phải được hiển thị lại ở trạng thái vừa được actor cập nhật.

**Điểm mở rộng**

Xem giỏ hàng.

#### Tìm sản phẩm

**Tóm tắt**

* Use-case này cho phép actor tìm một sản phẩm trong danh mục.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn chức năng “tìm sản phẩm”.

**Dòng sự kiện chính**

* Actor nhập tên sản phẩm hoặc thông tin về sản phẩm cần tìm.
* Actor chọn hình thức tìm kiếm.
* Actor chọn khởi động tiến trình tìm kiếm.
* Hệ thống tìm trong database những sản phẩm có tên trùng hay gần trùng với tên mà actor đã nhập.
* Hệ thống hiển thị các sản phẩm tìm được.

**Các dòng sự kiện khác**

* Nếu không có sản phẩm nào được tìm thấy thì hệ thống sẽ thông báo cho actor biết là không tìm thấy sản phẩm.
* Hệ thống sẽ quay lại đầu use-case.

**Các yêu cầu đặc biệt**

Không có

**Điều kiện tiên quyết**

Không có.

**Điều kiện hoàn tất.**

Tất cả các sản phẩm tìm được phải được hiển thị.

**Điểm mở rộng**

Không có.

#### Đăng ký thành viên

**Tóm tắt**

* Use-case cho phép actor đăng ký để trở thành thành viên của cửa hàng.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn chức năng “đăng ký thành viên”.

**Dòng sự kiện chính**

* Actor chọn chức năng “đăng ký thành viên”.
* Hệ thống hiển thị mục đăng ký thành viên.
* Actor điền vào các thông tin: Tên đăng nhập, mật khẩu, xác nhận mật khẩu, địa chỉ, email, điện thoại, fax.
* Actor chọn nút “đăng ký”.
* Hệ thống lưu toàn bộ thông tin mà actor vừa xác nhận xuống database.
* Hệ thống hiển thị màn hình của thành viên.

**Các dòng sự kiện khác**

Nếu actor không điền đầy đủ các thông tin bắt buộc thì hệ thống không cho đăng ký và buộc phải điền đầy đủ những thông tin này.

**Các yêu cầu đặc biệt**

Không có

**Điều kiện tiên quyết**

Không có.

**Điều kiện hoàn tất**

Phải có một thông báo cho actor biết tình trạng đăng ký của actor có được hoàn thành hay không.

Phải chuyển đến giao diện đăng nhập thành công.

**Điểm mở rộng**

Sửa thông tin thành viên.

#### Chỉnh sửa thông tin thành viên

**Tóm tắt**

* Use-case này cho phép actor chỉnh sửa những thông tin của mình trong hệ thống.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn chức năng “chỉnh sửa thông tin” thành viên.

**Dòng sự kiện chính**

* Actor chọn chức năng chỉnh sửa thông tin thành viên.
* Hệ thống xác định tên tài khoản và mật khẩu của actor đang truy cập.
* Hệ thống tìm trong database những thông tin của actor tương ứng với tên đăng nhập và mật khẩu và hiển thị các thông tin đó trên một mục.
* Actor chỉnh sửa lại thông tin.
* Actor chọn “cập nhật” hoặc “thoát”.

**Các dòng sự kiện khác**

* Nếu actor chọn “cập nhật” thì hệ thống sẽ cập nhật lại những thông tin mà actor vừa chỉnh sửa vào database.
* Nếu actor chọn “thoát” thì use-case kết thúc.
* Nếu actor chọn “sửa thông tin” mà chưa đăng nhập vào hệ thống thì một thông báo được hiển thị nhắc actor đăng nhập vào hệ thống và use-case kết thúc.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Actor đã được đăng nhập vào hệ thống.

**Điều kiện hoàn tất**

Hệ thống phải hiển thị một thông báo cho actor biết về quá trình chỉnh sửa thông tin có thành công hay không.

**Điểm mở rộng**

Không có.

#### Đăng nhập thành viên

**Tóm tắt**

* Use-case này cho phép actor đăng nhập vào hệ thống với vai trò là khách hàng.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn chức năng “đăng nhập” vào hệ thống với vai trò khách hàng.

**Dòng sự kiện chính**

* Actor chọn chức năng “đăng nhập”
* Hệ thống hiển thị form gồm tên truy cập và mật khẩu.
* Actor nhập tên truy cập và mật khẩu.
* Actor chọn đăng nhập.
* Hệ thống kiểm tra tên truy cập và mật khẩu, xác nhận là đúng và cho phép actor đăng nhập vào hệ thống với vai trò khách hàng.

**Các dòng sự kiện khác**

* Nếu actor nhập tên truy cập và mật khẩu không đúng thì hệ thống sẽ báo lỗi, sau ba lần nhập không thành công hệ thống sẽ hủy bỏ việc đăng nhập.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Không có

**Điều kiện hoàn tất**

Hệ thống thông báo việc đăng nhập có thành công hay không.

**Điểm mở rộng**

Không có.

#### Tạo đơn đặt hàng

**Tóm tắt**

* Use-case này cho phép actor tạo ra đơn đặt hàng.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn chức năng “tạo đơn đặt hàng”.

**Dòng sự kiện chính**

* Actor chọn chức năng “tạo đơn đặt hàng”
* Hệ thống hiển thị một form yêu cầu actor nhập các thông tin của actor, sau đó hệ thống load thông tin có sẵn của khách hàng lên form.
* Actor điền các thông tin được yêu cầu:họ và tên, địa chỉ email, điện thoại.
* Actor chọn “đặt hàng” hoặc “thoát”.

**Các dòng sự kiện khác**

* Nếu actor chọn đặt hàng , hệ thống xác nhận thông tin từ session về giỏ hàng và khách hàng, các thông tin từ đơn đặt hàng về người mua và lưu xuống database. Sau đó hệ thống tạo các đơn đặt hàng tương ứng với các sản phẩm có trong giỏ hàng và lưu thông tin đơn đặt hàng vào database.
* Nếu actor chọn thoát thì use-case kết thúc.
* Actor có thể xem giỏ hàng ngay khi lập đơn đặt hàng để xác nhận lại các thông tin về những sản phẩm mà actor muốn đặt hàng.
* Actor có thể chỉnh sửa số lượng mua của sản phẩm hay bỏ bất kỳ sản phẩm nào trong giỏ hàng.
* Nếu actor nhập không đầy đủ các thông tin của người nhận được yêu cầu, hệ thống sẽ thông báo lỗi buộc actor nhập đầy đủ các thông tin trước khi nhấn nút “tạo đơn đặt hàng”.
* Các thông tin người mua được hệ thống xác định và chuyển lên form không cần actor nhập.
* Nếu người nhận là người mua thì actor có thể chuyển thông tin từ người mua sang người nhận một các dễ dàng mà không cần phải nhập.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Actor đã được đăng nhập vào hệ thống dưới vai trò khách hàng. Giỏ hàng phải tồn tại ít nhất một sản phẩm.

**Điều kiện hoàn tất**

Hệ thống phải hiển thị một thông báo cho actor biết về những thông tin cần làm tiếp theo sau khi đặt hàng.

Nếu việc đặt hàng thất bại hệ thống phải hiển thị một thông báo.

**Điểm mở rộng**

Xem giỏ hàng.

Giảm sản phẩm có trong giỏ hàng.

Sửa số lượng sản phẩm có trong sản phẩm.

#### Xem sản phẩm

**Tóm tắt**

* Use-case này cho phép actor xem tất cả các sản phẩm có trong danh mục
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn một hãng điện thoại trong cửa hàng.

**Dòng sự kiện chính**

* Actor chọn một hãng điện thoại.
* Hệ thống lấy mã danh mục mà actor đang chọn.
* Hệ thống hiển thị các sản phẩm thuộc hãng điện thoại đó.
* Actor chọn sản phẩm mà actor quan tâm.
* Hệ thống xác định sản phẩm được chọn.
* Hệ thống lấy các sản phẩm thuộc một hãng điện thoại đang được chọn.
* Hệ thống hiển thị tất cả các sản phẩm của các hãng điện thoại mà actor đang viếng thăm.

**Các dòng sự kiện khác**

* Actor có thể xem chi tiết sản phẩm hoặc bỏ sản phẩm được chọn vào giỏ hàng.

**Các yêu cầu đặc biệt**

Không có

**Điều kiện tiên quyết**

Không có

**Điều kiện hoàn tất**

Không có.

**Điểm mở rộng**

Xem chi tiết sản phẩm.

Bỏ sản phẩm vào giỏ hàng.

#### Xem chi tiết sản phẩm

**Tóm tắt**

* Use-case này cho phép actor xem thông tin chi tiết sản phẩm mà khách hàng quan tâm có trong danh mục.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn một sản phẩm trong danh mục.

**Dòng sự kiện chính**

* Actor chọn một sản phẩm có trong danh mục.
* Hệ thống lưu lại mã sản phẩm mà khách hàng vừa chọn và tìm trong database những thông tin ứng với mã sản phẩm đó: Tên sản phẩm, hãng sản xuất, giá, hình ảnh minh họa, loại model, thông tin sản phẩm.

**Các dòng sự kiện khác**

Không có.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Không có.

**Điều kiện hoàn tất**

Không có.

**Điểm mở rộng**

Bỏ sản phẩm vào giỏ hàng.

#### Xem tin tức

**Tóm tắt**

* Use-case này cho phép khàch hàng xem các tin tức tại website.
* Khách hàng là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn một tin tức trong phần tin tức.

**Dòng sự kiện chính**

* Actor chọn một tin tức.
* Hệ thống lưu lại mã tin tức mà khách hàng vừa chọn và tìm trong database những thông tin ứng với mã tin tức đó: tiêu đề, nội dung, hình ảnh minh họa tin tức đó.

**Các dòng sự kiện khác**

Không có.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Không có.

**Điều kiện hoàn tất**

Không có.

**Điểm mở rộng**

Xem chi tiết tin tức.

#### Đăng nhập cửa hàng

**Tóm tắt**

* Use-case cho phép actor đăng nhập vào hệ thống với vai trò là người quản trị.
* Người quản trị là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu khi actor chọn chức năng “đăng nhập cửa hàng” dưới vai trò là người quản trị.

**Dòng sự kiện chính**

* Actor chọn chức năng “đăng nhập cửa hàng”.
* Hệ thống hiển thị form gồm tên đăng nhập và mật khẩu.
* Actor chọn đăng nhập.
* Hệ thống kiểm tra tên đăng nhập và mật khẩu, xác nhận là đúng và cho phép actor đăng nhập vào hệ thống với vai trò là người quản trị.

**Các dòng sự kiện khác**

Nếu actor nhập tên và mật khẩu không đúng thì hệ thống sẽ báo lỗi. Sau ba lần nhập không thành công hệ thống hủy bỏ việc đăng nhập.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Không có.

**Điều kiện hoàn tất**

Hệ thống thông báo việc đăng nhập thành công hay thất bại.

**Điểm mở rộng**

Không có.

#### Quản lý danh mục

**Tóm tắt**

* Use-case này cho phép actor xem, duyệt, xóa, thêm các hãng điện thoại.
* Người quản trị là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu sau khi actor đăng nhập, chọn chức năng quản lý danh mục.

**Dòng sự kiện chính**

* Hệ thống hiển thị tên danh mục sản phẩm.
* Hệ thống hỗ trợ cho actor các chức năng:
  + Thêm danh mục: actor nhập thông tin danh mục, chọn thêm.
  + Xoá danh mục: actor chọn danh mục cần xoá và nhấn nút xoá.
  + Cập nhật danh mục: actor chọn danh mục cần cập nhật, cập nhật thông tin và chọn sửa.
* Hệ thống sẽ cập nhật thông tin danh mục vào cơ sở dữ liệu.

**Các dòng sự kiện khác**

Không có.

**Điều kiện tiên quyết**

Người quản tri phải đăng nhập vào hệ thống.

**Điều kiện hoàn tất**

Không có

**Điểm mở rộng**

Không có.

#### Quản lý tin tức

**Tóm tắt**

* Use-case này cho phép actor xem, duyệt, xóa, tạo mới các mục tin.
* Người quản trị là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu sau khi actor đăng nhập, chọn chức năng “quản lý tin tức”.

**Dòng sự kiện chính**

* Hệ thống hiển thị danh sách các mục tin.
* Hệ thống hỗ trợ cho actor các chức năng
  + Tạo mới tin tức: actor nhập thông tin tin tức, chọn lưu.
  + Xoá tin tức: actor chọn tin tức cần xoá, nhấn nút xoá
  + Cập nhật tin tức: actor chọn tin tức cần cập nhật, nhập thông tin tin tức và chọn sửa.
* Hệ thống sẽ cập nhật thông tin mục tin vào cơ sở dữ liệu.

**Các dòng sự kiện khác**

Không có.

**Các yêu cầu đặc biệt**

Không có

**Điều kiện tiên quyết**

Người quản trị phải đăng nhập vào hệ thống

**Điều kiện hoàn tất**

Không có.

**Điểm mở rộng**

Không có.

#### Quản lý sản phẩm

**Tóm tắt**

* Use-case này cho phép actor xem, duyệt, thêm, xóa các sản phẩm của các hãng điện thoại.
* Người quản trị là actor của use-case này.

**Dòng sự kiện**

Use-case bắt đầu sau khi actor đăng nhập, chọn chức năng “quản lý sản phẩm”.

**Dòng sự kiện chính**

* Hệ thống hiển thị các sản phẩm của các hãng điện thoại.
* Hệ thống hỗ trợ việc tìm kiếm sản phẩm hay lọc sản phẩm theo từng hãng điện thoại.
* Actor có thể lọc sản phẩm chưa duyệt, đã duyệt.
* Hệ thống hỗ trợ cho actor duyệt sản phẩm và cập nhật thông tin vào cơ sở dữ liệu.
* Nếu actor chọn xóa sản phẩm thì hệ thống sẽ xóa sản phẩm trong hệ thống.

**Các dòng sự kiện khác**

Không có.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Người quản trị phải đăng nhập vào hệ thống.

**Điều kiện hoàn tất**

Không có.

**Điểm mở rộng**

Không có.

#### Quản lý đơn đặt hàng

**Tóm tắt**

* Use-case này cho phép actor dọn dẹp các đơn đặt hàng (đơn đặt hàng tồn tại quá một năm) hay một số đơn đặt hàng đã được backup trong database và xem cũng như thống kê các đơn đặt hàng như mong muốn..
* Người quản trị là actor là use-case này.

**Dòng sự kiện**

Use-case bắt đầu sau khi actor đăng nhập, chọn chức năng “quản lý đơn đặt hàng”.

**Dòng sự kiện chính**

* Actor chọn chức năng “quản lý đơn đặt hàng”.
* Hệ thống tìm trong database những đơn đặt hàng của khách hàng trong ngày.
* Actor nhập điều kiện lọc các đơn đặt hàng theo thời gian.
* Hệ thống hiển thị tất cả các đơn đặt hàng tìm được trong database.
* Hệ thống hỗ trợ chức năng duyệt đơn đặt hàng.
* Actor chọn một hoặc nhiều đơn đặt hàng cần hủy.
* Hệ thống xác định các mã đơn đặt hàng mà actor đã chọn.
* Hệ thống xóa các đơn đặt hàng trong database mà được actor chọn.

**Các dòng sự kiện khác**

* Nếu không có đơn đặt hàng nào tồn tại trong databse ở thời điểm hiện tại, hệ thống hiển thị một thông báo.
* Use-case kết thúc.

**Các yêu cầu đặc biệt**

Không có.

**Điều kiện tiên quyết**

Người quản trị phải đăng nhập vào hệ thống.

**Điều kiện hoàn tất**

Việc xóa các đơn đặt hàng thành công hay thất bại đều phải hiện một thông báo.

Các đơn đặt hàng còn lại trong database phải được hiển thị lại sau khi việc dọn dẹp thành công.

**Điểm mở rộng**

Không có.

## Phân tích hệ thống về dữ liệu

Mục đích của quá trình phân tích về dữ liệu là xác định các thực thể cần thiết, các thuộc tính của dữ liệu và mối quan hệ giữa các thực thể. Một hệ thống thông tin phải được tổ chức rõ ràng, đầy đủ, tránh dư thừa dữ liệu, tránh trùng lặp dữ liệu và đảm bảo độc lập dữ liệu.

### Xác định các thực thể và các thuộc tính cho thực thể

Ta cần lưu trữ những thực thể sau:

**Danh mục** (dòng điện thoại): gồm các thuộc tính: mã danh mục, tên danh mục, ngày tạo, trạng thái.

**Sản phẩm** (điện thoại): gồm các thuộc tính: mã sản phẩm, tên sản phẩm, giá xuất, ảnh sản phẩm, mô tả, ngày tạo, bộ nhớ, ram, màu, số lượng, trạng thái, mã danh mục.

**Ảnh sản phẩm**: gồm các thuộc tính: mã ảnh sản phẩm, mã sản phẩm, ảnh sản phẩm, ngày tạo.

**Tài khoản**: gồm các thuộc tính: mã tài khoản, tên tài khoản, mật khẩu, mã khách hàng, là quản trị viên(1) hay không(0), ngày tạo, trạng thái.

**Quyền trong website**: gồm các thuộc tính: mã quyền, tên quyền, trạng thái.

**Quyền gán cho tài khoản**: gồm các thuộc tính: mã quyền gán, mã tài khoản, mã quyền.

**Giảm giá**: gồm các thuộc tính: mã giảm giá, tên voucher giảm giá, ngày bắt đầu giảm, ngày kết thúc giảm, số lượng voucher, ngày tạo.

**Khách hàng**: gồm các thuộc tính: mã khách hàng, tên đầy đủ khách hàng, mật khẩu, số điện thoại, địa chỉ, email, ảnh đại diện, ngày tạo, trạng thái.

**Giỏ hàng**: gồm các thuộc tính: mã giỏ hàng, mã khách hàng, mã sản phẩm, số lượng.

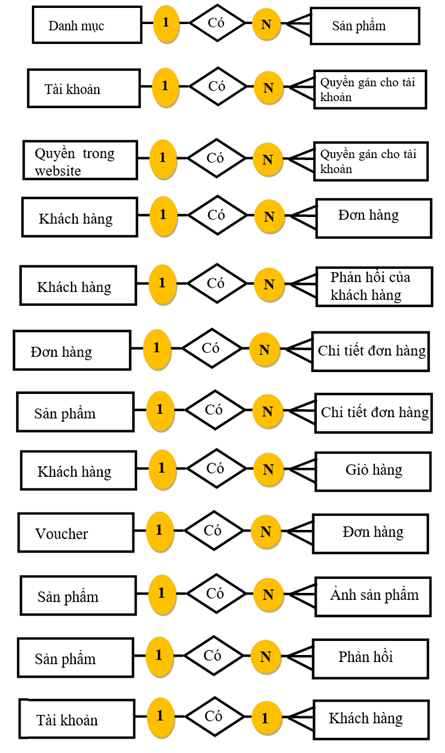
**Đơn hàng**: gồm các thuộc tính: mã đơn hàng, mã khách hàng, tên đầy đủ khách hàng, email, số điện thoại, địa chỉ, mã giảm giá, tổng giá trị đơn hàng, ngày tạo, trạng thái.

**Chi tiết đơn hàng**: gồm các thuộc tính: mã thông tin chi tiết, mã đơn hàng, mã sản phẩm, số lượng, giá tiền, ngày tạo.

**Phản hồi của khách hàng**: gồm các thuộc tính: mã phản hồi, mã khách hàng, mã sản phẩm, nội dung, ảnh phản hồi, ngày tạo, trạng thái.

### Xác định các liên kết và các mối quan hệ của thực thể

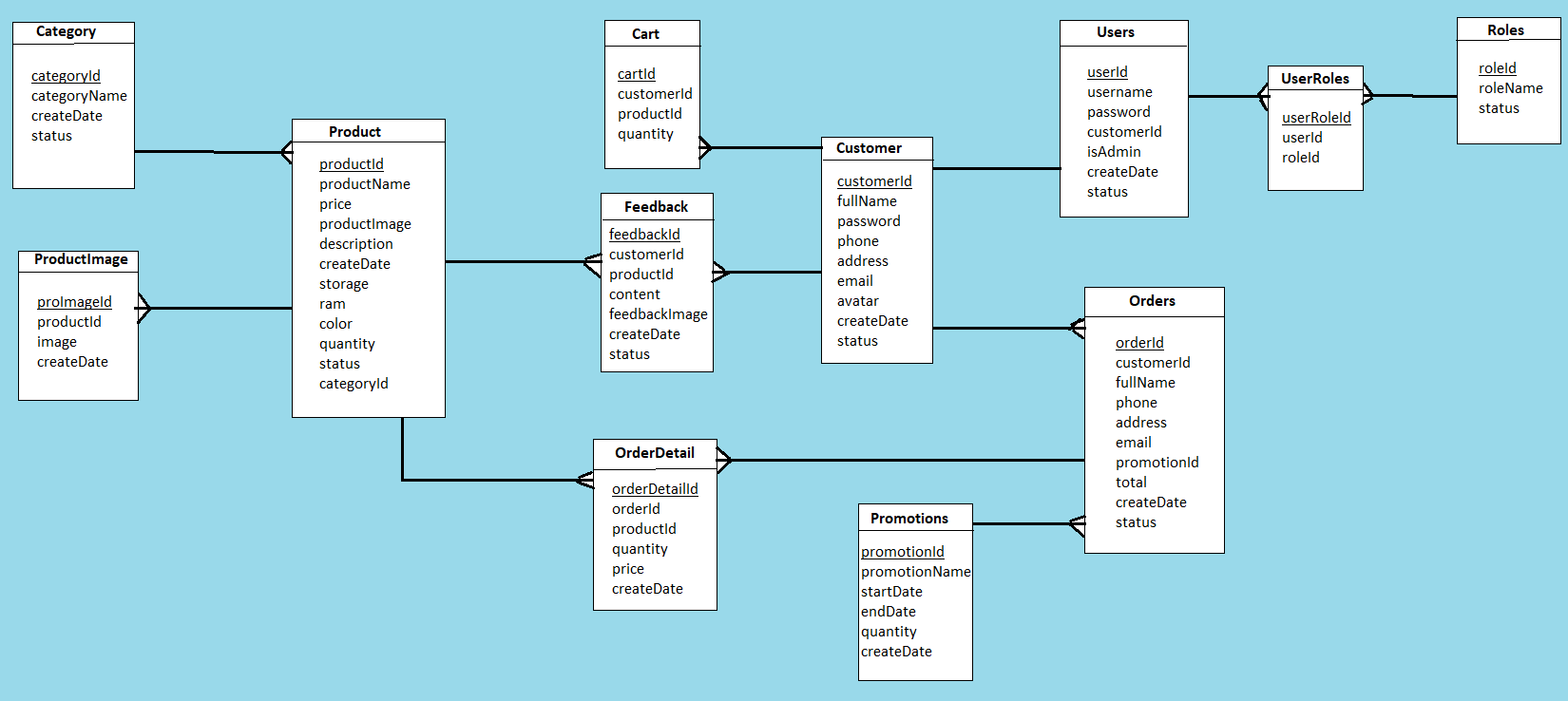
Dựa vào việc phân tích và tìm hiểu ta nhận thấy mối quan hệ và liên kết giữa các thực thể được xác định ở trên là:



Hình 3‑10: Liên kết giữa các thực thể

## Mô hình thực thể/ liên kết (Mô hình ER)

Dựa vào việc xác định các thực thể và mối quan hệ của các thực thể ta có mô hình thực thể liên kết sau:



Hình ‑: Mô hình thực thể liên kết ER

# : THIẾT KẾ CƠ SỞ DỮ LIỆU

## Mô tả thực thể kết hợp

Mô hình thực thể kết hợp diễn đạt các thành phần dữ liệu ở mức quan niệm dựa trên bảng danh mục các dữ liệu cơ bản. Các loại thực thể có những thuộc tính sau:

### Danh mục

**Category**(categoryId, categoryName, providerId, createDate, status)

**Tân từ**: Mỗi danh mục có 1 mã danh mục (categoryId) duy nhất để phân biệt với các danh mục khác. Mỗi danh mục còn được xác định bởi tên danh mục (categoryName). Danh mục có ngày tạo(createDate) và xác định danh mục ẩn(0) hay hiện(1) (status).

### Sản phẩm

**Product**(productId, productName, price, productImage, description, createDate, storage, ram, color, quantity, categoryId, status.)

**Tân từ**: Mỗi sản phẩm có 1 mã sản phẩm (productId) duy nhất để phân biệt với các sản phẩm khác. Mỗi sản phẩm còn được xác định bởi tên sản phẩm (productName). Sản phẩm có giá bán(price). Hình sản phẩm(productImage). Mô tả sản phẩm(description). Sản phẩm có ngày tạo(createDate). Dung lượng lưu trữ(storage). Số RAM(ram). Màu sản phẩm(color). Số lượng sản phẩm(quantity). Mã danh mục của sản phẩm(categoryId) và xác định danh mục ẩn(0) hay hiện(1) (status).

### Ảnh của sản phẩm

**ProductImage**(proImageId, productId, image, createDate)

**Tân từ**: Ảnh của sản phẩm có mã ảnh(proImageId), mã sản phẩm(productId), ảnh(image), ngày tạo(createDate).

### Khách hàng

**Customer**(customerId, fullName, password, phone, address, email, avatar, createDate, status)

**Tân từ**: Mỗi khách hàng có 1 mã khách hàng (customerId) duy nhất để phân biệt với các khách hàng khác. Mỗi khách hàng còn được xác định bởi tên khách hàng (fullName). Mỗi khách hàng có 1 mật khẩu riêng(password). Mỗi khách hàng có 1 số điện thoại riêng(phone). Khách hàng có địa chỉ giao hàng riêng. Tài khoản email riêng(email). Ảnh đại diện riêng(avatar). Khách hàng có ngày tạo(createDate) và xác định khách hàng ẩn(0) hay hiện(1) (status).

### Giỏ hàng

**Cart**(cartId, customerId, productId, quantity)

**Tân từ**: Mỗi giỏ hàng có 1 mã giỏ hàng(cartId) riêng để phân biệt với giỏ hàng khác. Có mã khách hàng tạo giỏ hàng ấy(customerId), có mã sản phẩm trong giỏ hàng(productId), số lượng sản phẩm(quantity).

### Voucher giảm giá

**Promotions**(promotionId, promotionName, startDate, endDate, quantity, createDate)

**Tân từ**: Mỗi voucher giảm giá có 1 mã giảm giá(promotionId), tên voucher(promotionName), ngày bắt đầu áp dụng voucher (startDate), ngày kết thúc áp dụng(endDate), số lượng voucher còn lại(quantity), ngày tạo voucher(createDate).

### Đơn hàng

**Orders**(orderId, customerId, fullName, phone, address, email, promotionId, total, createDate, status)

**Tân từ**: Mỗi đơn hàng có 1 mã đơn hàng (orderId) duy nhất để phân biệt với các đơn hàng khác. Mỗi đơn hàng được mua bởi khách hàng chứa thông tin là: mã khách hàng(customerId), tên khách hàng (fullName), số điện thoại khách hàng(phone), địa chỉ giao hàng(address), email khách hàng(email), mã giảm giá áp dụng(promotionId), tổng giá trị đơn hàng(total). Đơn hàng có ngày tạo(createDate) và xác định đơn hàng ẩn(0) hay hiện(1) (status).

### Chi tiết đơn hàng

**OrderDetail**(orderDetailId, orderId, productId, quantity, price, createDate)

**Tân từ**: Mỗi chi tiết đơn hàng có 1 mã chi tiết đơn hàng (orderDetailId) duy nhất để phân biệt với các chi tiết đơn hàng khác, mã đơn hàng(orderId), mã sản phẩm(productId), số lượng(quantity), giá chi tiết mặt hàng trong đơn(price), chi tiết đơn hàng có ngày tạo(createDate).

### Phản hồi của khách hàng

**FeedBack**(feedBackId, customerId, productId, content, feedbackImage, createDate, status)

**Tân từ**: Mỗi phản hồi của khách hàng có 1 mã phản hồi (feedBackId) duy nhất để phân biệt với các phản hồi khác, mã khách hàng phản hồi(customerId), mã sản phẩm(productId, nội dung phản hồi(content), ảnh phản hồi(feedBackImage). Phản hồi có ngày tạo(createDate) và xác định phản hồi ẩn(0) hay hiện(1) (status).

### Tài khoản

**Users**(userId, userName, password, customerId, isAdmin, createDate, status)

**Tân từ**: Mỗi tài khoản có mã tài khoản(userId), tên đăng nhập (userName), mật khẩu(password), mã khách hàng(customerId), là admin(1) hay người dùng thường(0)(isAdmin), ngày tạo tài khoản (createDate), trạng thái tài khoản(status).

### Quyền trong trang web

**Roles**(roleId, roleName, status)

**Tân từ**: Mỗi quyền có 1 mã quyền(roleId), tên quyền(roleName), trạng thái(status).

### Phân quyền cho người dùng

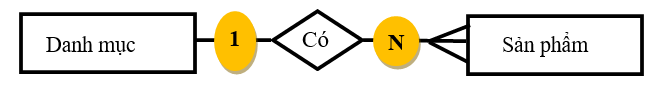
**UserRoles**(userRoleId, userId, roleId)

**Tân từ**: Mỗi phân quyền có 1 mã phân quyền(userRoleId), mã tài khoản(userId), mã quyền(roleId).

## Mô tả các mối kết hợp

Thể hiện sự quan hệ ngữ nghĩa giữa những thực thể ở ít nhất 2 loại thực thể khác nhau.

### Mối kết hợp: Category và Product



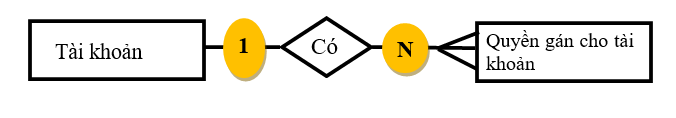
Hình ‑: Mối kết hợp giữa Danh mục và Sản phẩm

*Ý nghĩa:*

Một danh mục có thể có 1 hoặc nhiều sản phẩm.

Mỗi sản phẩm chỉ thuộc một danh mục.

### Mối kết hợp: Tài khoản và Quyền gán cho tài khoản



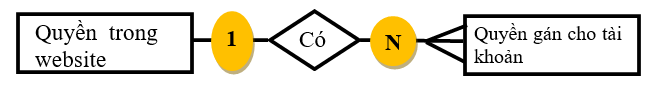
Hình ‑: Mối kết hợp giữa Tài khoản và Quyền gán cho tài khoản

*Ý nghĩa:*

Một tài khoản có thể có 1 hoặc nhiều quyền áp dụng cho trang web thông qua “Quyền gán cho tài khoản”(quyền người dùng và quyền admin).

Mỗi quyền gán cho mỗi tài khoản được tạo ra cho riêng tài khoản ấy.

### Mối kết hợp: Quyền trong website và Quyền gán cho tài khoản



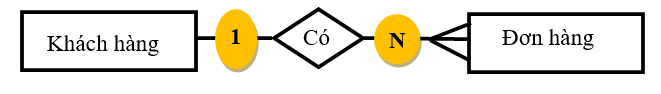
Hình ‑: Mối kết hợp giữa Quyền trong website và Quyền gán cho tài khoản

*Ý nghĩa:*

Một quyền trong website được gán vào 1 hoặc nhiều Quyền gán cho tài khoản để xác định quyền của tài khoản ấy.

Mỗi quyền gán cho tài khoản người dùng thì chỉ tồn tạo 1 quyền gán trong website.

### Mối kết hợp: Khách hàng và Đơn hàng



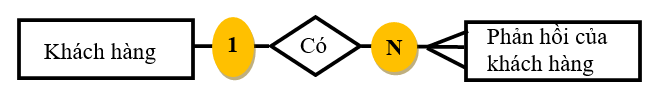
Hình ‑: Mối kết hợp giữa Khách hàng và Đơn hàng

*Ý nghĩa:*

Một khách hàng được có thể có 1 hoặc nhiều Đơn hàng.

Mỗi đơn hàng thì chỉ thuộc về 1 khách hàng.

### Mối kết hợp: Khách hàng và Phản hồi của khách hàng



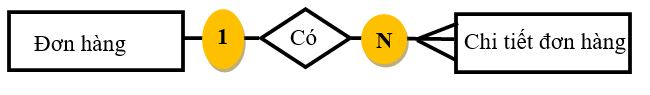
Hình ‑: Mối kết hợp giữa Khách hàng và Phản hồi của khách hàng

*Ý nghĩa:*

Một khách hàng được có thể có 1 hoặc nhiều Phản hồi của khách hàng.

Mỗi Phản hồi của khách hàng thì chỉ thuộc về 1 khách hàng.

### Mối kết hợp: Đơn hàng và Chi tiết đơn hàng



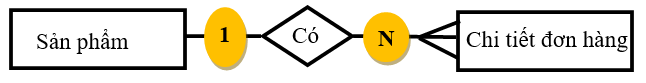
Hình ‑: Mối kết hợp giữa Đơn hàng và Chi tiết đơn hàng

*Ý nghĩa:*

Một Đơn hàng được có thể có 1 hoặc nhiều Chi tiết đơn hàng.

Mỗi Chi tiết đơn hàng thì chỉ thuộc về 1 Đơn hàng.

### Mối kết hợp: Sản phẩm và Chi tiết đơn hàng



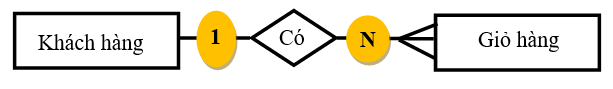
Hình ‑: Mối kết hợp giữa Sản phẩm và Chi tiết đơn hàng

*Ý nghĩa:*

Một Sản phẩm có thể nằm trong 1 hoặc nhiều Chi tiết đơn hàng.

Mỗi Chi tiết đơn hàng thì chỉ thuộc về 1 Sản phẩm.

### Mối kết hợp: Khách hàng và Giỏ hàng



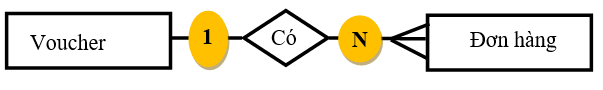
Hình ‑: Mối kết hợp giữa Khách hàng và Giỏ hàng

*Ý nghĩa:*

Một Khách hàng có tạo 1 hoặc nhiều Giỏ hàng.

Mỗi Giỏ hàng thì chỉ thuộc về 1 Khách hàng.

### Mối kết hợp: Voucher và Đơn hàng



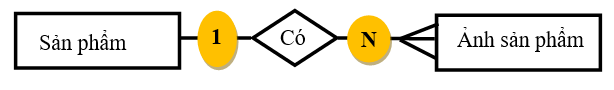
Hình ‑: Mối kết hợp giữa Voucher và Đơn hàng

*Ý nghĩa:*

Một Voucher có thể áp dụng cho 1 hoặc nhiều Đơn hàng.

Mỗi Đơn hàng thì chỉ dùng được 1 Voucher.

### Mối kết hợp: Sản phẩm và Ảnh sản phẩm



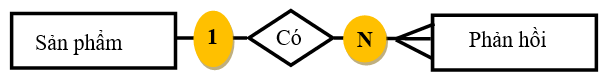
Hình ‑: Mối kết hợp giữa Sản phẩm và Ảnh sản phẩm

*Ý nghĩa:*

Một Sản phẩm có thể có 1 hoặc nhiều Ảnh sản phẩm.

Mỗi Ảnh sản phẩm thì chỉ thuộc về 1 Sản phẩm ấy.

### Mối kết hợp: Sản phẩm và Phản hồi của khách hàng



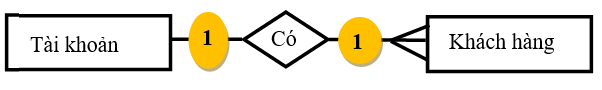
Hình ‑: Mối kết hợp giữa Sản phẩm và Phản hồi của khách hàng

*Ý nghĩa:*

Một Sản phẩm có thể có 1 hoặc nhiều Phản hồi của khách hàng.

Mỗi Phản hồi thì chỉ dùng để đánh giá 1 sản phẩm duy nhất.

### Mối kết hợp: Tài khoản và Khách hàng



Hình ‑: Mối kết hợp giữa Tài khoản và Khách hàng

*Ý nghĩa:*

Một Tải khoản chỉ dùng cho 1 Khách hàng duy nhất.

Một Khách hàng cũng chỉ có thể tạo duy nhất 1 tài khoản.

## Mô tả chi tiết các bảng

### Category

Bảng chứa thông tin các danh mục.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| categoryId | int |  | Mã danh mục | Khóa chính |
| categoryName | nvarchar | 100 | Tên danh mục |  |
| createDate | timestamp, | 1000 | Ngày tạo danh mục |  |
| status | bit |  | Kích hoạt  danh mục |  |

Bảng ‑: Bảng chứa thông tin các danh mục

### Product

Bảng chứa thông tin sản phẩm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| productId | Varchar | 10 | Mã sản phẩm | Khóa chính |
| productName | nvarchar | 100 | Tên danh mục |  |
| price | float | 1000 | Ngày tạo danh mục |  |
| productImage | nvarchar | max | Ảnh sản phẩm |  |
| description | ntext |  | Mô tả sản phẩm |  |
| createDate | timestamp |  | Ngày tạo |  |
| storage | int |  | Dung lượng lưu trữ |  |
| ram | int |  | Số RAM |  |
| color | varchar | 100 | Màu |  |
| quantity | int |  | Số lượng |  |
| categoryId | int |  | Mã danh mục | Khóa ngoại |
| status | bit |  | Kích hoạt  danh mục |  |

Bảng ‑: Bảng chứa thông tin sản phẩm

### ProductImage

Bảng chứa thông tin ảnh sản phẩm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| proImageId | int |  | Mã ảnh | Khóa chính |
| productId | varchar | 10 | Tên danh mục | Khóa ngoại |
| image | varchar | max | Ảnh |  |
| createDate | timestamp |  | Ngày tạo ảnh |  |

Bảng ‑: Bảng chứa thông tin ảnh sản phẩm

### Customer

Bảng chứa thông tin khách hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| customerId | int |  | Mã khách hàng | Khóa chính |
| fullName | nvarchar | 100 | Tên khách hàng |  |
| password | varchar | 100 | Mật khẩu |  |
| phone | varchar | 20 | Số điện thoại |  |
| address | nvarchar | 250 | Địa chỉ |  |
| email | varchar | 150 | Email |  |
| avatar | nvarchar | 250 | Ảnh đại diện |  |
| createDate | timestamp |  | Ngày tạo |  |
| status | bit |  | Trạng thái |  |

Bảng ‑: Bảng chứa thông tin ảnh khách hàng

### Cart

Bảng chứa thông tin giỏ hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| cartId | int |  | Mã giỏ hàng | Khóa chính |
| customerId | int |  | Mã khách hàng | Khóa ngoại |
| productId | int |  | Mã sản phẩm | Khóa ngoại |
| quantity | int |  | Số lượng giỏ hàng |  |

Bảng ‑: Bảng chứa thông tin ảnh giỏ hàng

### Promotions

Bảng chứa thông tin voucher giảm giá.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| promotionId | int |  | Mã giảm giá | Khóa chính |
| promotionName | varchar | 250 | Tên voucher |  |
| startDate | datetime | 10 | Ngày bắt đầu áp dụng |  |
| endDate | datetime |  | Ngày dừng áp dụng |  |
| quantity | int |  | Số lượng |  |
| createDate | timestamp |  | Ngày tạo |  |

Bảng ‑: Bảng chứa thông tin voucher giảm giá

### Orders

Bảng chứa thông tin đơn hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| orderId | int |  | Mã đơn hàng | Khóa chính |
| customerId | int |  | Mã khách hàng | Khóa ngoại |
| fullName | nvarchar | 100 | Tên khách hàng |  |
| phone | varchar | 20 | Số điện thoại |  |
| address | nvarchar | 250 | Địa chỉ |  |
| email | varchar | 150 | Email |  |
| promotionId | int |  | Mã giảm giá | Khóa ngoại |
| total | float |  | Tổng giá trị đơn hàng |  |
| createDate | timestamp |  | Ngày tạo |  |
| status | bit |  | Trạng thái |  |

Bảng ‑: Bảng chứa thông tin đơn hàng

### OrderDetail

Bảng chứa thông tin chi tiết đơn hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| orderDetailId | int |  | Mã chi tiết đơn hàng | Khóa chính |
| orderId | int |  | Mã đơn hàng | Khóa ngoại |
| productId | int |  | Mã sản phẩm | Khóa ngoại |
| quantity | int |  | Số lượng |  |
| price | float |  | Đơn giá |  |
| createDate | timestamp |  | Ngày tạo |  |

Bảng ‑: Bảng chứa thông tin chi tiết đơn hàng.

### FeedBack

Bảng chứa thông tin phản hồi của khách hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| feedBackId | int |  | Mã phản hồi | Khóa chính |
| customerId | int |  | Mã khách hàng | Khóa ngoại |
| productId | int |  | Mã sản phẩm | Khóa ngoại |
| content | ntext |  | Nội dung phản hồi |  |
| feedbackImage | varchar | max | Ảnh phản hồi |  |
| createDate | timestamp |  | Ngày tạo |  |
| status | bit |  | Trạng thái |  |

Bảng ‑: Bảng chứa thông tin phản hồi của khách hàng

### Users

Bảng chứa thông tin tài khoản.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| userId | int |  | Mã tài khoản | Khóa chính |
| userName | varchar | 150 | Tên đăng nhập |  |
| password | varchar | 100 | Mật khẩu |  |
| customerId | int |  | Mã khách hàng | Khóa ngoại |
| isAdmin | bit |  | Là admin hay không |  |
| createDate | timestamp |  | Ngày tạo |  |
| status | bit |  | Trạng thái |  |

Bảng ‑: Bảng chứa thông tin tài khoản

### Roles

Bảng chứa thông tin quyền.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| roleId | int |  | Mã quyền | Khóa chính |
| roleName | varchar | 50 | Tên quyền |  |
| status | bit |  | Trạng thái |  |

Bảng ‑: Bảng chứa thông tin quyền

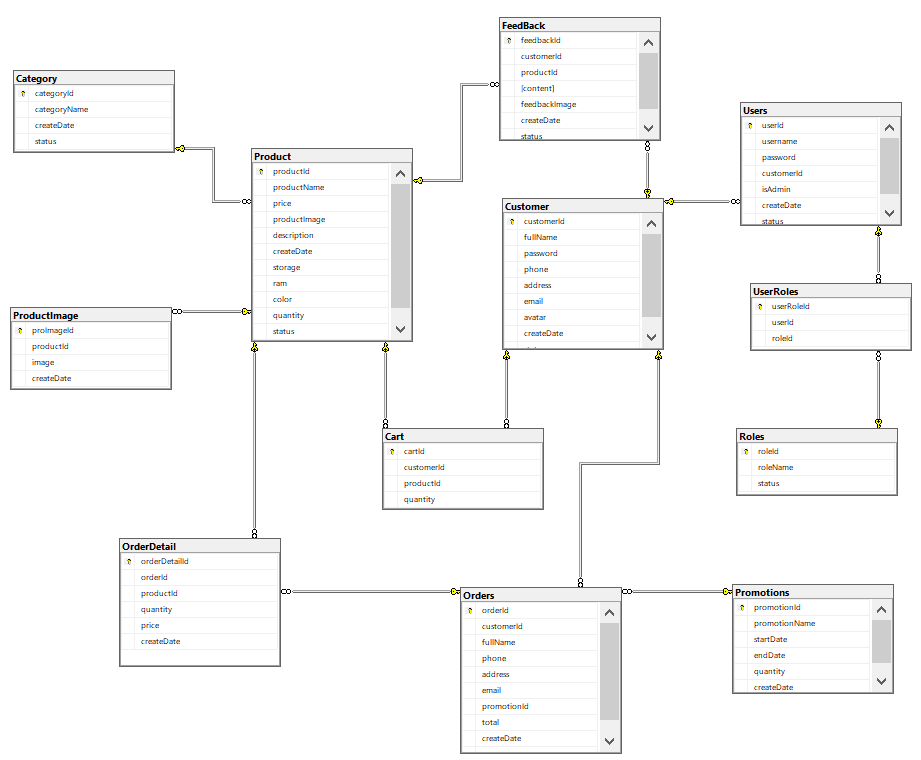
### UserRoles

Bảng chứa thông tin phân quyền.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vùng** | **Kiểu dữ liệu** | **Kích thước** | **Diễn giải** | **Ràng buộc** |
| userRoleId | int |  | Mã phân quyền | Khóa chính |
| userId | int |  | Mã tài khoản | Khóa ngoại |
| roleId | int |  | Mã quyền | Khóa ngoại |

Bảng ‑: Bảng chứa thông tin phân quyền

## Sơ đồ quan hệ dữ liệu giữa các bảng

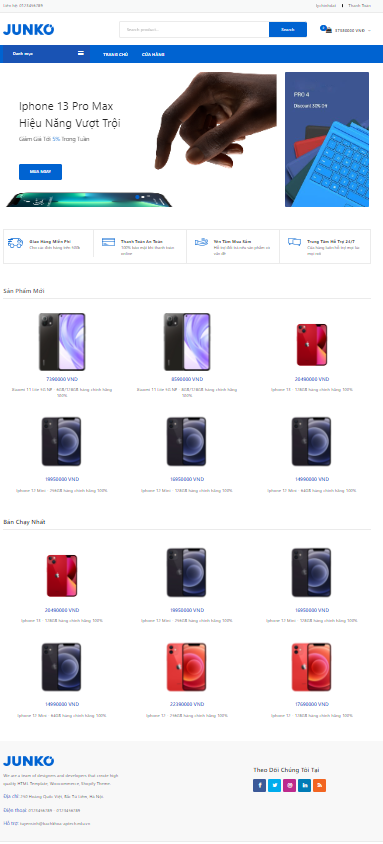


Hình ‑: Sơ đồ quan hệ dữ liệu giữa các bảng

# : DEMO SẢN PHẨM

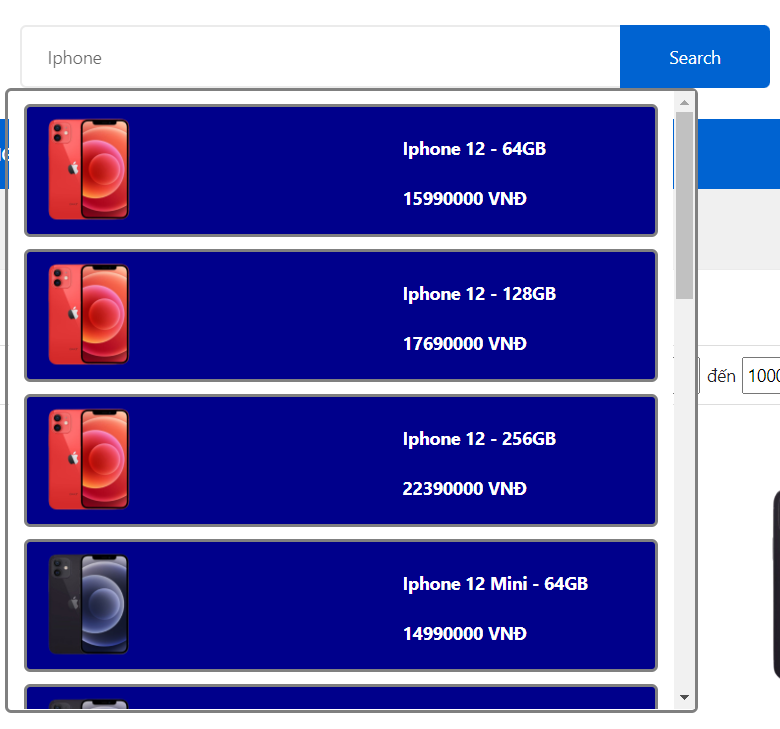
## Giao diện người dùng

Giao diện Trang chủ của website như sau:



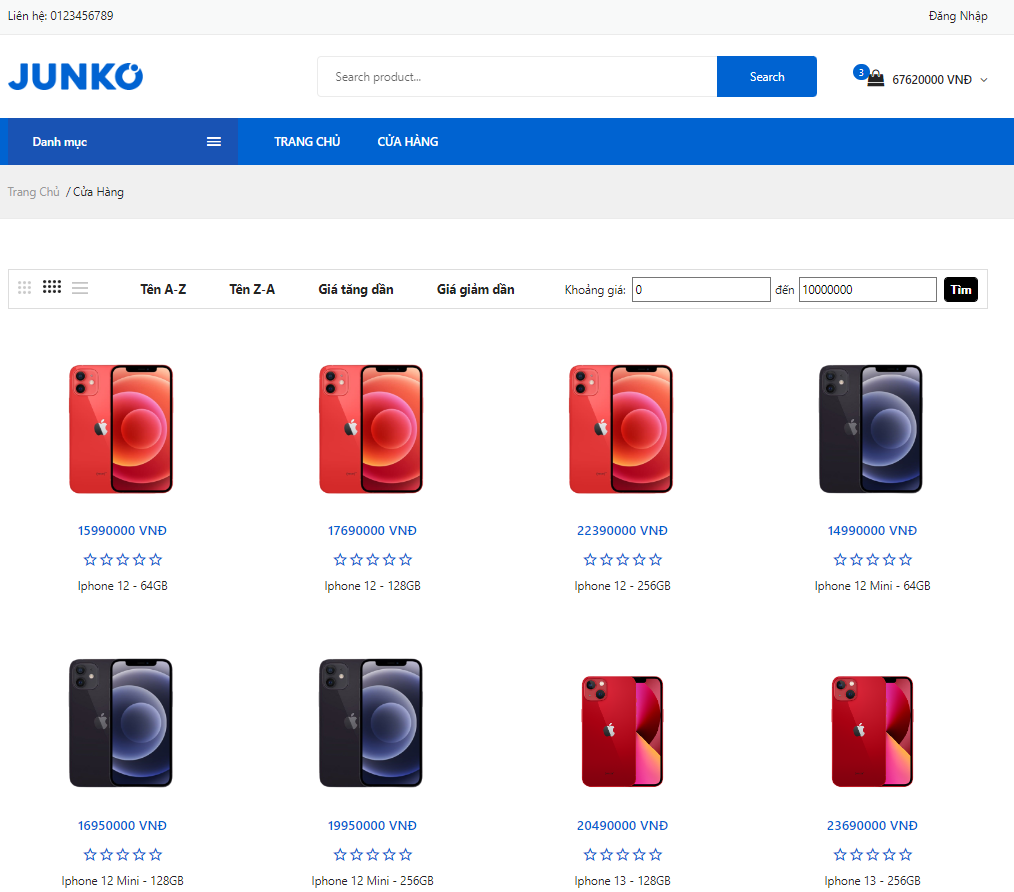
Hình ‑: Giao diện trang chủ

Trên thanh tìm kiếm, nếu ta nhập tên sản phẩm ở mức gần đúng và nhấn nút tìm kiếm, các sản phẩm thỏa mãn tìm kiếm sẽ hiện ra



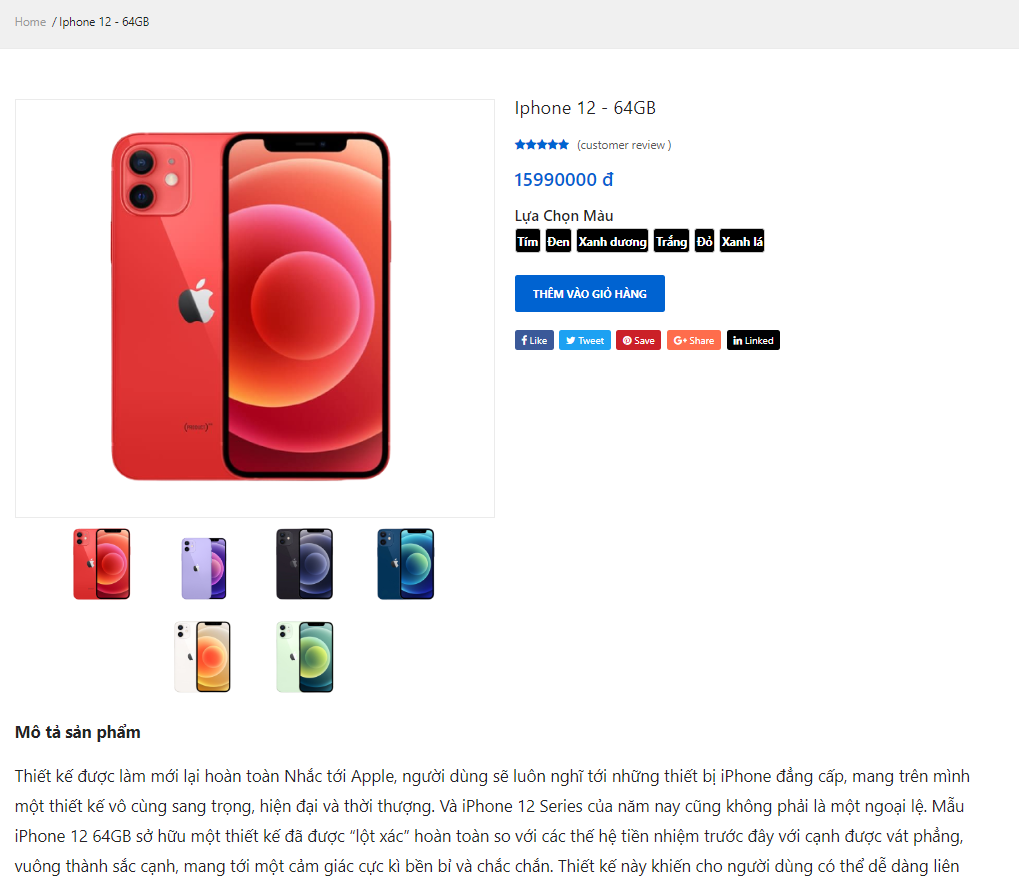
Hình ‑: Thanh tìm kiếm sản phẩm

Nhấn mục Cửa hàng sẽ đưa ta tới giao diện các sản phẩm



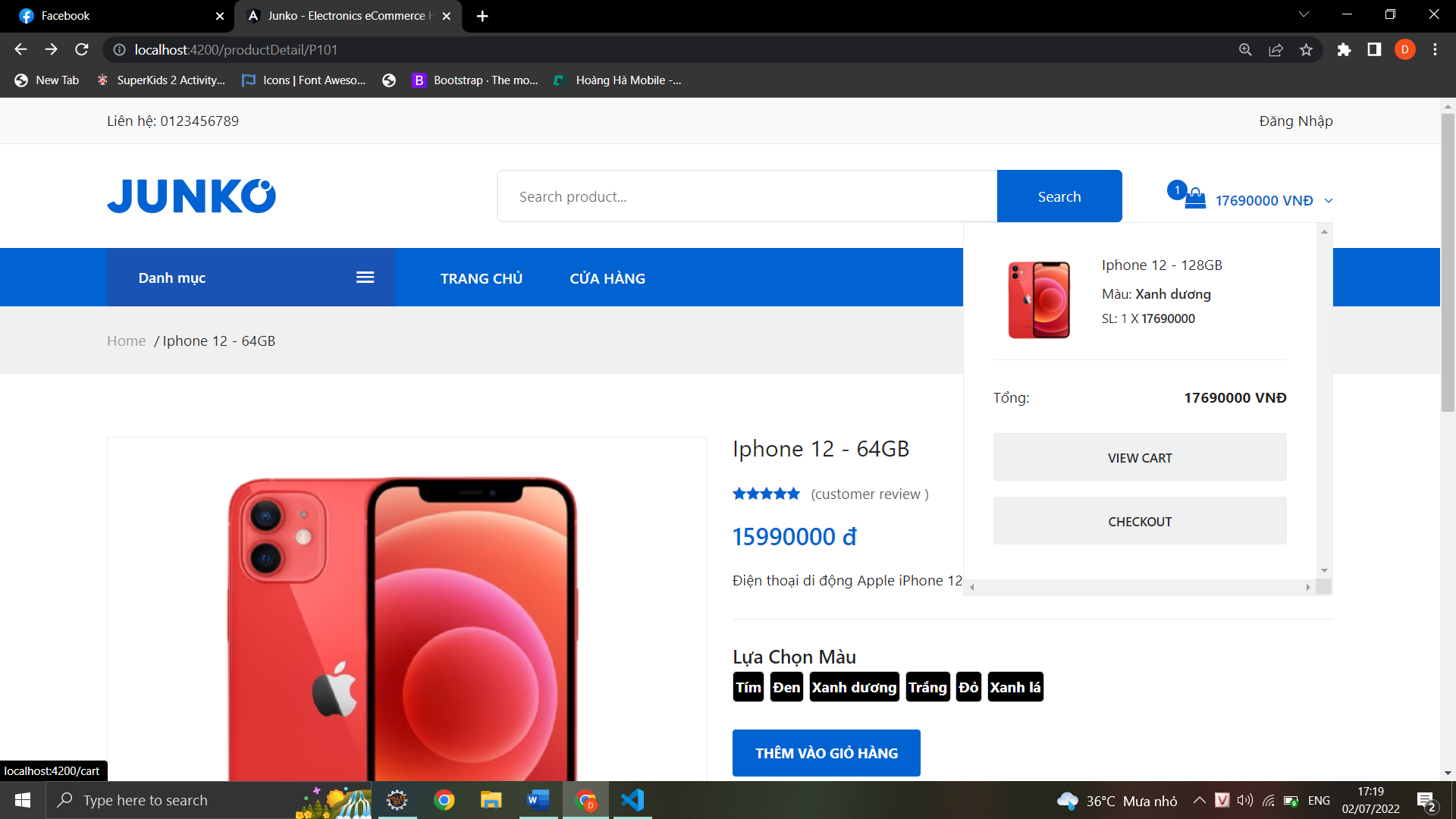
Hình ‑: Giao diện Cửa hàng

Chọn sản phẩm bất kỳ sẽ đưa ta tới giao diện chi tiết của sản phẩm ấy



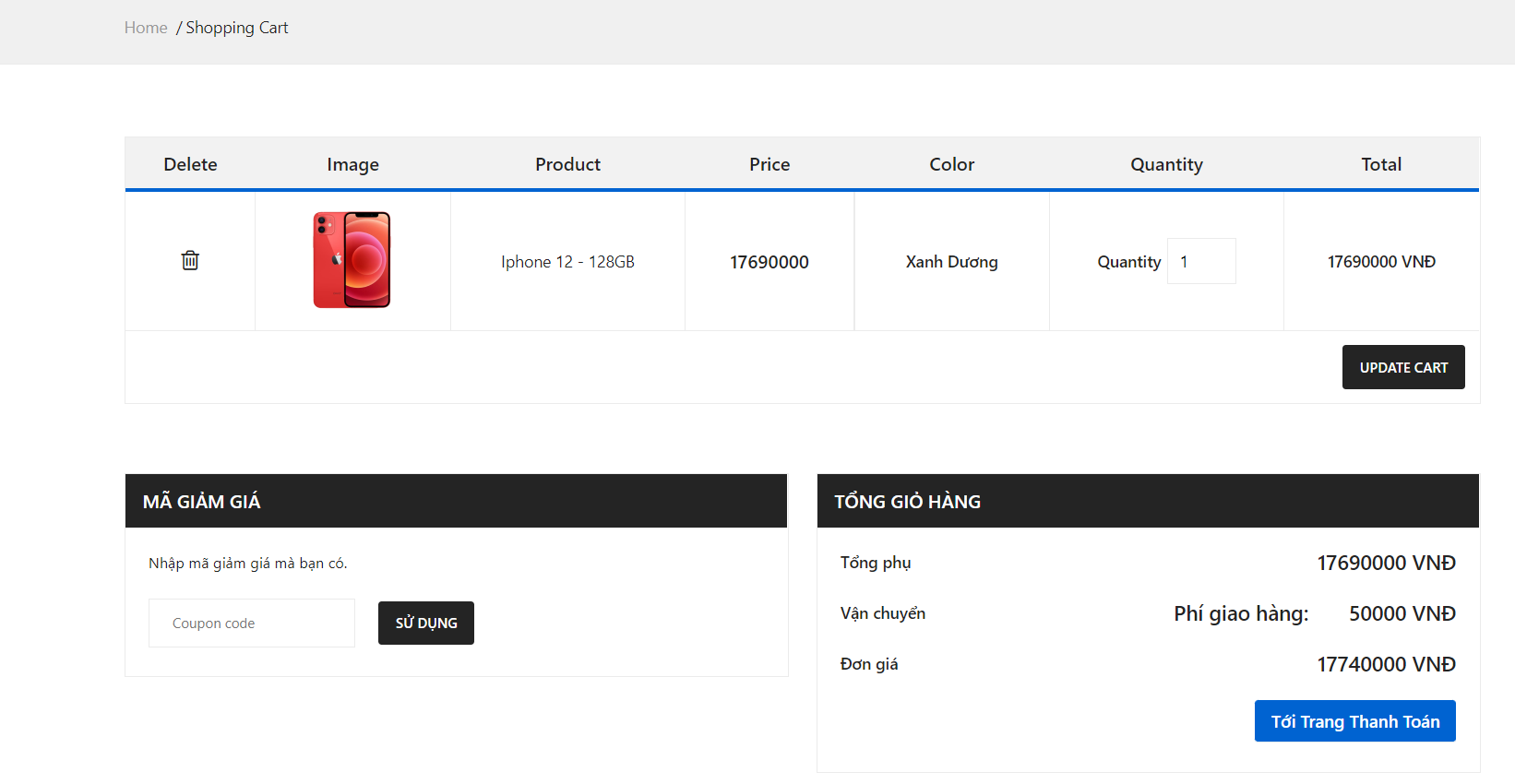
Hình ‑: Giao diện chio tiết sản phẩm

Sau khi chọn Thêm sản phẩm, sản phẩm sẽ được cập nhật trên giỏ hàng



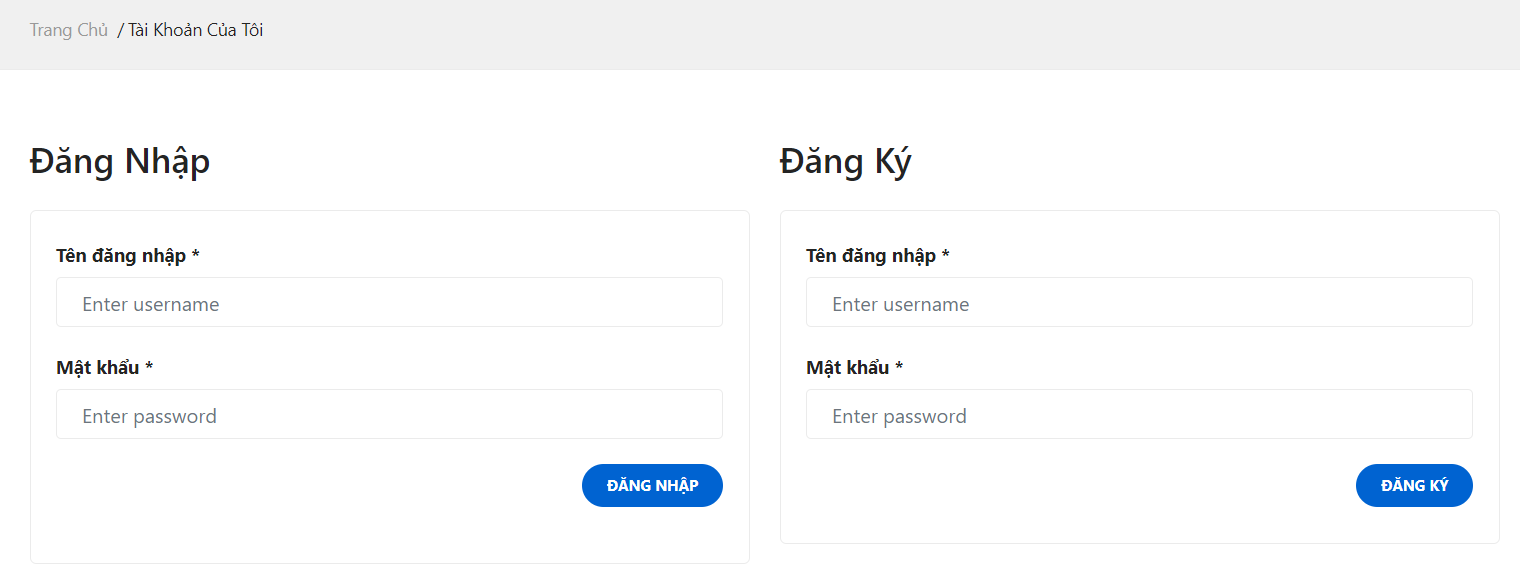
Hình ‑: Giao diện xem trước giỏ hàng khi thêm sản phẩm vào giỏ

Nhấn vào giỏ hàng ta sẽ tới Giao diện giỏ hàng



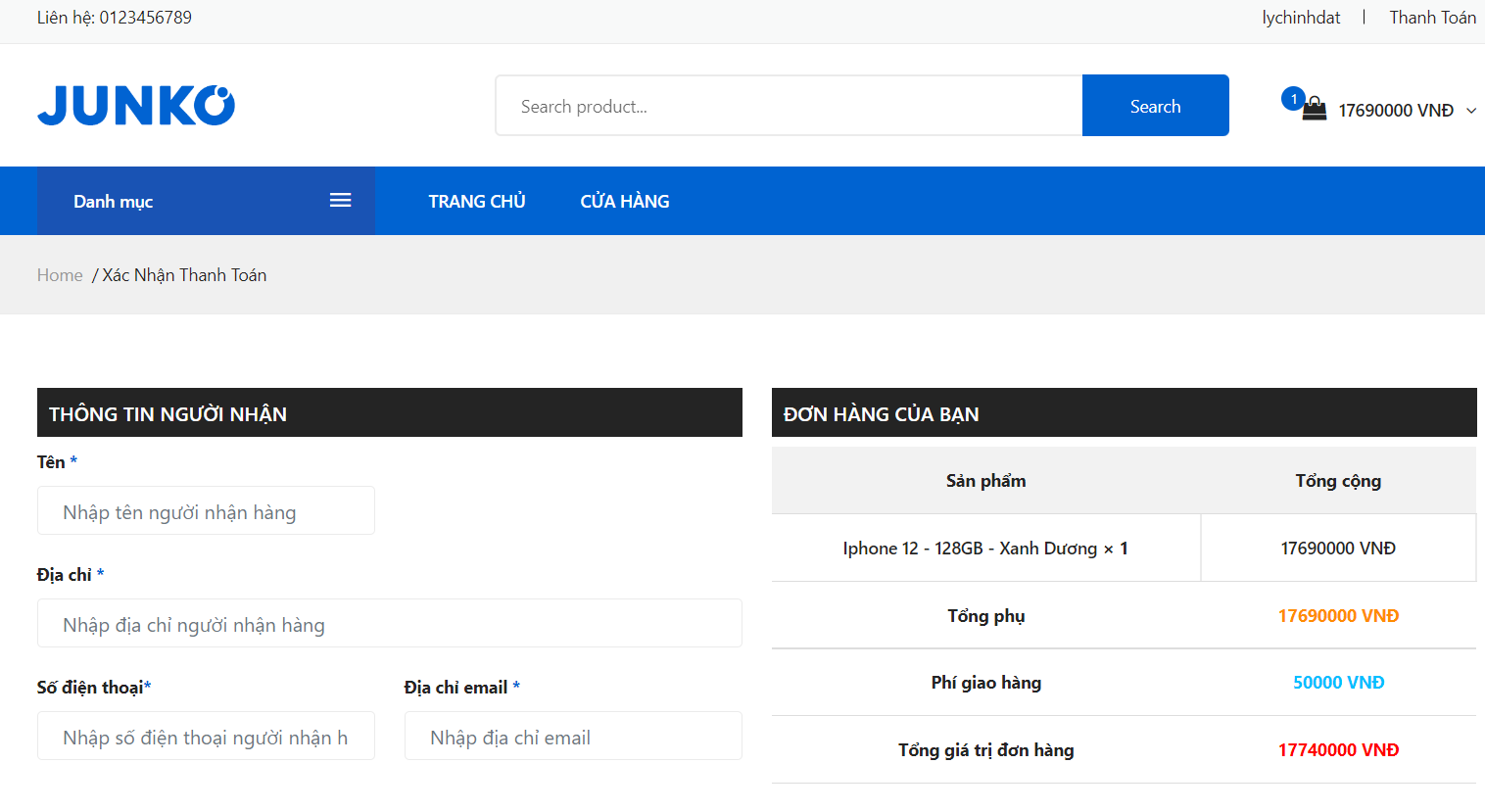
Hình ‑: Giao diện giỏ hàng

Nếu nhấn nút Tới trang thanh toán mà chưa đăng nhập, ta sẽ bị chuyển hướng tới trang Đăng nhập/Đăng ký



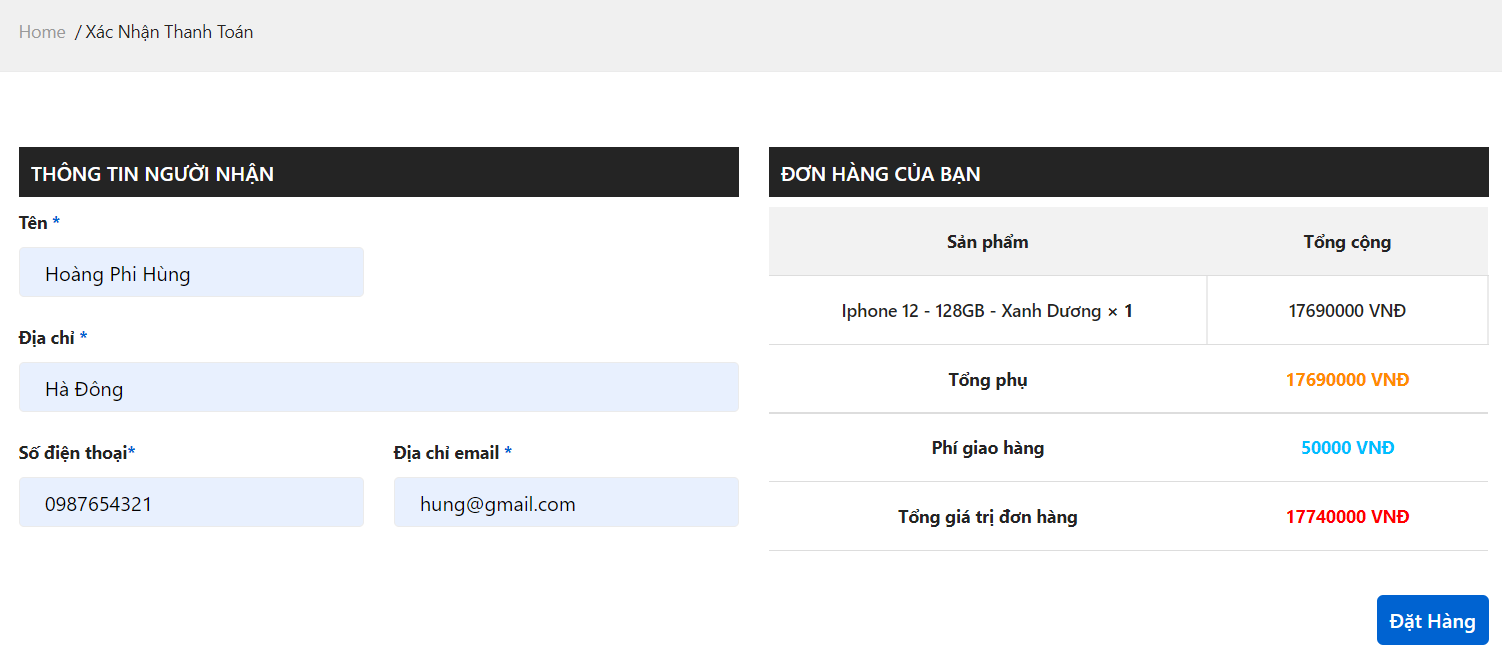
Hình ‑: Giao diện Đăng nhập/Đăng ký (bắt buộc phải đăng nhập để tới trang thanh toán)

Đăng nhập xong, ta có thể tới được trang Thanh toán



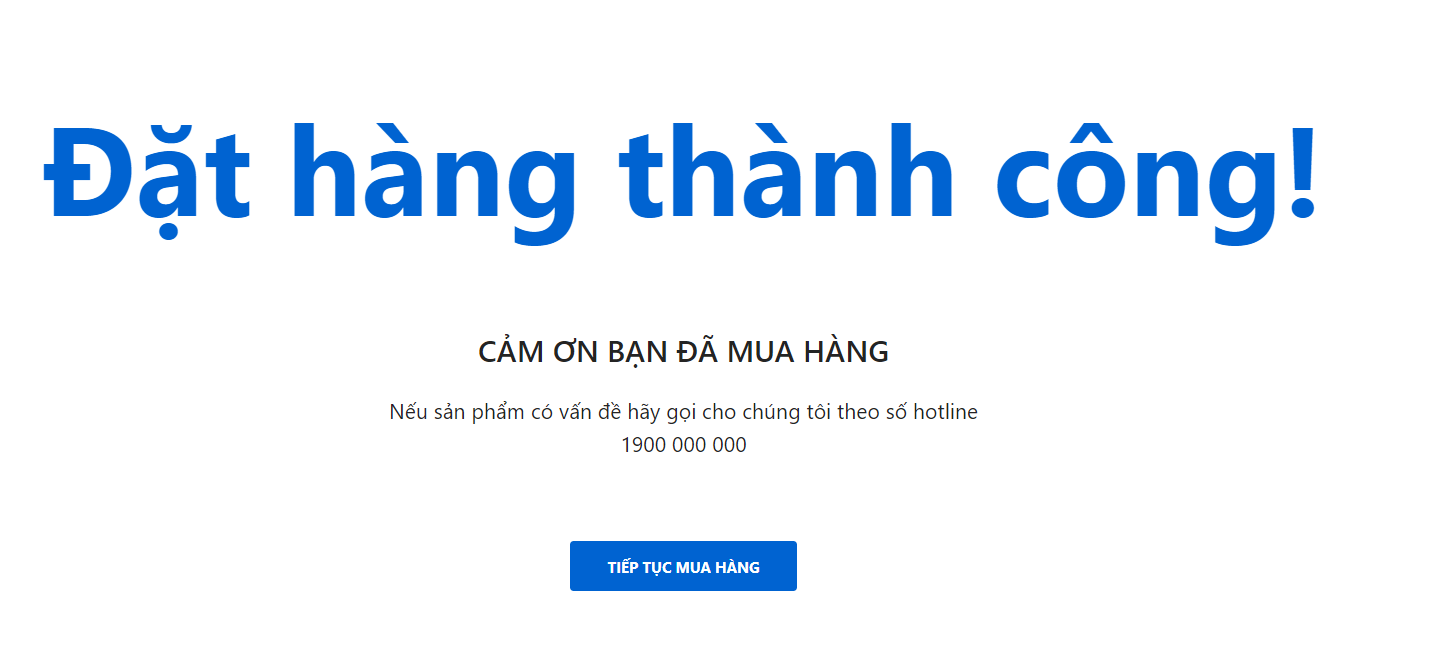
Hình ‑: Giao diện thanh toán sau khi đăng nhập xong

Nhập đầy đủ thông tin khách hàng và nhấn nút Đặt hàng

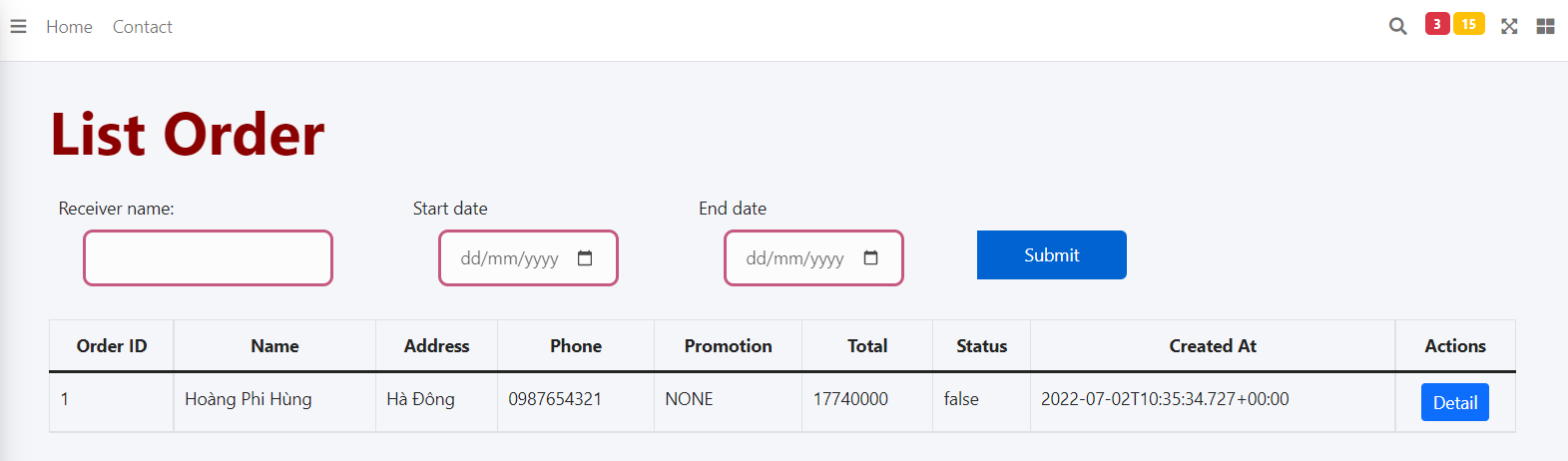


Hình ‑: Sau khi nhập thông tin đầy đủ, nút Đặt hàng sẽ hiện ra

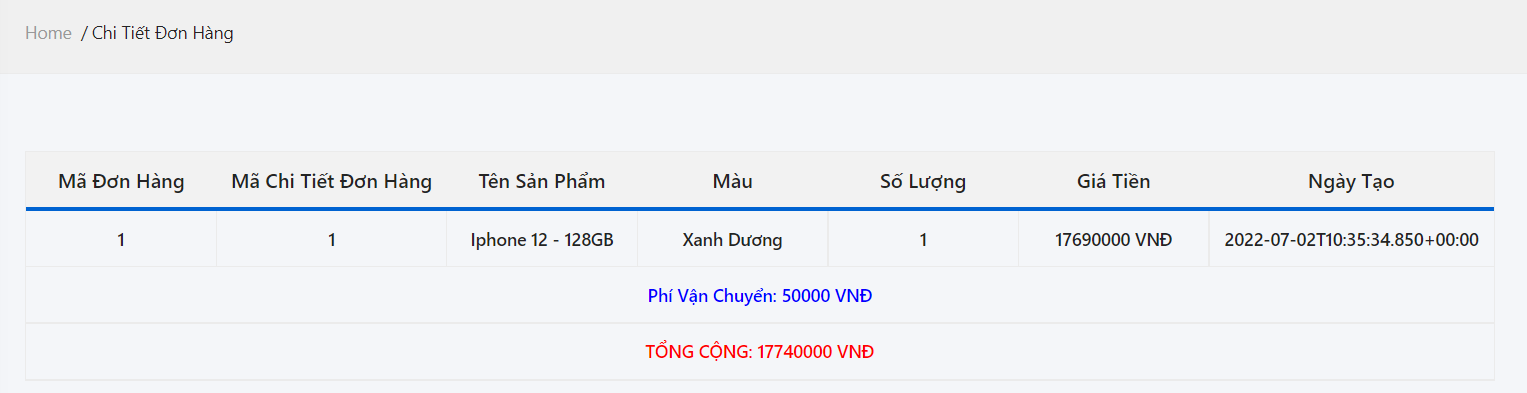
Giao diện xác nhận đã đặt hàng thành công



Hình ‑: Sau khi nhấn nút Đặt hàng sẽ chuyển tới trang xác nhận đặt hàng thành công



Hình ‑: Ngay lập tức thông tin sẽ được lưu vào cơ sở dữ liệu và hiển thị trong trang quản trị đơn hàng của admin



Hình ‑: Người quản trị có thể chọn nút Detail để xem đơn hàng chi tiết