|  |  |
| --- | --- |
| NGHIÊM VĂN TRƯỜNG NGÀNH KHOA HỌC MÁY TÍNH | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------**  ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  NGÀNH KHOA HỌC MÁY TÍNH  **Xây dựng website bán máy tính cho cửa hàng**  **PC Thành Công, tích hợp AI đề xuất mặt hàng tương tự**  **CBHD:** ThS. Vũ Minh Yến  **Sinh viên:** Nghiêm Văn Trường  **Mã sinh viên:** 2019602531    **Hà Nội – Năm 2023** |

LỜI NÓI ĐẦU

Để có được kết quả như ngày hôm nay, trước tiên em xin gửi lời cảm ơn sâu sắc nhất tới **ThS. Vũ Minh Yến**, giảng viên khoa Công nghệ thông tin Trường Đại học Công Nghiệp Hà Nội, đã tận tình hướng dẫn, giúp đỡ em trong suốt quá trình thực hiện đồ án tốt nghiệp.

Em cũng xin bày tỏ lòng biết ơn chân thành tới các thầy cô giáo trong Trường Đại học Công Nghiệp Hà Nội, đặc biệt là các thầy cô khoa Công nghệ thông tin, đã tận tụy dạy dỗ em trong quá trình học tập và rèn luyện suốt thời gian qua, giúp em có được những kiến thức không chỉ để hoàn thành báo cáo đồ án tốt nghiệp mà còn áp dụng vào công việc sau này.

Trong quá trình thực hiện đồ án tốt nghiệp, mặc dù đã cố gắng để hoàn thành tốt nhất có thể, nhưng vì thời gian chỉ vỏn vẹn 2 tháng cũng như những điều kiện hiện tại, chắc chắn em không thể tránh khỏi những thiếu sót. Vì vậy, em rất mong nhận được sự góp ý của thầy cô và các bạn để báo cáo của em được hoàn thiện hơn.

*Em xin chân thành cảm ơn!*

Hà Nội, ngày 15 tháng 05 năm 2023

Sinh viên

**Nghiêm Văn Trường**

MỤC LỤC

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 7](#_Toc135152724)

[1.1 Tìm hiểu về ngôn ngữ JAVA 7](#_Toc135152725)

[1.1.1 Khái niệm JAVA 7](#_Toc135152726)

[1.1.2 Lịch sử hình thành các phiên bản JAVA đã ra đời 7](#_Toc135152727)

[1.1.3 Ưu điểm của ngôn ngữ JAVA 8](#_Toc135152728)

[1.1.4 Một số hạn chế của ngôn ngữ JAVA 8](#_Toc135152729)

[1.1.5 Ứng dụng của ngôn ngữ JAVA 9](#_Toc135152730)

[1.2 Tìm hiểu về JavaScript 10](#_Toc135152731)

[1.2.1 Khái niệm JavaScript 10](#_Toc135152732)

[1.2.2 Lịch sử phát triển của JavaScript 11](#_Toc135152733)

[1.2.3 Thư viện và Framework của JavScript 11](#_Toc135152734)

[1.2.4 Ưu điểm và một số hạn chế của JavaScript 12](#_Toc135152735)

[1.3 Tìm hiểu về MySQL Workbench 13](#_Toc135152736)

[1.3.1 Khái niệm MySQL Workbench 13](#_Toc135152737)

[1.3.2 Lịch sử hình thành và phát triển của MySQL Workbench 13](#_Toc135152738)

[1.3.3 Tính năng của MySQL Workbench 14](#_Toc135152739)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 16](#_Toc135152740)

[2.1 Khảo sát hệ thống 16](#_Toc135152741)

[2.1.1 Giới thiệu 16](#_Toc135152742)

[2.1.2 Hiện trạng 17](#_Toc135152743)

[2.1.3 Đánh giá hiện trạng 17](#_Toc135152744)

[2.1.4 Yêu cầu hệ thống mới 17](#_Toc135152745)

[2.2 Biểu đồ usecase 21](#_Toc135152746)

[2.2.1 Biểu đồ usecase chính của hệ thống 21](#_Toc135152747)

[2.2.2 Mô tả các usecase trong hệ thống 22](#_Toc135152748)

[1. Mô tả usecase DangNhap (Đăng nhập) 22](#_Toc135152749)

[2. Mô tả usecase DangKy (Đăng ký) 22](#_Toc135152750)

[3. Mô tả usecase TimKiemSP (Tìm kiếm sản phẩm) 23](#_Toc135152751)

[4. Mô tả usecase ThemVaoGioHang (Thêm vào giỏ hàng) 24](#_Toc135152752)

[5. Mô tả usecase XemGioHang (Xem giỏ hàng) 24](#_Toc135152753)

[6. Mô tả usecase DatHang (Đặt hàng) 25](#_Toc135152754)

[7. Mô tả usecase XemDonHangDaDat (Xem đơn hàng đã đặt) 26](#_Toc135152755)

[8. Mô tả usecase XoaSanPhamTrongGioHang (Xóa sản phẩm trong giỏ hàng) 26](#_Toc135152756)

[9. Mô tả usecase XemSanPhamTrongDanhMuc (Xem sản phẩm trong danh mục) 27](#_Toc135152757)

[10. Mô tả usecase QuanLySanPham (Quản lý sản phẩm) 27](#_Toc135152758)

[11. Mô tả usecase QuanLyDanhMuc (Quản lý danh mục) 29](#_Toc135152759)

[12. Mô tả usecase QuanLyDonHang (Quản lý đơn hàng) 30](#_Toc135152760)

[2.3 Biểu đồ trình tự của usecase 33](#_Toc135152761)

[1. Usecase Đăng nhập 33](#_Toc135152762)

[2. Usecase Đăng ký 34](#_Toc135152763)

[3. Usecase Tìm kiếm sản phẩm 35](#_Toc135152764)

[4 Usecase Thêm vào giỏ hàng 35](#_Toc135152765)

[5. Usecase Xem giỏ hàng 36](#_Toc135152766)

[6. Usecase Đặt hàng 37](#_Toc135152767)

[7. Usecase Xem chi tiết đơn hàng đã đặt 38](#_Toc135152768)

[8. Usecase Xóa sản phẩm trong giỏ hàng 39](#_Toc135152769)

[9. Usecase Xem sản phẩm trong danh mục 40](#_Toc135152770)

[10. Usecase Quản lý sản phẩm 41](#_Toc135152771)

[11. Usecase Quản lý danh mục 42](#_Toc135152772)

[12. Usecase Quản lý đơn hàng 43](#_Toc135152773)

[2.5 Thiết kế cơ sở dữ liệu 44](#_Toc135152774)

[2.5.1 Danh sách các bảng cơ sở dữ liệu 44](#_Toc135152775)

[2.5.2 Bảng Category 44](#_Toc135152776)

[2.5.3 Bảng Sales 45](#_Toc135152777)

[2.5.4 Bảng Orders 45](#_Toc135152778)

[2.5.5 Bảng Products 46](#_Toc135152779)

[2.5.6 Bảng Attribute 46](#_Toc135152780)

[2.5.7 Bảng Accounts 47](#_Toc135152781)

[2.5.8 Bảng Account\_Detail 47](#_Toc135152782)

[2.5.9 Bảng Order\_Detail 47](#_Toc135152783)

[2.5.10 Bảng Images 48](#_Toc135152784)

[2.5.11 Bảng Brands 48](#_Toc135152785)

[2.5.12 Bảng Notifications 48](#_Toc135152786)

[2.5.13 Bảng Cart\_Item 49](#_Toc135152787)

[2.5.14 Bảng Order\_Status 49](#_Toc135152788)

[2.5.15 Bảng Roles 49](#_Toc135152789)

[2.5.16 Bảng Vouchers 50](#_Toc135152790)

[2.5.8 Mô hình cơ sở dữ liệu 51](#_Toc135152791)

[2.6 Thiết kế giao diện 52](#_Toc135152792)

[2.6.1 Giao diện người dùng 52](#_Toc135152793)

[2.6.2 Giao diện Admin 58](#_Toc135152794)

[Chương 3. Cài Đặt và kiểm thử chương trình 59](#_Toc135152795)

[KẾT LUẬN 60](#_Toc135152796)

[TÀI LIỆU THAM KHẢO 61](#_Toc135152797)

DANH MỤC HÌNH ẢNH

[Hình 2. 1 Sơ đồ tổ chức của cửa hàng 16](file:///D:\Document_Study\MySchool\Nam4\Ky%20II\DATN\BaoCao_DATN_NghiemVanTruong_2019602531.docx#_Toc134956175)

[Hình 2. 2 Biểu đồ usecase tổng quát của hệ thống 21](#_Toc134956176)

[Hình 2. 3 Biểu đồ trình tự usecase Đăng nhập 33](file:///D:\Document_Study\MySchool\Nam4\Ky%20II\DATN\BaoCao_DATN_NghiemVanTruong_2019602531.docx#_Toc134956177)

[Hình 2. 4 Biểu đồ trình tự usecase Đăng ký 34](#_Toc134956178)

[Hình 2. 5 Biểu đồ trình tự usecase Tìm kiếm sản phẩm 35](#_Toc134956179)

[Hình 2. 6 Biểu đồ trình tự usecase Thêm vào giỏ hàng 35](#_Toc134956180)

[Hình 2. 7 Biểu đồ trình tự usecase Xem giỏ hàng 36](#_Toc134956181)

[Hình 2. 8 Biểu đồ trình tự usecase Xem chi tiết đơn hàng đã đặt 38](#_Toc134956182)

[Hình 2. 9 Biểu đồ trình tự usecase Xóa sản phẩm trong giỏ hàng 39](#_Toc134956183)

[Hình 2. 10 Biểu đồ trình tự usecase Xem danh sách sản phẩm trong danh mục 40](#_Toc134956184)

[Hình 2. 11 Biểu đồ trình tự usecase Quản lý sản phẩm 41](#_Toc134956185)

[Hình 2. 12 Biểu đồ trình tự usecase Quản lý danh mục 42](#_Toc134956186)

[Hình 2. 13 Biểu đồ trình tự usecase Quản lý đơn hàng 43](#_Toc134956187)

[Hình 2. 14 Màn hình đăng nhập 51](#_Toc134956188)

[Hình 2. 15 Màn hình trang chủ 52](#_Toc134956189)

[Hình 2. 16 Màn hình danh mục 53](#_Toc134956190)

DANH MỤC BẢNG BIỂU

[Bảng 2. 1 Bảng chức năng quản lý của chủ cửa hàng 19](#_Toc134952879)

[Bảng 2. 2 Bảng chức năng của người dùng 20](#_Toc134952880)

[Bảng 2. 3 Danh sách các bảng cơ sở dữ liệu 44](#_Toc134952881)

[Bảng 2. 4 Bảng Category 44](#_Toc134952882)

[Bảng 2. 5 Bảng News 45](#_Toc134952883)

[Bảng 2. 6 Bảng Orders 45](#_Toc134952884)

[Bảng 2. 7 Bảng Products 46](#_Toc134952885)

[Bảng 2. 8 Bảng Attribute 46](#_Toc134952886)

[Bảng 2. 9 Bảng Accounts 47](#_Toc134952887)

[Bảng 2. 10 Bảng Account\_Detail 47](#_Toc134952888)

[Bảng 2. 11 Bảng Order\_Detail 47](#_Toc134952889)

[Bảng 2. 12 Bảng Images 48](#_Toc134952890)

[Bảng 2. 13 Bảng Brands 48](#_Toc134952891)

[Bảng 2. 14 Bảng Notifications 48](#_Toc134952892)

[Bảng 2. 15 Bảng Cart\_Item 49](#_Toc134952893)

[Bảng 2. 16 Bảng Order\_Status 49](#_Toc134952894)

[Bảng 2. 17 Bảng Roles 49](#_Toc134952895)

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## 1.1 Tìm hiểu về ngôn ngữ JAVA

### 1.1.1 Khái niệm JAVA

* 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 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 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.

### 1.1.2 Lịch sử hình thành các phiên bản JAVA đã ra đời

* James Gosling, Mike Sheridan và Patrick Naughton khởi xướng dự án ngôn ngữ Java vào tháng 6 năm 1991. Java ban đầu được thiết kế cho truyền hình tương tác, nhưng nó quá tiên tiến đối với ngành truyền hình cáp kỹ thuật số vào thời điểm đó.
* Sun Microsystems đã phát hành bản triển khai công khai đầu tiên dưới dạng Java 1.0 vào năm 1996. Nó hứa hẹn khả năng viết một lần, chạy mọi nơi (WORA), cung cấp thời gian chạy miễn phí trên các nền tảng phổ biến.
* Vào ngày 13 tháng 11 năm 2006, Sun đã phát hành phần lớn máy ảo Java (JVM) của mình dưới dạng phần mềm mã nguồn mở và miễn phí (FOSS), theo các điều khoản của Giấy phép Công cộng GNU (GPL).
* Vào ngày 8 tháng 5 năm 2007, Sun đã hoàn thành quá trình, cung cấp tất cả mã cốt lõi của JVM theo các điều khoản phân phối phần mềm miễn phí / nguồn mở, ngoại trừ một phần nhỏ mã mà Sun không giữ bản quyền.
* Tính đến thời điểm hiện tại, Java SE 17 là bản phát hành mới nhất và được phát hành vào ngày 14 tháng 9 năm 2021.

### 1.1.3 Ưu điểm của ngôn ngữ JAVA

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

### 1.1.4 Một số hạn chế của ngôn ngữ JAVA

Mặc dù có nhiều điểm mạnh nhưng JAVA cũng sở hữu những điểm yếu cần khắc phục:

* Java thể hiện hiệu suất kém, nguyên nhân chính là do bộ thu gom rác, cấu hình bộ nhứ đệm không hợp lệ và bế tắc giữa các quy trình.
* Có rất ít trình tạo GUI – Swing, SWT, JSF và JavaFX trong số những trình xây dựng phổ biến hơn.
* Để viết mã thực hiện một tập hợp các hoạt động đơn giản, chúng ta có thể phải viết những đoạn mã dài và phức tạp. So với Pyhton ta nhận thấy Python không cần dấu chấm phẩy, dấu ngoặc đơn hoặc dấu ngoặc nhọn và có mã ngắn hơn rất nhiều.

### 1.1.5 Ứng dụng của ngôn ngữ JAVA

* Các ứng dụng yêu cầu tính bảo mật cao
* Các ứng dụng sử dụng trong giao dịch của các ngân hàng yêu cầu tính bảo mật cao. Họ cần bảo mật tối đa thông tin tài khoản của khách hàng. Trên thế giới đã có các ngân hàng sử dụng Java để viết các hệ thống giao dịch điện tử như Goldman Sachs, Citigroup, Barclays, Standard Chartered … Ngoài viết các ứng dụng giao dịch, họ còn sử dụng Java trong việc khác như hệ thống xác nhận và kiểm toán, các dự án xử lý dữ liệu …
* Ứng dụng của ngôn ngữ lập trình Java
* Ngôn ngữ lập trình Java được đánh giá là một ngôn ngữ có độ bảo mật cao. Chính vì vậy, nó thường được sử dụng vào các ứng dụng của ngành dịch vụ tài chính hay ngân hàng.
* Các ứng dụng cho hệ điều hành Android
* Java hỗ trợ tối đa cho hệ điều hành Android. Vì thế ngôn ngữ lập trình này được áp dụng rất nhiều vào các ứng dụng dành cho Android.. Số người sử dụng Android chiếm hơn một nửa thị phần vì vậy có thể thấy, nhu cầu sử dụng các ứng dụng là rất lớn.
* Nếu trong tay bạn đang cầm một chiếc điện thoại chạy trên hệ điều hành Android thì đừng bất ngờ, bất cứ một ứng dụng nào trên đó cũng đều được hình thành và phát triển trên nền tảng Java. Chính điều này đã tạo ra cơ hội lớn cho các lập trình viên Java: nếu họ học tốt và có kỹ năng thì chắc chắn những lập trình Java sẽ không bao giờ sợ thất nghiệp.
* Điện toán đám mây
* Cũng giống như Internet vạn vật (IoT), điện toán đám mây là một miếng bánh béo bở mà bất kỳ công ty nào cũng muốn chiếm thị phần. Java với tính năng di động của nó sẽ là chìa khóa giúp bạn quản lý các giải pháp điện toán đám mây. Ngoài ra, Java là ngôn ngữ lập trình đa mục đích. Nó cho phép bạn làm việc trên ứng dụng di động, máy tính để bàn và ứng dụng đám mây. Ví dụ, Heroku cung cấp một nền tảng đám mây sử dụng Java. Ngoài ra, có rất nhiều PaaS giúp bạn đi sâu vào điện toán đám mây. Cả Google Cloud Platform và Microsoft Azure đều có các điều khoản để lưu trữ các ứng dụng Java và quản lý chúng một cách hiệu quả.

## 1.2 Tìm hiểu về JavaScript

### 1.2.1 Khái niệm JavaScript

JavaScript thường được viết tắt là JS, là [ngôn ngữ lập trình](https://en.wikipedia.org/wiki/Programming_language) là một trong những công nghệ cốt lõi của world wide web bên cạnh [HTML](https://en.wikipedia.org/wiki/HTML) và [CSS](https://en.wikipedia.org/wiki/CSS). Kể từ năm 2022, 98% [trang web](https://en.wikipedia.org/wiki/Website) sử dụng JavaScript ở phía [máy khách](https://en.wikipedia.org/wiki/Client_(computing)) cho hành vi [của trang web, thường kết hợp](https://en.wikipedia.org/wiki/Web_page)[các thư viện](https://en.wikipedia.org/wiki/Library_(computing)) của bên thứ ba. [Tất cả các trình duyệt web](https://en.wikipedia.org/wiki/Web_browser) chính đều có một [công cụ JavaScript](https://en.wikipedia.org/wiki/JavaScript_engine) chuyên dụng để thực thi [mã](https://en.wikipedia.org/wiki/Source_code) trên thiết bị của [người dùng](https://en.wikipedia.org/wiki/User_(computing)) .

JavaScript là một ngôn ngữ [cấp cao](https://en.wikipedia.org/wiki/High-level_programming_language), thường [được biên dịch đúng lúc](https://en.wikipedia.org/wiki/Just-in-time_compilation), phù hợp với tiêu chuẩn [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript). Nó có [kiểu gõ động](https://en.wikipedia.org/wiki/Dynamic_typing), [hướng đối tượng](https://en.wikipedia.org/wiki/Object-oriented_programming)[dựa trên nguyên mẫu](https://en.wikipedia.org/wiki/Prototype-based_programming) và [các hàm hạng nhất](https://en.wikipedia.org/wiki/First-class_function). Nó là [đa mô hình](https://en.wikipedia.org/wiki/Programming_paradigm), hỗ trợ [các phong cách lập trình](https://en.wikipedia.org/wiki/Programming_paradigm)[hướng sự kiện](https://en.wikipedia.org/wiki/Event-driven_programming), [chức năng](https://en.wikipedia.org/wiki/Functional_programming) và [mệnh lệnh](https://en.wikipedia.org/wiki/Imperative_programming). Nó có [các giao diện lập trình ứng dụng](https://en.wikipedia.org/wiki/Application_programming_interface) (API) để làm việc với văn bản, ngày tháng, [biểu thức chính quy](https://en.wikipedia.org/wiki/Regular_expression), [cấu trúc dữ liệu](https://en.wikipedia.org/wiki/Data_structure) tiêu chuẩn và [Mô hình đối tượng tài liệu](https://en.wikipedia.org/wiki/Document_Object_Model) (DOM).

Tiêu chuẩn ECMAScript không bao gồm bất kỳ [đầu vào/đầu ra](https://en.wikipedia.org/wiki/Input/output) (I/O) nào, chẳng hạn như thiết bị [mạng](https://en.wikipedia.org/wiki/Computer_network), [lưu trữ](https://en.wikipedia.org/wiki/Data_storage) hoặc [đồ họa](https://en.wikipedia.org/wiki/Computer_graphics). Trên thực tế, trình duyệt web hoặc [hệ thống thời gian chạy](https://en.wikipedia.org/wiki/Runtime_system) khác cung cấp API JavaScript cho I/O.

[Công cụ JavaScript](https://en.wikipedia.org/wiki/JavaScript_engines) ban đầu chỉ được sử dụng trong các trình duyệt web, nhưng giờ đây là thành phần cốt lõi của một số [máy chủ](https://en.wikipedia.org/wiki/Server_(computing)) và nhiều [ứng dụng](https://en.wikipedia.org/wiki/Application_software) khác nhau. Hệ thống thời gian chạy phổ biến nhất cho cách sử dụng này là [Node.js](https://en.wikipedia.org/wiki/Node.js) .

Mặc dù [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) và JavaScript giống nhau về tên, [cú pháp](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) và [các thư viện tiêu chuẩn](https://en.wikipedia.org/wiki/Standard_library) tương ứng , hai ngôn ngữ này khác biệt và khác nhau rất nhiều về thiết kế.

### 1.2.2 Lịch sử phát triển của JavaScript

Trình duyệt web phổ biến đầu tiên có giao diện người dùng đồ họa, Mosaic được phát hành vào năm 1993. Có thể truy cập được đối với những người không có kỹ thuật, nó đóng một vai trò nổi bật trong sự phát triển nhanh chóng của World Wide Web mới ra đời. Các nhà phát triển hàng đầu của Mosaic sau đó đã thành lập tập đoàn Netscape, công ty này đã phát hành một trình duyệt bóng bẩy hơn, Netscape Navigator, vào năm 1994. Trình duyệt này nhanh chóng trở thành trình duyệt được sử dụng nhiều nhất.

Trong những năm hình thành này của Web, các trang web chỉ có thể ở trạng thái tĩnh, thiếu khả năng hoạt động động sau khi trang được tải trong trình duyệt. Trong bối cảnh phát triển web đang phát triển mạnh mẽ, có một mong muốn loại bỏ hạn chế này, vì vậy vào năm 1995, Netscape đã quyết định thêm một ngôn ngữ kịch bản vào Navigator. Họ theo đuổi hai con đường để đạt được điều này hợp tác với Sun Microsystems để nhúng ngôn ngữ lập trình Java, đồng thời thuê Brendan Eich để nhúng ngôn ngữ Scheme.

Ban quản lý Netscape sớm quyết định rằng lựa chọn tốt nhất là để Eich nghĩ ra một ngôn ngữ mới, với cú pháp tương tự như Java và ít giống Scheme hoặc các ngôn ngữ kịch bản hiện có khác. Mặc dù ngôn ngữ mới và triển khai trình thông dịch của nó được gọi là LiveScript khi lần đầu tiên được vận chuyển như một phần của bản Navigator beta vào tháng 9 năm 1995, tên này đã được đổi thành JavaScript để phát hành chính thức vào tháng 12.

### 1.2.3 Thư viện và Framework của JavScript

Hơn 80% trang web sử dụng thư viện JavaScript của bên thứ ba hoặc khung web cho tập lệnh phía máy khách của họ.

jQuery cho đến nay là thư viện phổ biến nhất, được hơn 75% trang web sử dụng. Facebook đã tạo thư viện React cho trang web của mình và sau đó phát hành dưới dạng mã nguồn mở các trang web khác, bao gồm cả Twitter , hiện đang sử dụng nó. Tương tự như vậy, khung Angular do Google tạo cho các trang web của mình, bao gồm YouTube và Gmail, hiện là một dự án nguồn mở được những người khác sử dụng.

### 1.2.4 Ưu điểm và một số hạn chế của JavaScript

* **Ưu điểm của việc sử dụng JavaScript**
* Ít tương tác với máy chủ, chúng ta có thể xác thực đầu vào của người dùng trước khi gửi trang đến máy chủ. Điều này giúp tiết kiệm lưu lượng máy chủ, có nghĩa là tải ít hơn trên máy chủ của bạn.
* Phản hồi ngay lập tức cho khách truy cập, họ không phải đợi tải lại trang để xem họ có quên nhập nội dung nào không.
* Tăng tính tương tác chúng ta có thể tạo các giao diện phản ứng khi người dùng di chuột qua chúng hoặc kích hoạt chúng thông qua bàn phím.
* Giao diện phong phú hơn chúng ta có thể sử dụng JavaScript để bao gồm các mục như các thành phần kéo và thả để trượt giao diện phong phú cho khách truy cập trang web của bạn.
* **Hạn chế của JavaScript**
* Js phía máy khách không cho phép đọc hoặc ghi tệp. Điều này đã được giữ vì lý do an ninh.
* Js không thể được sử dụng cho các ứng dụng mạng vì không có hỗ trợ như vậy.
* Js không có bất kỳ khả năng đa luồng hoặc đa xử lý nào.
* Một lần nữa, Js là ngôn ngữ lập trình nhẹ, được giải thích cho phép bạn xây dựng tính tương tác vào các trang HTML tĩnh.

## 1.3 Tìm hiểu về MySQL Workbench

### 1.3.1 Khái niệm MySQL Workbench

MySQL Workbench là một công cụ thiết kế cơ sở dữ liệu trực quan tích hợp việc phát triển, quản trị, thiết kế cơ sở dữ liệu, tạo và bảo trì SQL vào một môi trường phát triển tích hợp duy nhất cho hệ thống cơ sở dữ liệu MySQL . Nó là phiên bản kế thừa của DBDesigner 4 từ fabFORCE.net và thay thế gói phần mềm trước đó, MySQL GUI Tools Bundle.

### 1.3.2 Lịch sử hình thành và phát triển của MySQL Workbench

DBDesigner4 là một công cụ truy vấn và thiết kế cơ sở dữ liệu trực quan mã nguồn mở cho cơ sở dữ liệu MySQL được phát hành theo GPL. Nó được lập trình viên người Áo Michael G. Zinner viết vào năm 2002/2003 cho nền tảng fabFORCE.net của anh ấy bằng Delphi 7/Kylix 3.

Mặc dù là công cụ duy nhất để tạo mô hình vật lý nhưng DBDesigner4 cung cấp bộ tính năng toàn diện bao gồm kỹ thuật đảo ngược cơ sở dữ liệu MySQL, đồng bộ hóa mô hình với cơ sở dữ liệu, in áp phích mô hình, kiểm soát phiên bản cơ bản của mô hình lược đồ và trình tạo truy vấn SQL. Nó có sẵn cho MS Windows, Mac OS X và Linux.

Cuối năm 2003, Zinner được các đại diện của MySQL AB tiếp cận và gia nhập công ty để đảm nhận việc phát triển các công cụ giao diện người dùng đồ họa (GUI) cho MySQL. Điều này dẫn đến việc tạo Gói công cụ GUI của MySQL

**Phát hành của MySQL Workbench**

Phiên bản xem trước đầu tiên của MySQL Workbench được phát hành vào tháng 9 năm 2005, và không được bao gồm trong Gói công cụ GUI của MySQL. Quá trình phát triển được bắt đầu lại vào năm 2007 và MySQL Workbench được thiết lập để trở thành sản phẩm chủ lực của MySQL GUI.

Việc đánh số phiên bản được bắt đầu từ 5.0 để nhấn mạnh rằng MySQL Workbench được phát triển như là phiên bản kế nhiệm của DBDesigner4

MySQL Workbench 5.0 và 5.1

MySQL Workbench 5.2

MySQL Workbench 6.0 (22/05/2013)

MySQL Workbench 6.1 (23/01/2014)

MySQL Workbench 6.2 (19/08/2014)

MySQL Workbench 6.3 (05/03/2015)

MySQL Workbench 8.0 (05/04/2018)

### 1.3.3 Tính năng của MySQL Workbench

Các tính năng nổi bật của MySQL Workbench là:

* Tổng quan
* Kết nối cơ sở dữ liệu & quản lý phiên bản
* Các mục hành động do thuật sĩ điều khiển
* Hoàn toàn có thể viết kịch bản với Python và Lua
* Hỗ trợ cho các plugin tùy chỉnh
* Tuân thủ MSAA (Windows Accessibility API)
* Hỗ trợ các tính năng của MySQL Enterprise (Nhật ký kiểm tra, Tường lửa và Sao lưu doanh nghiệp)
* Trình soạn thảo SQL
* Duyệt, kiểm tra và tìm kiếm đối tượng lược đồ
* Công cụ đánh dấu cú pháp SQL và trình phân tích cú pháp câu lệnh
* Hoàn thành mã SQL và trợ giúp theo ngữ cảnh
* Nhiều tập hợp kết quả có thể chỉnh sửa
* GIẢI THÍCH bằng hình ảnh
* Bộ sưu tập đoạn mã SQL
* Đường hầm kết nối SSH
* Hỗ trợ Unicode
* Mô hình hóa dữ liệu
* Sơ đồ ER
* Mô hình trực quan Drag'n'Drop
* Kỹ thuật đảo ngược từ Tập lệnh SQL và cơ sở dữ liệu trực tiếp
* Chuyển tiếp kỹ thuật sang SQL Scripts và cơ sở dữ liệu trực tiếp
* Đồng bộ hóa lược đồ
* In ấn các mô hình
* Nhập từ fabFORCE.net DBDesigner4
* Quản lý cơ sở dữ liệu
* Bắt đầu và dừng phiên bản cơ sở dữ liệu
* Cấu hình phiên bản
* Quản lý tài khoản cơ sở dữ liệu
* Duyệt biến thể hiện
* Trình duyệt tệp nhật ký
* Xuất/nhập kết xuất dữ liệu
* Giám sát hiệu suất
* Số liệu lược đồ hiệu suất
* Bảng điều khiển phiên bản MySQL
* Thống kê truy vấn
* Di chuyển cơ sở dữ liệu
* Bất kỳ cơ sở dữ liệu tuân thủ ODBC nào
* Hỗ trợ gốc: Microsoft SQL Server, PostgreSQL, MySQL Anywhere, SQLite và Sybase ASE

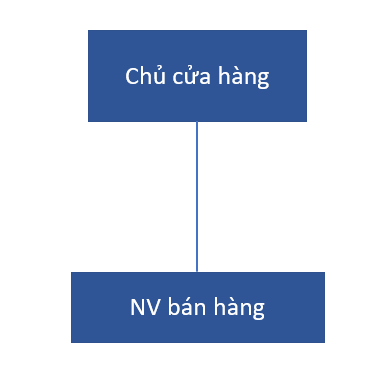
# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1 Khảo sát hệ thống

### 2.1.1 Giới thiệu

Cửa hàng PC Thành Công được thành lập năm 2010 tại 76 Núi Đôi, Sóc Sơn, Hà Nội. Với mô hình kinh doanh nhỏ lẻ và chỉ quảng cáo sản phẩm qua hình thức trực tiếp tại cửa hàng vẫn chưa lan tỏa được hình ảnh và thương hiệu của cửa hàng rộng rãi. Người mua hàng chủ yếu là những khách quen đã từng ghé thăm cửa hàng và số lượng khách đến còn khá thưa thớt. Vì vậy chủ cửa hàng đã quyết định nâng cấp cửa hàng để có thể lan tỏa tới tất cả với mọi người về sản phẩm của cửa hàng.

* Sơ đồ tổ chức của cửa hàng:



Hình 2. Sơ đồ tổ chức của cửa hàng

### 2.1.2 Hiện trạng

Khi khách hàng muốn mua sản phẩm thì cần phải trực tiếp qua cửa hàng để có thể lựa chọn, thanh toán sản phẩm. Chính vì vậy có những khách hàng tới cửa hàng chỉ là khách hàng trong khu vực địa phương và các khách thân quen.

- Khi khách hàng có nhu cầu mua sản phẩm nào đó thì người dùng bán hàng sẽ tư vấn và đưa ra các mẫu sản phẩm để khách hàng lựa chọn, khi khách hàng đã lựa chọn thì người dùng tiến hành lập hóa đơn và đem hóa đơn để đi lấy sản phẩm trong cửa hàng (hoặc trong kho). Sau khi lấy các sản phẩm có trong hóa đơn thì người dùng lắp ráp thiết bị sẽ lắp hoàn chỉnh (nếu khách hàng có nhu cầu). Tiếp theo là tiến hành thanh toán cho khách hàng.

### 2.1.3 Đánh giá hiện trạng

Hệ thống cửa hàng hiện tại vẫn còn rất còn ít khách đến mua sản phẩm bởi chưa thu hút được khách hàng ngoài địa phương, truyền thông quảng bá hình ảnh cửa hàng vẫn còn hạn chế. Các thông tin về các sản phẩm thì chưa tới được khách hàng tiềm năng, và những chương trình khuyến mãi thì chỉ có thể quảng cáo trong phạm vi nhỏ hẹp bằng hình thức quảng cáo tại cửa hàng. Chính vì vậy hệ thống cửa hàng cần phải được cải thiện để có thể tiếp cận được nhiều khách hàng tiềm năng nhất.

### 2.1.4 Yêu cầu hệ thống mới

Hệ thống mới được thiết kế và xây dựng lên nhằm thu hút được nhiều khách hàng hơn, và có thể tăng số lượng đơn hàng bán ra và có thể quảng bá được hình ảnh, sản phẩm, chất lượng uy tín của cửa hàng tới tất cả mọi người. Có thể trao đổi dễ dàng với khách hàng bằng hình thức trực tuyến. Hệ thống sẽ cung cấp các thông tin chi tiết của các sản phẩm trong cửa hàng để khách hàng thuận tiện theo dõi và đánh giá được sản phẩm của cửa hàng. Mọi việc mua sản phẩm đều rất thuận tiện không cần phải đến tận cửa hàng để xem sản phẩm mà khách hàng vẫn có thể yên tâm mà mua hàng qua website.

* **Hệ thống quản trị trang web**
* Quản lý các đơn hàng các giao dịch.
* Quản lý danh mục sản phẩm.
* Quản lý sản phẩm: Cập nhật thông tin về các sản phẩm.
* Quản lý tin tức,
* Thống kê.
* **Hệ thống giới thiệu sản phẩm**
* Hiển thị danh mục của sản phẩm.
* Hiển thị thông tin sản phẩm (nhà cung cấp, giá, giảm giá, thông tin sản phẩm, hình ảnh sản phẩm, sản phẩm liên quan...)
* Giới thiệu về công ty.
* Hiển thị tin tức, bài viết.
* Cho phép đăng ký, đăng nhập, hiển hị thông tin người dùng, quản lý thay đổi mật khẩu, thay đổi thông tin người dùng.
* Đăng ký mua hàng, đặt hàng.
* Xem danh sách sản phẩm sau khi mua hàng.

**Các tác nhân của hệ thống**

Dựa vào mô tả bài toán, ta có thể xác định được các tác nhân chính của hệ thống như sau:

Tác nhân Chủ cửa hàng: người thực hiện các chức năng quản trị hệ thống, tài khoản, khách hàng, quản lý giao diện. Để thực hiện các chức năng quản trị trong hệ thống tác nhân này bắt buộc phải thực hiện đăng nhập.

Tác nhân khách hàng: có thể thực hiện các chức năng như đăng ký tài khoản, đăng nhập, quản lý thông tin tài khoản, thay đổi thông tin của tài khoản, thay đổi mật khẩu. Tìm kiếm sản phẩm, xem tin tức, đặt hàng, xem giỏ hàng, xem hóa đơn.

* **Chức năng quản lý của chủ cửa hàng**

| **Mục** | **Tên chức năng** | **Mô tả** |
| --- | --- | --- |
| 1 | Đăng nhập | Thực hiện đăng nhập vào hệ thống, nhập vào email và password. Kiểm tra hợp lệ trùng với dữ liệu có sẵn thì hệ thống sẽ cho phép đăng nhập để thực hiện các chức năng trong hệ thống. |
| 2 | Quản trị hệ thống | Quản lý toàn hộ hệ thống, có quyền thay đổi các thông tin sản phẩm, bài viết, trong hệ thống. |
| 3 | Quản lý sản phẩm | Quản lý, lưu trữ thông tin về sản phẩm như mã sản phẩm, tên sản phẩm… |
| 4 | Quản lý danh mục sản phẩm | Quản lý thông tin về các danh mục sản phẩm. |
| 5 | Quản lý đặt hàng | Quản lý thông tin về các đơn đặt hàng như: tên người đặt hàng, địa chỉ người nhận, số điện thoại, tổng tiền… |
| 6 | Quản lý chi tiết đặt hàng | Quản lý thông tin về đặt hàng gồm: mã chi tiết đặt hàng, mã đặt hàng, mã sản phẩm, số lượng, thành tiền. |
| 7 | Quản lý khách hàng | Quản lý hoạt động mua bán của khách hàng, lưu lại thông tin về khách hàng như tên khách hàng, địa chỉ, mã khách hàng, số điện thoại, email... |
| 8 | Quản lý chi tiết hóa đơn | Quản lý chi tiết hóa đơn của khách hàng có những thông tin như: mã giao dịch, mã hóa đơn, mã sản phẩm, số lượng, thành tiền. |
| 9 | Giỏ hàng | Giỏ hàng là để lưu thông tin mà khách hàng đã chọn có những trường như: mã giỏ hàng, mã sản phẩm, mã giao dịch, ngày tạo. |
| 10 | Quản lý thống kê, | Quản lý thống kê sản phẩm, doanh thu. |

Bảng 2. Bảng chức năng quản lý của chủ cửa hàng

* **Chức năng của người dùng.**

| **Mục** | **Tên chức năng** | **Mô tả** |
| --- | --- | --- |
| 1 | Tạo một tài khoản mới | Khách hàng có thể thực hiện thao tác đăng ký để trở thành người dùng thuận tiện cho việc mua sản phẩm và hưởng những khuyến mại của công ty. |
| 2 | Quản lý thông tin tài khoản | Khách hàng có thể thay đổi thông tin của tài khoản ,thay đổi mật khẩu , địa chỉ , thông tin liên lạc. |
| 3 | Đăng nhập | Thực hiện đăng nhập vào hệ thông, bắt buộc phải nhập email và password. Kiểm tra hợp lệ trùng với dữ liệu có sẵn thì hệ thống sẽ cho phép đăng nhập. |
| 4 | Tìm kiếm thông tin | Tìm kiếm, nhanh về thông tin sản phẩm, dựa vào các tiêu chí tìm kiếm của website như tìm theo tên của sản phẩm… |
| 5 | Đặt hàng | Khách hàng có thể thực hiện chức năng này để đặt mua hàng, nhằm giảm thiểu thời gian và chi phí cho khách hàng. |
| 6 | Giỏ hàng | Sau khi đăng nhập và chọn mua sản phẩm, giỏ hàng là nơi chứa thông tin về sản phẩm mà khách hàng đã chọn mua như số lượng sản phẩm, tên sản phẩm và giá tiền sản phẩm đó. |

Bảng 2. Bảng chức năng của người dùng

1. Yêu cầu chức năng

* Yêu cầu các chức năng phải đảm bảo được tính thiết thực của người sử dụng.
* Dễ dàng thực hiện, thao tác trên website, giao diện bắt mắt, không cầu kì và trải nghiệm mượt mà khi mua sản phẩm trên website

1. Yêu cầu phi chức năng

|  |  |  |
| --- | --- | --- |
| **Mục** | **Tên yêu cầu** | **Mô tả yêu cầu** |
| 1 | Giao diện | Giao diện hệ thống phải dễ sử dụng, trực quan, thân thiện với người dùng. |
| 2 | Tốc độ xử lý | Hệ thống phải xử lý nhanh chóng và chính xác. |
| 3 | Bảo mật | Tính bảo mật và độ an toàn cao. |
| 4 | Tương thích | Tương thích với đa phần các trình duyệt web hiện tại. |

## 2.2 Biểu đồ usecase

### 2.2.1 Biểu đồ usecase chính của hệ thống



Hình 2. Biểu đồ usecase tổng quát của hệ thống

### 2.2.2 Mô tả các usecase trong hệ thống

#### 1. Mô tả usecase DangNhap (Đăng nhập)

* **Mô tả vắn tắt :** Use case cho phép người dùng truy cập vào hệ thống.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    1. use case bắt đầu khi người dùng nhấn vào “ Đăng nhập “ trên menu công cụ , hệ thống sẽ hiện thị màn hình đăng nhập.
    2. Hệ thống yêu cầu người dùng nhập email và mật khẩu và kích vào “ Đăng nhập “.
    3. Hệ thống sẽ kiểm tra email và mật khẩu trong CSDL và hiển thị màn hình trang chủ theo đúng phân quyền của tài khoản nhập vào và use case kết thúc.
  + **Luồng rẽ nhánh** :
    1. Tại bước 2 , nếu người dùng nhập sai tài khoản hoặc mật khẩu thì hệ thống sẽ hiện thị thông báo lỗi “Tên đăng nhập hoặc mật khẩu không đúng”. Người dùng có thể nhập lại và nhấn vào đăng nhập và use case sẽ kết thúc.
    2. Tại bất cứ bước nào nếu bị mất kết nối với CSDL thì hệ thống sẽ thông báo “ Lỗi kết nối “ và use case kết thúc.
* **Yêu cầu đặc biệt:** không
* **Tiền điều kiện:** Không có
* **Hậu điều kiện:**

Sau khi thực hiện usecase thành công, người dùng phải truy cập được vào hệ thống và có thể thực hiện các chức năng khác.

* **Điểm mở rộng**: Đăng ký

#### 2. Mô tả usecase DangKy (Đăng ký)

* **Mô tả vắn tắt :** Use case cho phép người dùng tạo tài khoản truy cập cho riêng mình.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    1. usecase bắt đầu khi người dùng nhấn vào “ Đăng ký “ trên menu công cụ , hệ thống sẽ hiện thị màn hình đăng ký.
    2. Hệ thống yêu cầu người dùng nhập các thông tin trên màn hình và kích vào “ Đăng ký “.
    3. Hệ thống sẽ thực hiện thêm tài khoản mới vào CSDL, hệ thống hiển thị màn hình đăng nhập và use case kết thúc.
  + **Luồng rẽ nhánh** :
    1. Tại bước 2 , nếu người dùng nhập sai sai một số thong tin thì hệ thống sẽ hiện thị thông báo lỗi. Người dùng có thể nhập lại và nhấn vào đăng ký và use case sẽ kết thúc.
    2. Tại bất cứ bước nào nếu bị mất kết nối với CSDL thì hệ thống sẽ thông báo “ Lỗi kết nối “ và use case kết thúc.
* **Yêu cầu đặc biệt:** không
* **Tiền điều kiện:** Không
* **Hậu điều kiện:**

Sau khi thực hiện use thành công, người dùng có thể sử dụng tài khoản vừa tạo để truy cập được vào hệ thống và có thể thực hiện các chức năng khác.

* **Điểm mở rộng**: không

#### 3. Mô tả usecase TimKiemSP (Tìm kiếm sản phẩm)

* **Mô tả vắn tắt:** Use case cho phép khách hàng tìm kiếm sản phẩm.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    1. Usecase bắt đầu khi người dùng nhập tên sản phẩm trên thanh tìm kiếm và kích vào nút “tìm kiếm”.
    2. Hệ thống thực hiện tìm kiếm các sản phẩm có tên của người dùng nhập vào.
    3. Hệ thống đưa ra màn hình tên các sản phẩm tìm được.
    4. Usecase kết thúc.
  + **Luồng rẽ nhánh** :
    1. Nếu không có sản phẩm có tên chứa từ cần tìm kiếm thì hệ thống không hiển thị và usecase kết thúc.
    2. Tại bất cứ bước nào nếu bị mất kết nối với CSDL thì hệ thống sẽ thông báo “Lỗi kết nối “ và use case kết thúc.
* **Yêu cầu đặc biệt:** Không
* **Tiền điều kiện:** Không có
* **Hậu điều kiện:** Không có
* **Điểm mở rộng**: Không có

#### 4. Mô tả usecase ThemVaoGioHang (Thêm vào giỏ hàng)

* **Mô tả vắn tắt :** Use case cho phép người dùng truy cập thực hiện thêm vào giỏ hàng.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    1. Sau khi chọn được sản phẩm muốn mua
    2. Người truy cập thực hiện kích vào nút “Thêm vào giỏ ” trên sản phẩm đó
    3. Hệ thống thực hiện thêm sản phẩm đó vào giỏ và hiển thị thông tin của sản phẩm đó và số lượng sản phẩm đó trong “giỏ hàng”.
    4. Usecase kết thúc.
  + **Luồng rẽ nhánh** :
    1. Tại bất cứ bước nào nếu bị mất kết nối với CSDL thì hệ thống sẽ thông báo “ Lỗi kết nối “ và use case kết thúc.
* **Yêu cầu đặc biệt:** không
* **Tiền điều kiện:** Đăng nhập để thực hiện usecase này
* **Hậu điều kiện:** Sau khi thực hiện usecase này , sản phẩm cần thêm vào giỏ hàng.
* **Điểm mở rộng**: không có.
* **Dữ liệu liên quan:**

#### 5. Mô tả usecase XemGioHang (Xem giỏ hàng)

* **Mô tả vắn tắt :** Use case cho phép người dùng có thể thực hiện xem các sản phẩm mà mình đã thực hiện thêm vào giỏ hàng.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    1. use case bắt đầu khi người dùng nhấn vào “ Giỏ hàng “ trên menu công cụ.
    2. Hệ thống sẽ hiển thị các sản phẩm mà người truy cập đã thực hiện thêm vào giỏ hàng gồm các thông tin : Tên sản phẩm, hình ảnh, đơn giá, số lượng và tổng tiền của đơn hàng, số lượng đơn hàng.
    3. Usecase kết thúc
  + **Luồng rẽ nhánh** :
    1. Tại bất cứ bước nào nếu bị mất kết nối với CSDL thì hệ thống sẽ thông báo “ Lỗi kết nối “ và use case kết thúc.
* **Yêu cầu đặc biệt:** không
* **Tiền điều kiện:** Đăng nhập
* **Hậu điều kiện:** không có
* **Điểm mở rộng**: Đặt hàng

#### 6. Mô tả usecase DatHang (Đặt hàng)

* **Mô tả vắn tắt :** Use case cho phép người dùng truy cập thực hiện đặt hàng.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    1. Sau khi thực hiện xem các hàng đã thêm vào giỏ hàng
    2. Người truy cập thực hiện kích vào “Đặt hàng”.
    3. Hệ thống hiển thị giao diện form để khách hàng điền thông tin nhận hàng (nếu chưa đăng nhập). Nếu đăng nhập rồi thì khách hàng chỉ cần kiểm tra lại thông tin nhận hàng và ghi chú cho cửa hàng.
    4. Hệ thống sẽ xóa đơn hàng đó khỏi giỏ hàng và thông báo “đặt hàng thành công” và usecase kết thúc
  + **Luồng rẽ nhánh** :
    1. Tại bất cứ bước nào nếu bị mất kết nối với CSDL thì hệ thống sẽ thông báo “ Lỗi kết nối “ và use case kết thúc.
* **Yêu cầu đặc biệt:** không
* **Tiền điều kiện:** Không
* **Hậu điều kiện:**

Đơn hàng phải bị xóa khỏi giỏ hàng và thêm vào mục danh sách đơn hàng

* **Điểm mở rộng**: không có

#### 7. Mô tả usecase XemDonHangDaDat (Xem đơn hàng đã đặt)

* **Mô tả vắn tắt :** Use case cho phép người dùng truy cập thực hiện xem thông tin tiết đơn hàng đã đặt.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    1. Sau khi thực hiện xem tình trạng đơn hàng , người truy cập có thể thực hiện xem thông tin tiết của đơn hàng bằng cách nhấn vào nút “Danh sách đơn hàng” trên đơn hàng muốn xem.
    2. Hệ thống sẽ hiển thị màn hình các thông tin gồm : Mã đơn, tên sản phẩm, tổng tiền, ngày đặt.
    3. Usecase kết thúc.
  + **Luồng rẽ nhánh** :
    1. Tại bất cứ bước nào nếu bị mất kết nối với CSDL thì hệ thống sẽ thông báo “ Lỗi kết nối “ và use case kết thúc.
* **Yêu cầu đặc biệt:** không
* **Tiền điều kiện:** Đăng nhập để thực hiện usecase này
* **Hậu điều kiện:** không có
* **Điểm mở rộng**: không có

#### 8. Mô tả usecase XoaSanPhamTrongGioHang (Xóa sản phẩm trong giỏ hàng)

* **Mô tả vắn tắt :** Use case cho phép người dùng truy cập thực hiện xóa đi các sản phẩm đã được thêm vào giỏ hàng trước đó.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    1. Sau khi chọn được sản phẩm muốn xóa trong giỏ hàng
    2. Người truy cập thực hiện kích chọn “Xóa” trên sản phẩm muốn xóa.
    3. Hệ thống thực hiện xóa sản phẩm đó ra khỏi giỏ hàng và hiển thị thông báo “Xóa sản phẩm khỏi giỏ hàng thành công” . Usecase kết thúc.
  + **Luồng rẽ nhánh** :
    1. Tại bất cứ bước nào nếu bị mất kết nối với CSDL thì hệ thống sẽ thông báo “ Lỗi kết nối “ và use case kết thúc.
* **Yêu cầu đặc biệt:** không
* **Tiền điều kiện:** không
* **Hậu điều kiện:** Sau khi thực hiện usecase này thành công thì sản phẩm bị xóa phải bị xóa khỏi giỏ hàng của người truy cập.
* **Điểm mở rộng**: không có
* **Dữ liệu liên quan:**

#### 9. Mô tả usecase XemSanPhamTrongDanhMuc (Xem sản phẩm trong danh mục)

* **Mô tả vắn tắt :** Use case cho phép người dùng truy cập thực hiện xem các sản phẩm của danh mục chỉ định nào đó.
* **Luồng sự kiện:**
  + **Luồng cơ bản:**
    1. Người truy cập chọn danh mục cần xem
    2. Usecase bắt đầu khi người truy cập thực hiện kích vào danh mục nào đó trên hệ thống.
    3. Hệ thống thực hiện lấy danh sách các sản phẩm thuộc danh mục đó hiển thị lên màn hình.
    4. Usecase kết thúc.
  + **Luồng rẽ nhánh** :
    1. Tại bất cứ bước nào nếu bị mất kết nối với CSDL thì hệ thống sẽ thông báo “ Lỗi kết nối “ và use case kết thúc.
* **Yêu cầu đặc biệt:** không
* **Tiền điều kiện:** không có
* **Hậu điều kiện:** không có
* **Điểm mở rộng**: ThemVaoGioHang, XemChiTietSanPham

#### 10. Mô tả usecase QuanLySanPham (Quản lý sản phẩm)

* **Mô tả vắn tắt :**  usecase cho phép người quản lí được các sản phẩm có trong hệ thống.
* **Luồng sự kiện :**
* Luồng cơ bản :

1. Usecase bắt đầu khi Chủ cửa hàng chọn vào “QUẢN LÝ SẢN PHẨM” trên menu chính. Hệ thống sẽ lấy dữ liệu các thuộc tính của sản phẩm từ CSDL hiển thị lên màn hình
2. Chủ cửa hàng có thể thực hiện :

Thêm sản phẩm mới :

Chủ cửa hàng kích vào “Thêm mới” trên màn hình. Hệ thống hiện thị màn hình yêu cầu người dùng nhập thông tin sản phẩm mới.

Chủ cửa hàng kích vào nút “Lưu” trên màn hình và hệ thống sẽ hiện thị thông báo “Sản phẩm đã được thêm thành công”.

Sửa thông tin Sản phẩm

Chủ cửa hàng kích vào nút “Xem/Sửa” ở một hàng sản phẩm trong danh sách sản phẩm. Hệ thống sẽ hiển thị màn hình các thông tin cũ của sản phẩm và yêu cầu sửa các thông tin cần sửa.

Chủ cửa hàng thực hiện nhập lại các thông tin sai và kích vào nút “Lưu” trên màn hình.

Hệ thống thực hiện cập nhật thông tin và hiển thị thông báo “Cập nhật sản phẩm thành công”.

Khóa sản phẩm

Chủ cửa hàng chọn sản phẩm cần khóa

Chủ cửa hàng kích vào nút “Khóa” trên hàng sản phẩm đó trong danh sách sản phẩm.

Hệ thống thực hiện khóa sản phẩm lại. và hiển thị thông báo “Khóa sản phẩm thành công”.

Mở sản phẩm

Chủ cửa hàng chọn sản phẩm bị khóa cần khóa

Chủ cửa hàng kích vào nút “Mở” trên hàng sản phẩm đó trong danh sách sản phẩm.

Hệ thống thực hiện khóa sản phẩm lại. và hiển thị thông báo “Kích hoạt sản phẩm thành công”.

* Luồng rẽ nhánh :
  1. Tại bước 2\_3 nếu có lỗi gì đó sảy ra , hệ thống sẽ thông báo lỗi “ Khóa sản phẩm thất bại ” và usecase kết thúc.
  2. Tại bước 2\_4 nếu có lỗi gì đó xảy ra , hệ thống sẽ thông báo lỗi “ Kích hoạt sản phẩm thất bại ” và usecase kết thúc.
  3. Tại bất cứ bước nào , nếu có lỗi gì đó xảy ra , hệ thống sẽ thông báo lỗi “Có lỗi xảy ra!” và usecase kết thúc.
* **Yêu cầu đặc biệt:** không có
* **Tiền điều kiện:** người truy cậpcần phải đăng nhập với phân quyền là admin thì mới có thể sử dụng được usecase này
* **Hậu điều kiện:** Sau khi thực hiện usecase này , các bảng liên quan trong hệ thống cơ sở dữ liệu cần được cập nhật lại.
* **Điểm mở rộng**: không có.

#### 11. Mô tả usecase QuanLyDanhMuc (Quản lý danh mục)

* **Mô tả vắn tắt :**  usecase cho phép người quản lí được các danh mục có trong hệ thống.
* **Luồng sự kiện :**
* Luồng cơ bản :

1. Usecase bắt đầu khi Chủ cửa hàng chọn vào “QUẢN LÝ DANH MỤC” trên menu chính.Hệ thống sẽ lấy dữ liệu từ CSDL hiển thị lên màn hình những loại sản phẩm có trong danh mục.
2. Chủ cửa hàng có thể thực hiện :

Thêm danh mục mới :

Chủ cửa hàng kích vào “Thêm mới” trên màn hình. Hệ thống hiện thị màn hình yêu cầu Chủ cửa hàng nhập tên danh mục mới.

Chủ cửa hàng nhập và kích vào nút “Lưu” trên màn hình và hệ thống sẽ hiện thị thông báo “Thêm danh mục thành công”.

Sửa thông tin danh mục

Người dùng kích vào nút “Xem/Sửa” ở một hàng danh mục trong danh sách sản phẩm. Hệ thống sẽ hiển thị màn hình các thông tin cũ của danh mục và yêu cầu sửa tên danh mục cần sửa.

Người dùng thực hiện nhập lại các thông tin sai và kích vào nút “Lưu” trên màn hình.

Hệ thống thực hiện cập nhật thông tin và hiển thị thông báo “Cập nhật danh mục thành công”.

Khóa sản phẩm

Chủ cửa hàng chọn danh mục cần khóa

Chủ cửa hàng kích vào nút “Khóa” trên hàng danh mục đó trong danh sách danh mục.

Hệ thống thực hiện khóa danh mục lại và hiển thị thông báo “Khóa danh mục thành công!”.

Mở sản phẩm

Chủ cửa hàng chọn danh mục bị khóa cần khóa

Chủ cửa hàng kích vào nút “Mở” trên hàng danh mục đó trong danh sách danh mục.

Hệ thống thực hiện khóa danh mục lại. và hiển thị thông báo “Kích hoạt danh mục thành công”.

* Luồng sự kiện :
  1. Tại bước 2\_3 nếu có lỗi gì đó sảy ra , hệ thống sẽ thông báo lỗi “ Khóa danh mục thất bại ” và usecase kết thúc.
  2. Tại bước 2\_4 nếu có lỗi gì đó xảy ra , hệ thống sẽ thông báo lỗi “ Kích hoạt danh mục thất bại ” và usecase kết thúc.
  3. Tại bất cứ bước nào , nếu có lỗi gì đó xảy ra , hệ thống sẽ thông báo lỗi “Có lỗi xảy ra!” và usecase kết thúc.
* **Yêu cầu đặc biệt:** không có
* **Tiền điều kiện:** cần phải đăng nhập với phân quyền là Admin thì mới có thể sử dụng được usecase này
* **Hậu điều kiện:** Sau khi thực hiện usecase này , các bảng liên quan trong hệ thống cơ sở dữ liệu cần được cập nhật lại.
* **Điểm mở rộng**: không có.

#### 12. Mô tả usecase QuanLyDonHang (Quản lý đơn hàng)

* **Mô tả vắn tắt :**  usecase cho phép người quản lí được các đơn hàng có trong hệ thống.
* **Luồng sự kiện :**
* Luồng cơ bản :
  + 1. Usecase bắt đầu khi Chủ cửa hàng chọn vào “QUẢN LÝ ĐƠN HÀNG” trên menu chính.
    2. Chủ cửa hàng có thể thực hiện :

Xem tình trạng các đơn hàng có trong hệ thống

Xem tình trạng các đơn hàng chờ xác nhận có trong hệ thống

Xem tình trạng các đơn hàng đang xử lý

Chủ cửa hàng kích vào checkbox “Đang xử lý” trên màn hình. Hệ thống hiện thị màn hình các thông tin tên khách hàng, số điện thoại, địa chỉ nhận hàng, tên sản phẩm và số lượng để và tích vào check box để xác nhận cho khách hàng.

Xem tình trạng các đơn hàng đang vận chuyển

Chủ cửa hàng kích vào “Đang vận chuyển” trên màn hình. Hệ thống hiện thị danh sách các phương thức vận chuyển để người quản lý chọn và nhập ngày nhận dự kiến trên màn hình để lên đơn vận chuyển cho đơn hàng

Xem tình trạng các đơn hàng đã giao

Chủ cửa hàng kích vào “Đã giao” trên màn hình. Hệ thống hiện thị màn hình checkbox “xác nhận đã giao hàng”. Nếu sản phẩm đã được giao tới khách hàng thì người quản lý tích vào ô checkbox

Xem tình trạng đơn hàng đã hủy

Cập nhật đơn hàng.

Người quản lý có thể cập nhật thông tin của đơn hàng khi đơn đó chưa giao tới khách hàng.

* Luồng sự kiện :
  1. Tại bước 2 nếu không có đơn hàng nào, hệ thống sẽ thông báo “ Chưa có đơn hàng nào chờ xử lý” và usecase kết thúc.
  2. Tại bước 3 nếu không có đơn hàng nào, hệ thống sẽ thông báo “ Chưa có đơn hàng nào đang giao” và usecase kết thúc.
  3. Tại bước 4 nếu không có đơn hàng nào, hệ thống sẽ thông báo “ Chưa có đơn hàng nào hoàn thành” và usecase kết thúc.
* **Yêu cầu đặc biệt:** không có
* **Tiền điều kiện:** cần phải đăng nhập với phân quyền là Chủ cửa hàng thì mới có thể sử dụng được usecase này
* **Hậu điều kiện :** không có
* **Điểm mở rộng**: không có.

## 2.3 Biểu đồ trình tự của usecase

#### 1. Usecase Đăng nhập



Hình 2. Biểu đồ trình tự usecase Đăng nhập

#### 2. Usecase Đăng ký



Hình 2. Biểu đồ trình tự usecase Đăng ký

#### 3. Usecase Tìm kiếm sản phẩm



Hình 2. Biểu đồ trình tự usecase Tìm kiếm sản phẩm

#### 4 Usecase Thêm vào giỏ hàng



Hình 2. Biểu đồ trình tự usecase Thêm vào giỏ hàng

#### 5. Usecase Xem giỏ hàng



Hình 2. Biểu đồ trình tự usecase Xem giỏ hàng

#### 6. Usecas**e Đặt hàng**



Hình 2 Biểu đồ trình tự usecase Đặt hàng

#### 7. Usecase Xem chi tiết đơn hàng đã đặt



Hình 2. Biểu đồ trình tự usecase Xem chi tiết đơn hàng đã đặt

#### 8. Usecase Xóa sản phẩm trong giỏ hàng



Hình 2. Biểu đồ trình tự usecase Xóa sản phẩm trong giỏ hàng

#### 9. Usecase Xem sản phẩm trong danh mục



Hình 2. Biểu đồ trình tự usecase Xem danh sách sản phẩm trong danh mục

#### 10. Usecase Quản lý sản phẩm



Hình 2. Biểu đồ trình tự usecase Quản lý sản phẩm

#### 11. Usecase Quản lý danh mục



Hình 2. Biểu đồ trình tự usecase Quản lý danh mục

#### 12. Usecase Quản lý đơn hàng



Hình 2. Biểu đồ trình tự usecase Quản lý đơn hàng

## 2.5 Thiết kế cơ sở dữ liệu

### 2.5.1 Danh sách các bảng cơ sở dữ liệu

| **STT** | **Tên thực thể** | **Mô tả** |
| --- | --- | --- |
| 1 | Categories | Lưu thông tin danh mục sản phẩm |
| 2 | Orders | Lưu thông tin danh sách đơn hàng |
| 3 | Products | Lưu thông tin danh sách sản phẩm |
| 4 | Cart\_Item | Lưu thông tin của các sản phẩm trong giỏ hàng |
| 5 | Accounts | Lưu thông tin danh sách thành viên |
| 6 | Account\_detail | Lưu thông tin chi tiết của từng tài khoản |
| 7 | Attribute | Lưu thuộc tính đặc trưng của từng sản phẩm |
| 8 | Brands | Lưu thông tin hãng của sản phẩm |
| 9 | Images | Lưu thông tin ảnh sản phẩm |
| 10 | Notifications | Lưu thông của các thông báo trong hệ thống |
| 11 | Order\_detail | Lưu thông tin chi tiết của đơn đặt hàng |
| 12 | Order\_status | Lưu thông tin trạng thái của đơn hàng |
| 13 | Roles | Lưu thông tin vai trò của các tài khoản |
| 14 | Sales | Lưu thông của của các chương trình sale |
| 15 | Vouchers | Lưu thông tin của các Voucher |

Bảng 2. Danh sách các bảng cơ sở dữ liệu

### 2.5.2 Bảng Category

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | Id | Int(11) | Khóa chính | Khóa chính của bảng |
| 2 | Name | Varchar(128) | Not null | Tên danh mục |
| 3 | description | Varchar(255) | Not null | Mô tả |
| 4 | Is\_active | bit | Null | Trạng thái |
| 5 | modify\_date | date | Not null | Ngày cập nhật |
| 6 | Create\_date | Timestamp | Not null | Ngày tạo |

Bảng 2. Bảng Category

### 2.5.3 Bảng Sales

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | Id | Int (11) | Khóa chính | Khóa chính của bảng |
| 2 | Create\_date | Date | Null | Ngày tạo |
| 3 | Description | Varchar(32) | Not null | Mô tả |
| 4 | Discount | Int | Null | Giảm giá bao nhiêu % |
| 5 | Is\_active | Bit | Not null | Trạng thái hoạt động |
| 6 | Modify\_date | Date | Null | Ngày cập nhật |
| 6 | Name | Varchar(50) | Not null | Tên của chương trình sales |

Bảng 2. Bảng News

### 2.5.4 Bảng Orders

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | Id | Int(11) | Not null | Khóa chính |
| 2 | Address | varchar (255) | Not null | Địa chỉ nhận hàng |
| 3 | Create\_date | date | Not null | Ngày tạo |
| 4 | Fullname | varchar (50) | Not null | Tên khách hàng |
| 5 | Is\_pending | tinyint | Not null | Tình trạng xử lý |
| 6 | Modify\_date | Date | Not null | Ngày cập nhật |
| 7 | Note | Varchar(100) | Not null | Ghi chú |
| 8 | Phone | Varchar(11) | Not null | Số điện thoại khách hàng |
| 9 | Total | Double | Not null | Tổng số lượng |
| 10 | Payment | Varchar(1000) | Not null | Loại thanh toán |
| 11 | Email | Varchar(255) | Not null | Email khách hàng |

Bảng 2. Bảng Orders

### 2.5.5 Bảng Products

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | Int(11) | Khóa chính | Khóa chính |
| 2 | Code | Int(11) | Not null | Code của sản phẩm |
| 3 | Name | Varchar(100) | Not null | Tên sản phẩm |
| 4 | Price | Double | Not null | Giá sản phẩm |
| 5 | Description | Text | Not null | Mô tả sản phẩm |
| 6 | View | Int(11) | Not null | View sản phẩm |
| 7 | Is\_active | Tinyint | Not null | Trạng thái của sản phẩm |
| 8 | Create\_date | Date | Not null | Ngày tạo sản phẩm |
| 9 | Modify\_date | Date | Not null | Ngày cập nhật sản phẩm |

Bảng 2. Bảng Products

### 2.5.6 Bảng Attribute

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | Bigint(20) | Khóa chính | Id Attribute |
| 2 | Create\_date | Date | Not null | Ngày tạo |
| 3 | Modify\_date | Date | Not null | Ngày cập nhật |
| 4 | Price | Double | Not null | Giá của sản phẩm |
| 5 | Amount | Int | Not null | Số lượng sản phẩm |
| 6 | Name | Varchar(50) | Not null | Tên của sản phẩm |

Bảng 2. Bảng Attribute

### 2.5.7 Bảng Accounts

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | Int | Khóa chính | Id người dùng |
| 2 | Username | Varchar(50) | Not null | User đăng nhập người dùng |
| 3 | Password | Varchar(255) | Not null | Mật khẩu |
| 4 | Modify\_date | Date | Not null | Ngày cập nhật |
| 5 | Role\_id | Varchar(50) | Not null | Quyền của user |
| 6 | Is\_active | Tinyint | Not null | Trạng thái của tài khoản |
| 7 | Create\_date | Date | Not null | Ngày tạo |

Bảng 2. Bảng Accounts

### 2.5.8 Bảng Account\_Detail

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | BigInt | Khóa chính | Id người dùng |
| 2 | Birthday | DateTime | Not null | Ngày sinh người dùng |
| 3 | Email | Varchar(50) | Not null | Email người dùng |
| 4 | Fullname | Varchar(50) | Not null | Tên người dùng |
| 5 | Gender | Varchar(10) | Not null | Giới tính |
| 6 | Phone | Varchar(11) | Not null | Số điện thoại |
| 7 | Address | Varchar(255) | Not null | Địa chỉ |

Bảng 2. Bảng Account\_Detail

### 2.5.9 Bảng Order\_Detail

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | Id | Int(11) | Not null | Khóa chính |
| 2 | Origin\_price | Double | Not null | Giá gốc |
| 3 | Quantity | Int | Not null | Số lượng |
| 4 | Sell\_price | Double | Not null | Giá bán |

Bảng 2. Bảng Order\_Detail

### 2.5.10 Bảng Images

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | BigInt | Khóa chính | Id ảnh |
| 2 | Create\_date | Date | Not null | Ngày tạo |
| 3 | Modify\_date | Date | Not null | Ngày cập nhật |
| 4 | Image\_link | Bit | Not null | Đường link của ảnh |
| 5 | Name | Varchar(25) | Not null | Tên của ảnh |
| 6 | Is\_active | Bit | Not null | Trạng thái của ảnh |

Bảng 2. Bảng Images

### 2.5.11 Bảng Brands

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | BigInt | Khóa chính | Id Brands |
| 2 | Create\_date | Date | Not null | Ngày tạo |
| 3 | Description | Varchar(255) | Not null | Mô tả |
| 4 | Image | Varchar(255) | Not null | Ảnh |
| 5 | Is\_active | Bit | Not null | Trạng thái |
| 6 | Modify\_date | Date | Not null | Ngày cập nhật |
| 7 | Name | Varchar(50) | Not null | Tên Brand |

Bảng 2. Bảng Brands

### 2.5.12 Bảng Notifications

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | BigInt | Khóa chính | Id Notification |
| 2 | Content | Varchar(255) | Not null | Nội dung thông báo |
| 3 | Read | TinyInt | Not null | Trạng thái đã đọc hay chưa |
| 4 | Deliver | TinyInt | Not null | Vận chuyển |
| 5 | Order\_id | BigInt | Not null | Mã đơn hàng |
| 6 | Type | Int | Not null | Type thông báo |
| 7 | Product\_id | BigInt | Not null | Mã sản phẩm |

Bảng 2. Bảng Notifications

### 2.5.13 Bảng Cart\_Item

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | BigInt | Khóa chính | Id Cart\_Item |
| 2 | Quantity | Int | Not null | Số lượng |
| 3 | Account\_id | BigInt | Not null | Mã tài khoản |
| 4 | Attribute\_id | BigInt | Not null | Mã attribute |
| 5 | Last\_price | Double | Not null | Giá bán |
| 6 | Is\_active | TinyInt | Not null | Trạng thái |

Bảng 2. Bảng Cart\_Item

### 2.5.14 Bảng Order\_Status

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | BigInt | Khóa chính | ID Order\_status |
| 2 | Create\_date | Date | Not null | Ngày tạo |
| 3 | Description | Varchar(255) | Not null | Mô tả |
| 4 | Name | Varchar(50) | Not null | Tên đơn hàng |
| 5 | Update\_date | Date | Not null | Ngày cập nhật |

Bảng 2. Bảng Order\_Status

### 2.5.15 Bảng Roles

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | BigInt | Khóa chính | ID Roles |
| 2 | Create\_date | Date | Not null | Ngày tạo |
| 3 | Modify\_date | Date | Not null | Ngày cập nhật |
| 4 | Name | Varchar(50) | Not null | Tên role |

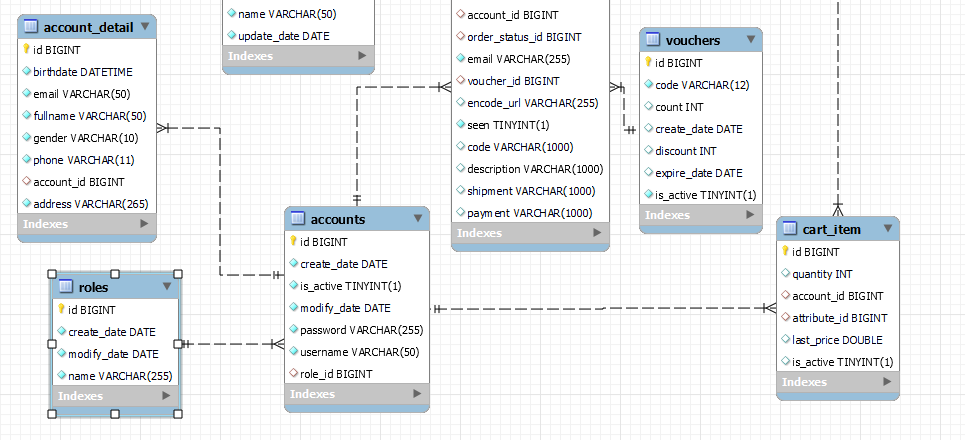
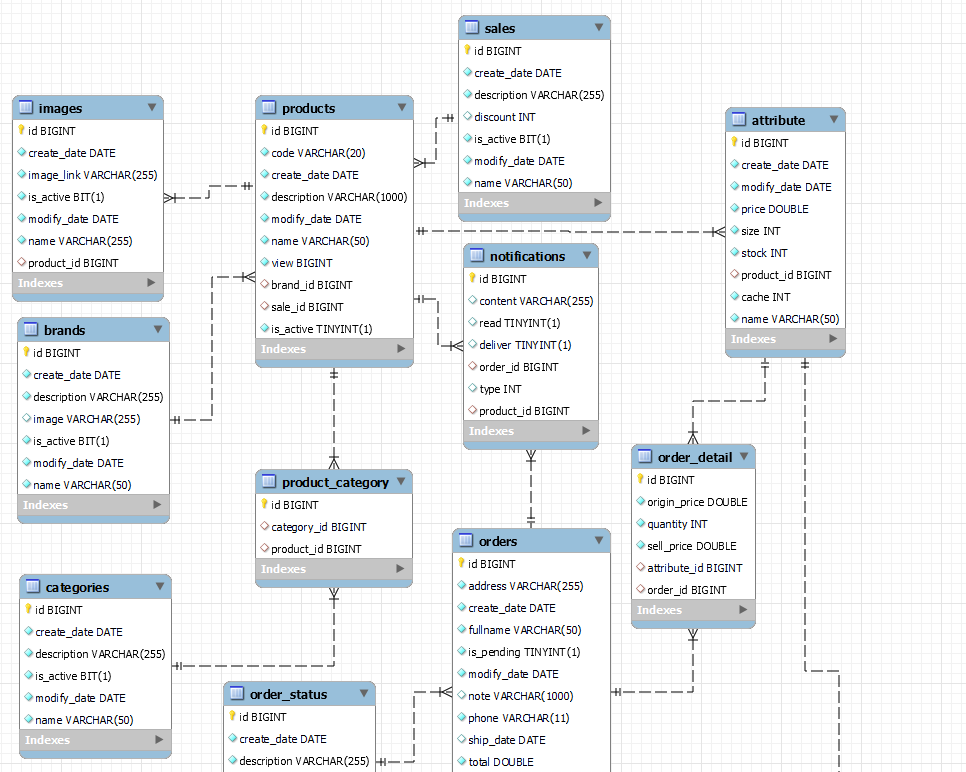
Bảng 2. Bảng Roles

### 2.5.16 Bảng Vouchers

| **STT** | **Tên thực thể** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- |
| 1 | ID | BigInt | Khóa chính | ID Voucher |
| 2 | Create\_date | Date | Not null | Ngày tạo |
| 3 | Code | Varchar(12) | Not null | Mã code |
| 4 | Count | Int | Not null | Số lượng |
| 5 | Discount | Int | Not null | Giảm giá bao nhiêu % |
| 6 | Expri\_date | Date | Not null | Ngày hết hạn |
| 7 | Is\_active | Tinyint | Not null | Trạng thái hoạt động |

Bảng 2. Bảng Vouchers

### 2.5.8 Mô hình cơ sở dữ liệu



## 2.6 Thiết kế giao diện

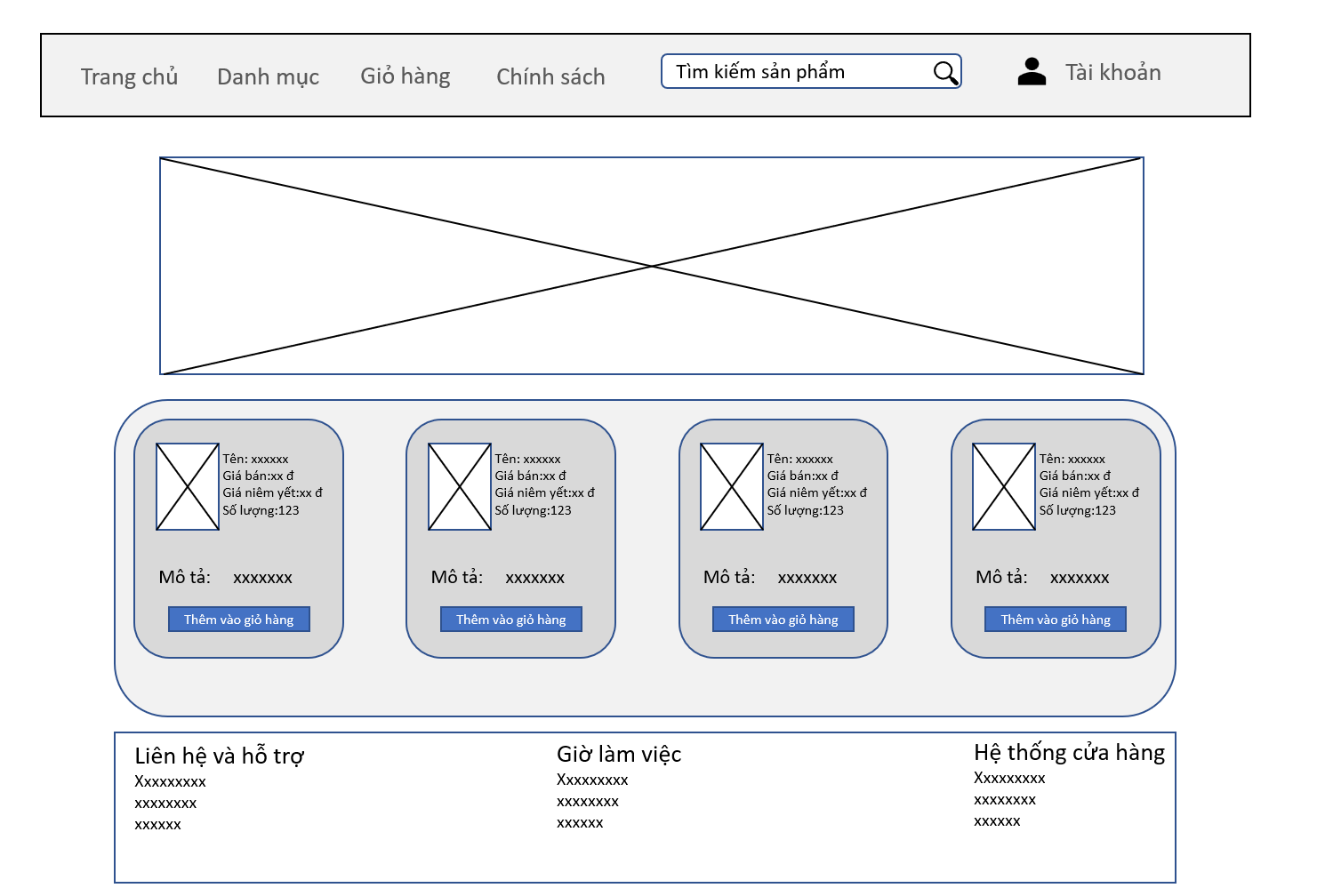
### 2.6.1 Giao diện người dùng



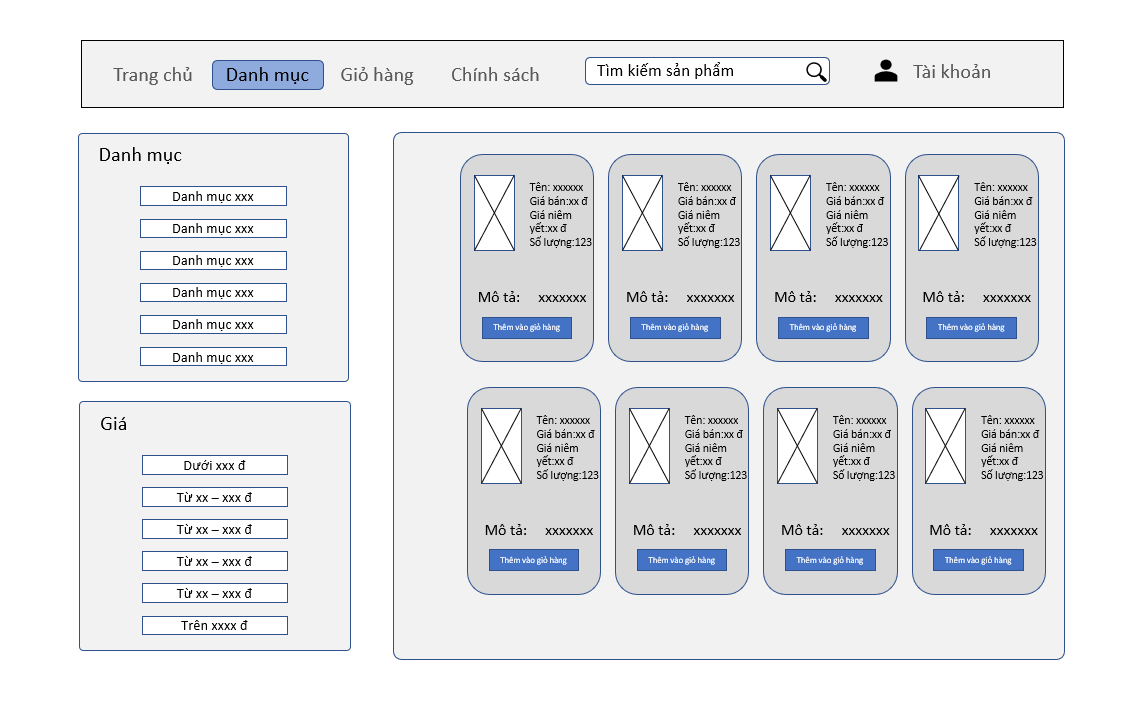
Hình 2. Màn hình đăng nhập



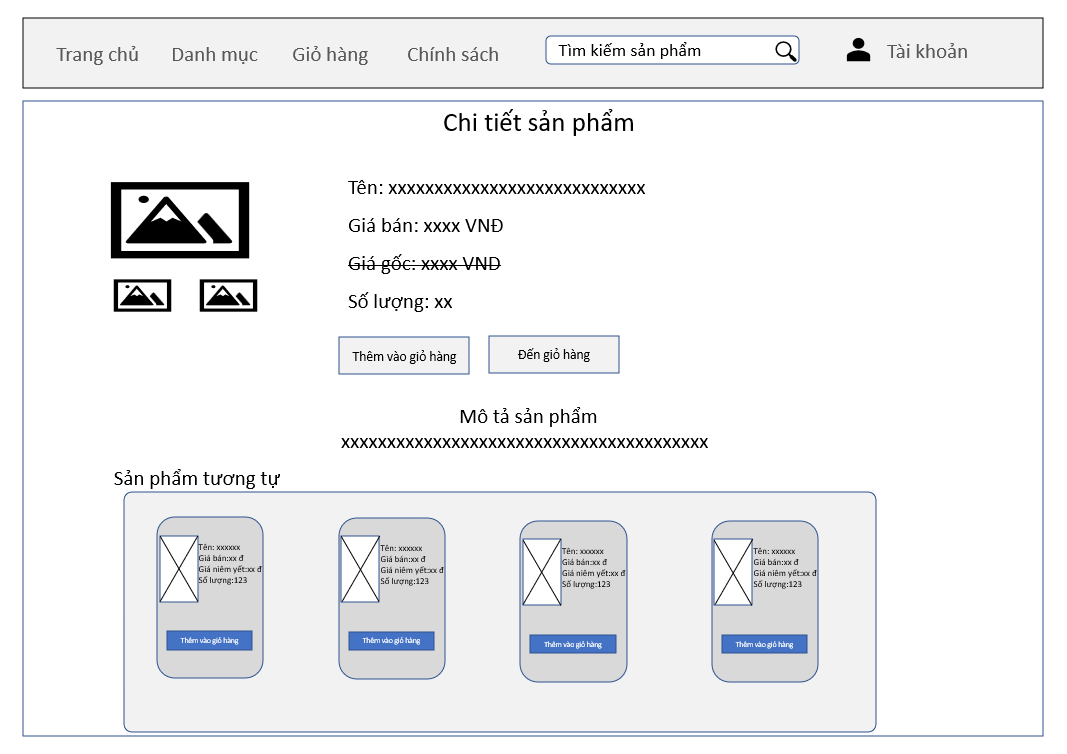
Hình 2. Màn hình đăng ký



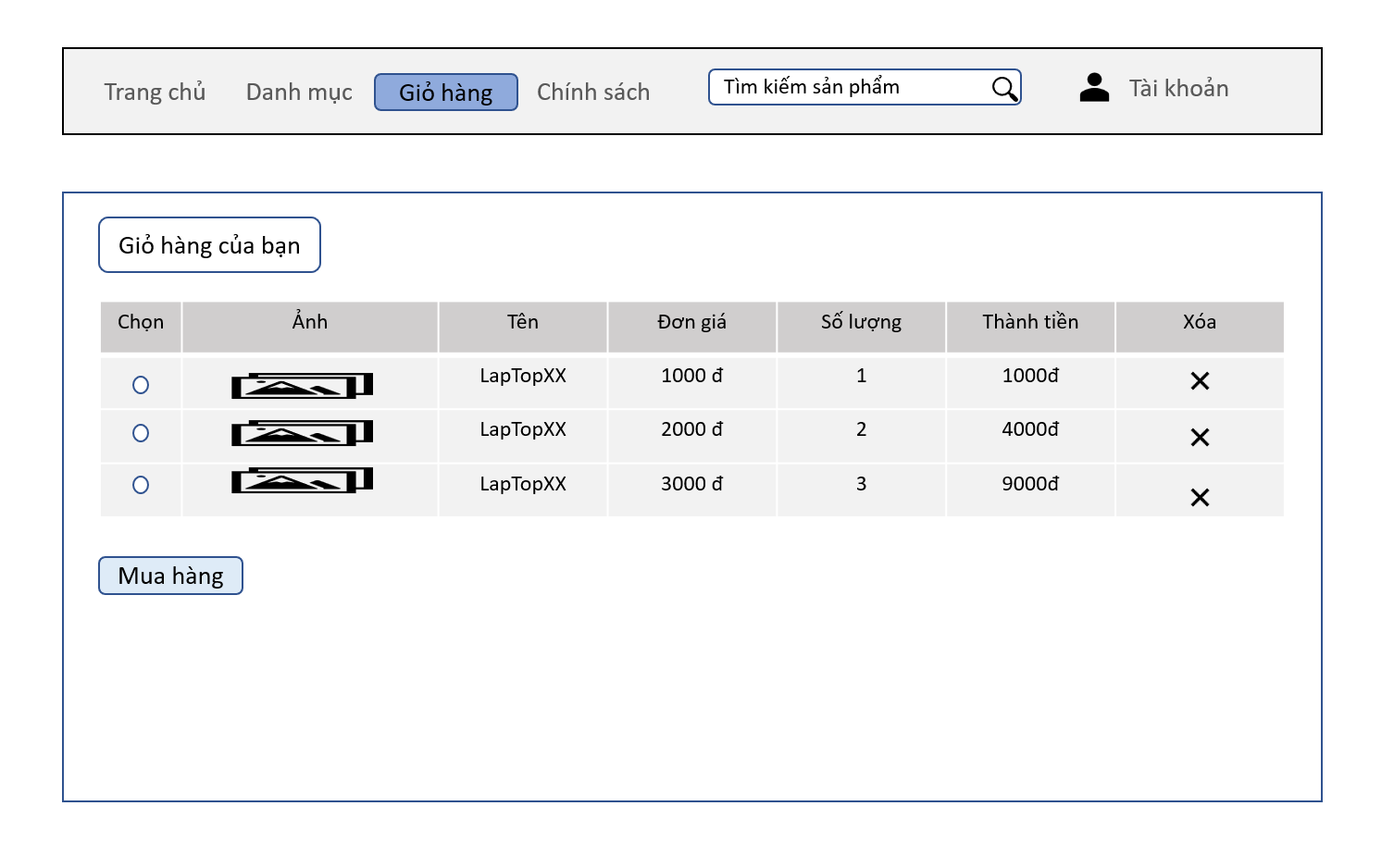
Hình 2. Màn hình trang chủ



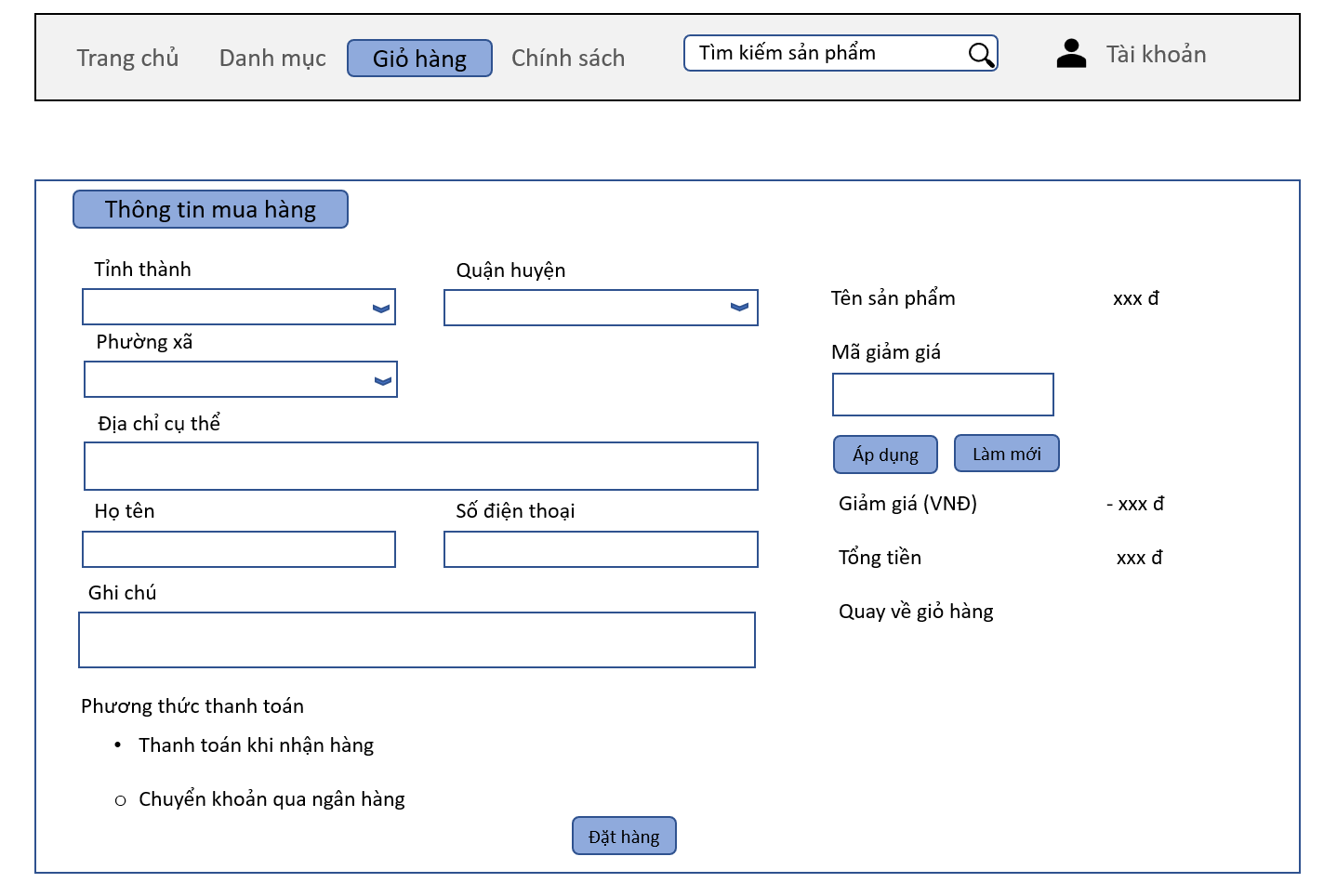
Hình 2. Màn hình danh mục



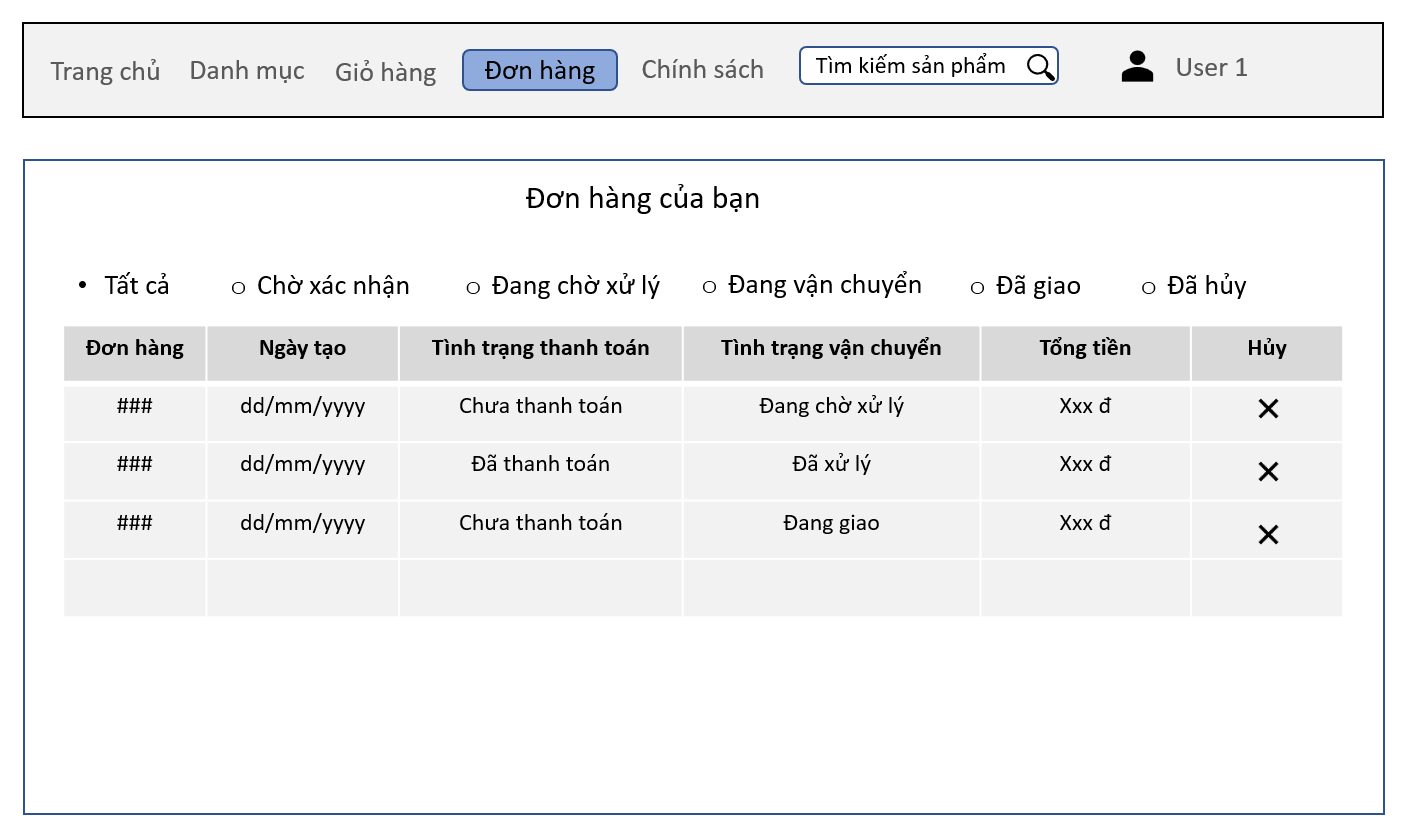
Hình 2. Màn hình xem chi tiết sản phẩm



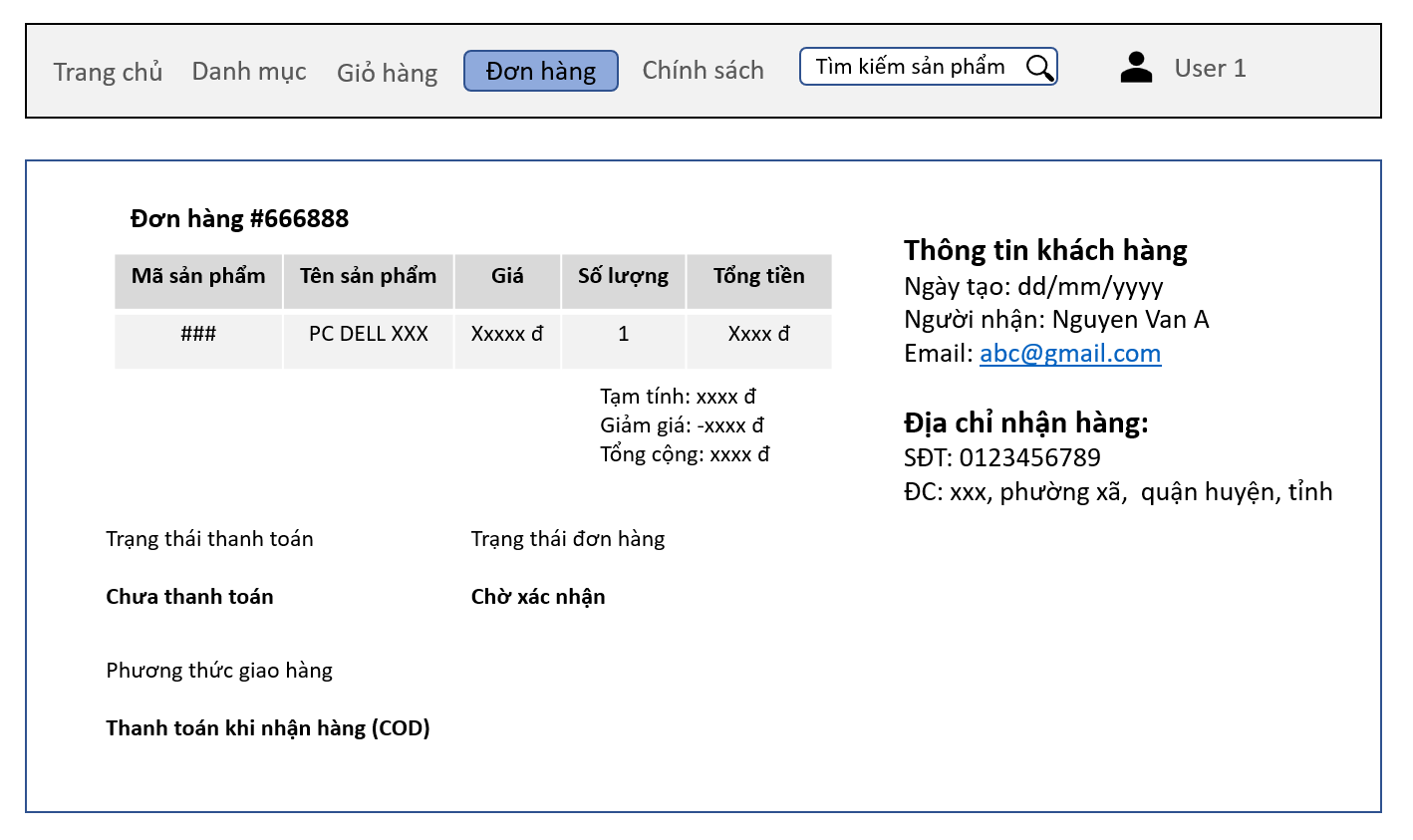
Hình 2. Màn hình của giỏ hàng



Hình 2. Màn hình điền thông tin để đặt hàng

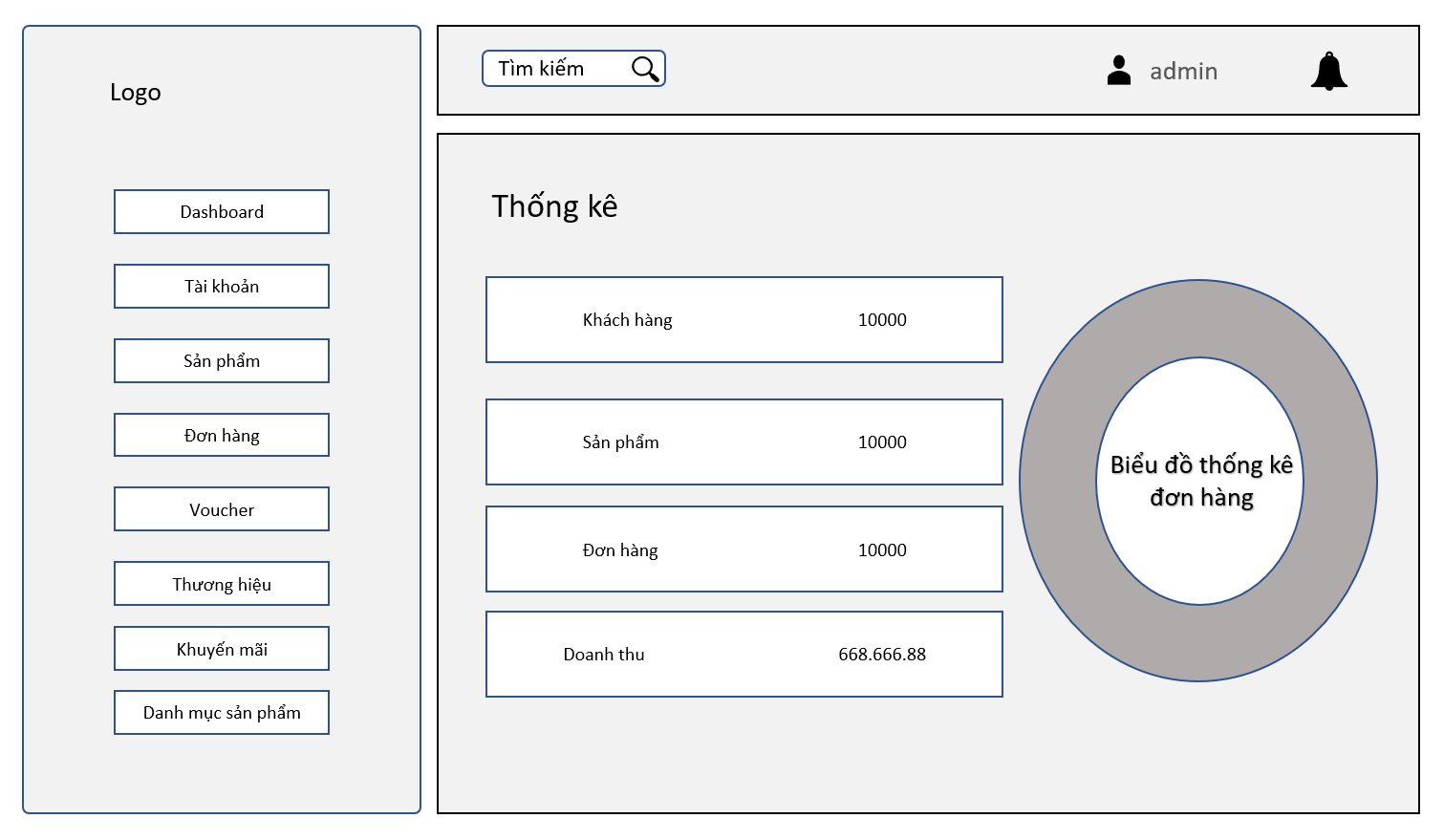


Hình 2. Màn hình của các đơn hàng

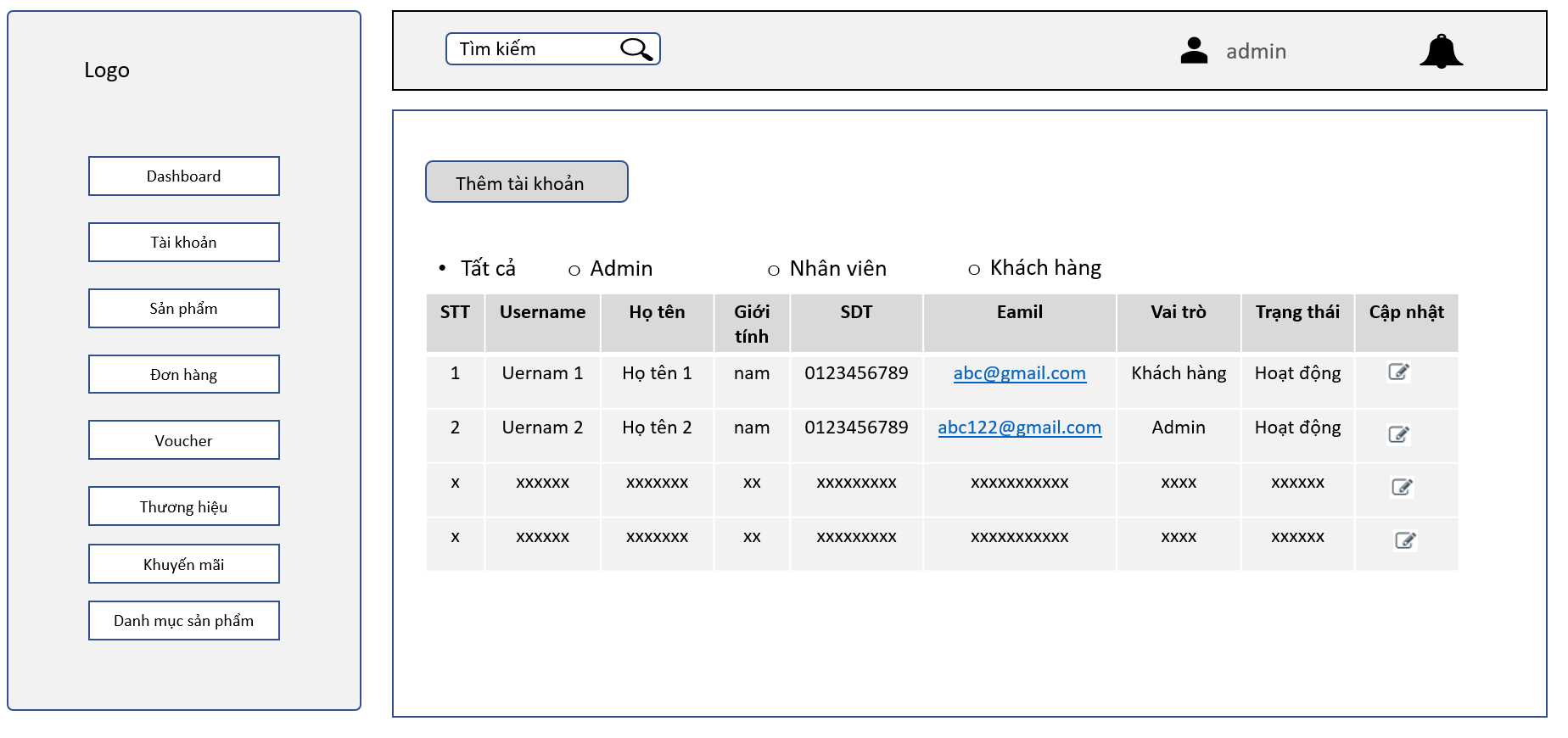


Hình 2. Màn hình chi tiết của một đơn hàng đã đặt

### 2.6.2 Giao diện Admin



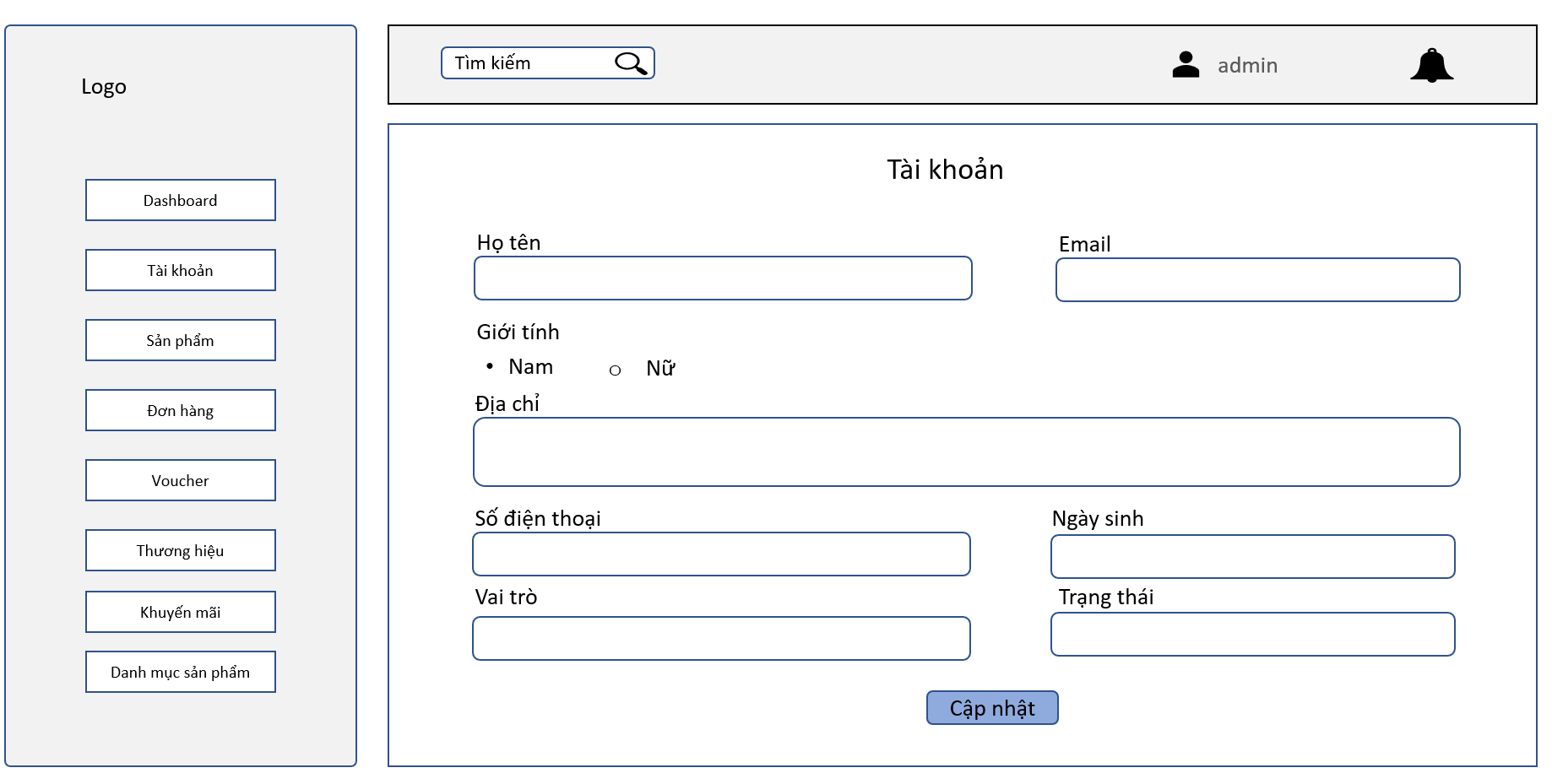
Hình 2. Màn hình trang chủ Admin

****

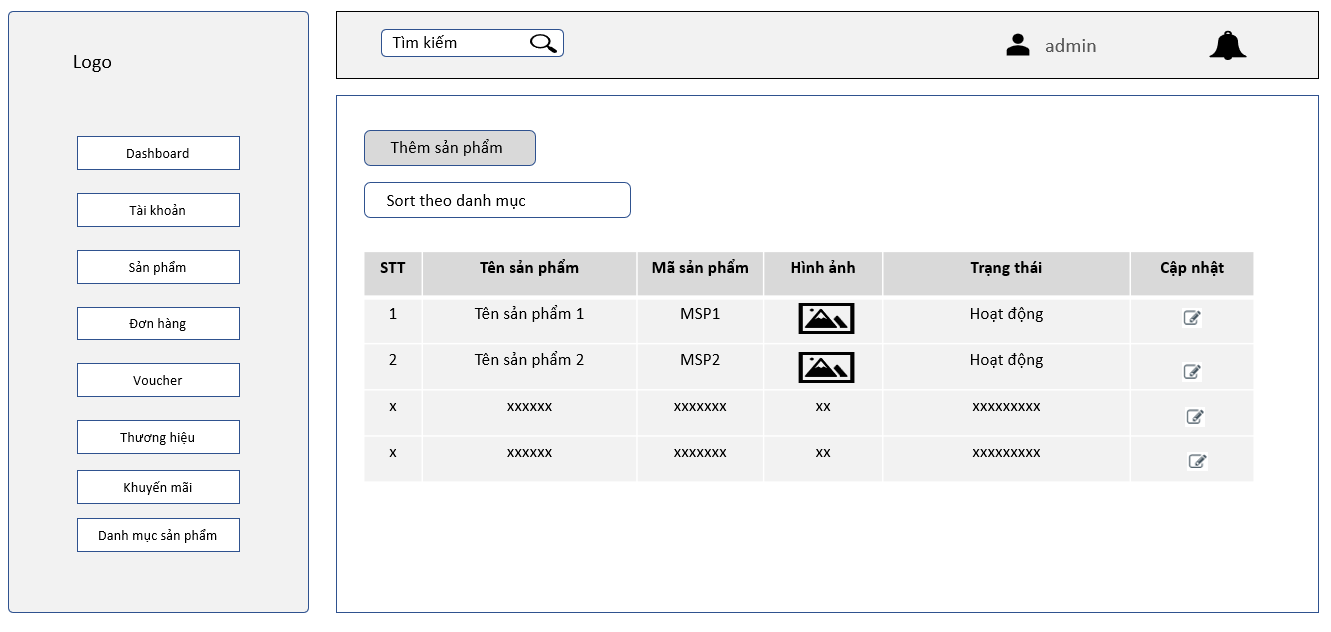
Hình 2. Màn hình quản lý tài khoản



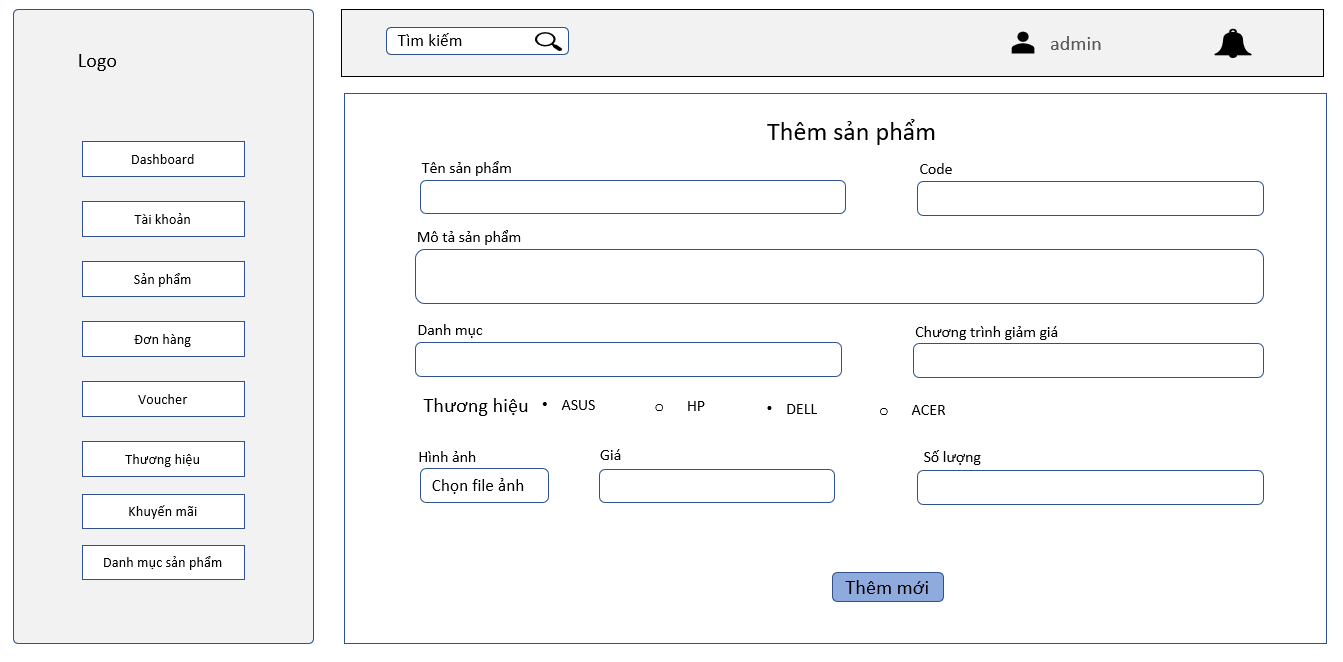
Hình 2. Màn hình thêm mới tài khoản



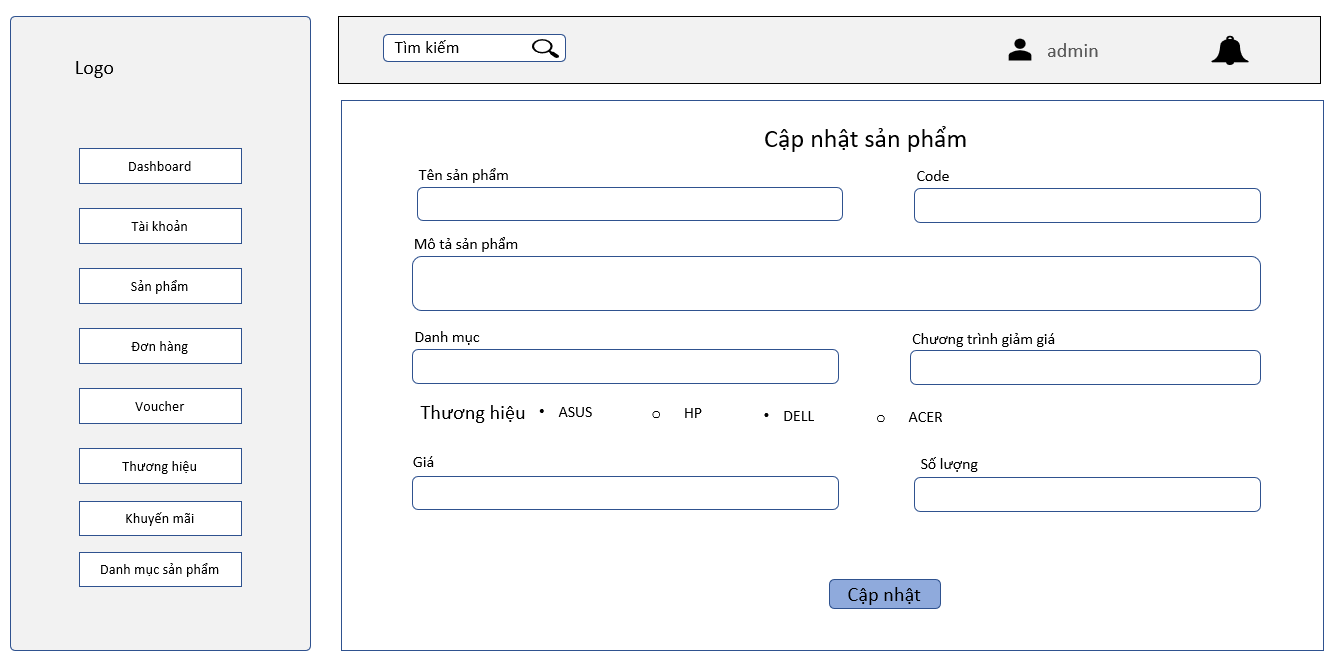
Hình 2. Màn hình cập nhật tài khoản



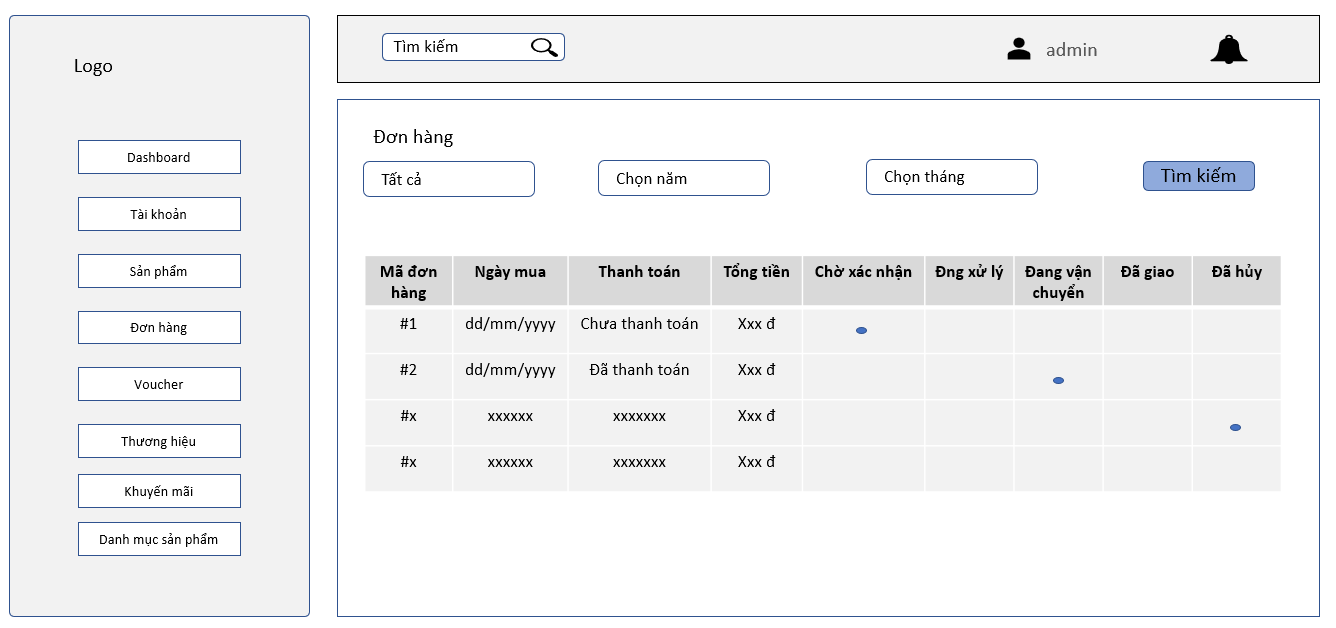
Hình 2. Màn hình quản lý sản phẩm



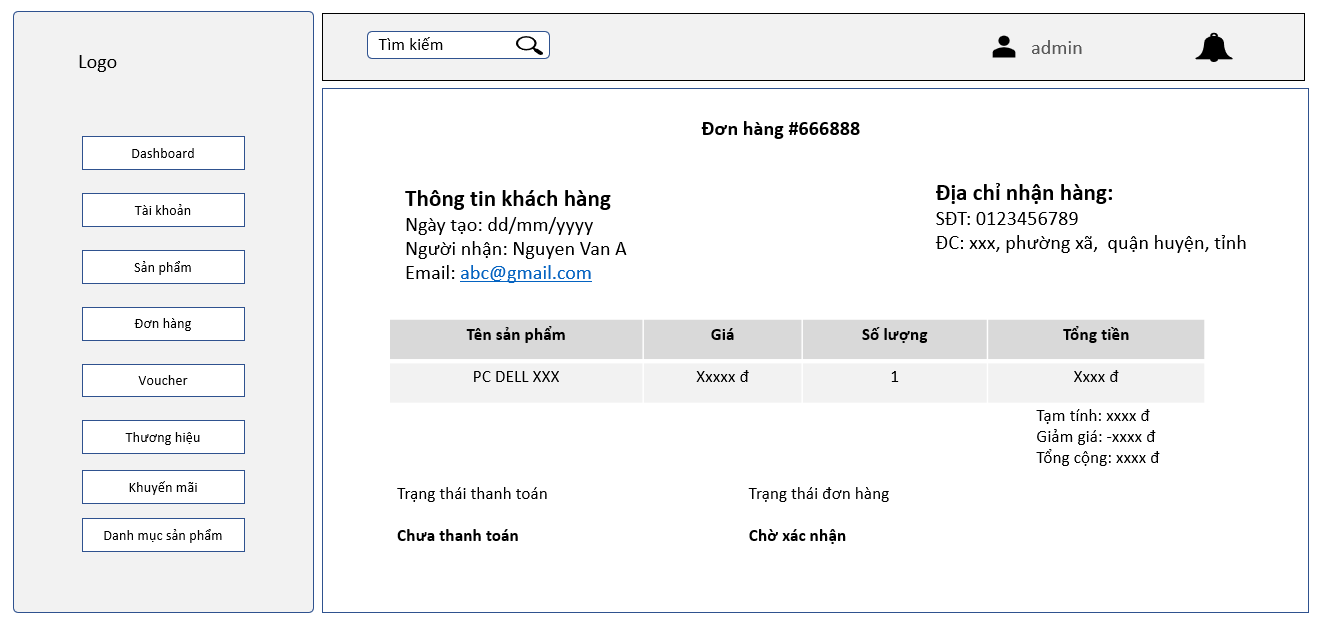
Hình 2. Màn hình thêm sản phẩm mới



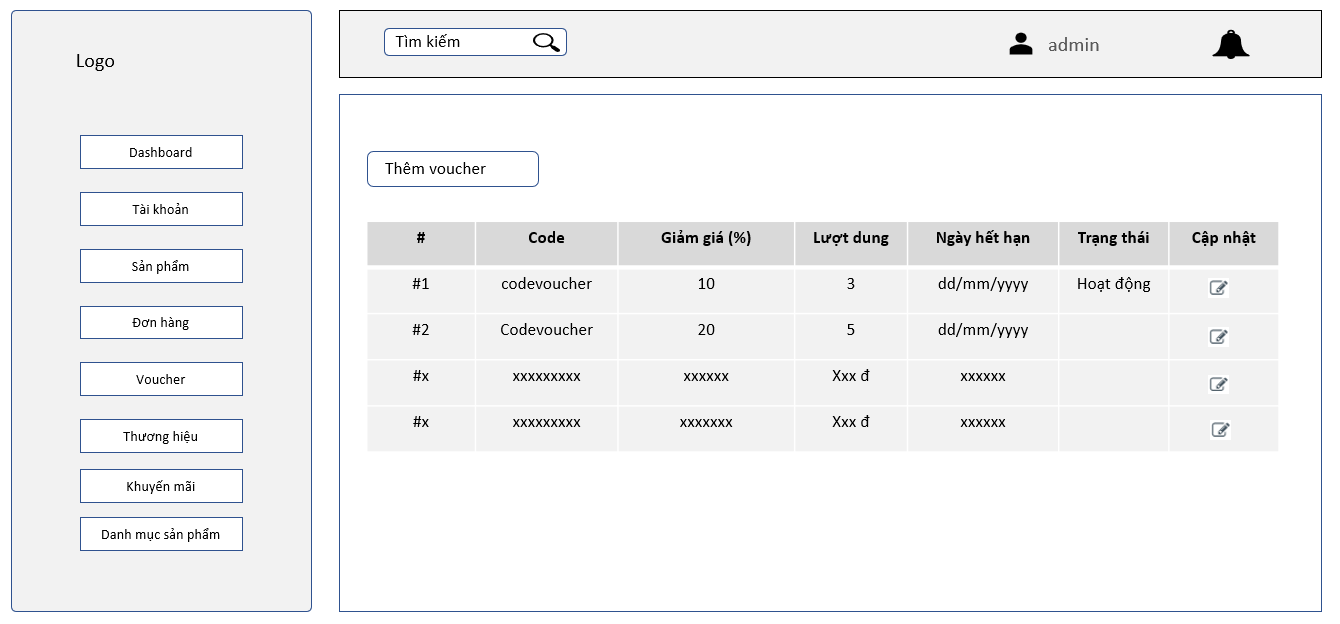
Hình 2. Màn hình cập nhật sản phẩm



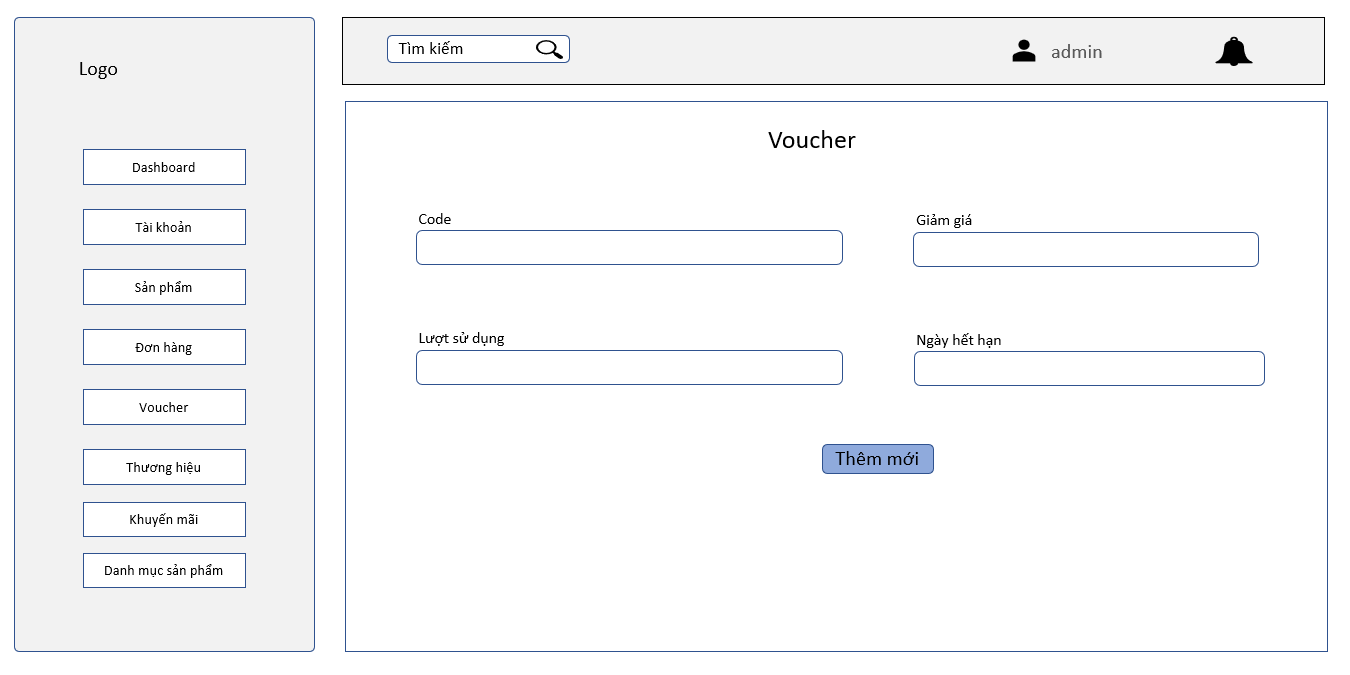
Hình 2. Màn hình quản lý đơn hàng



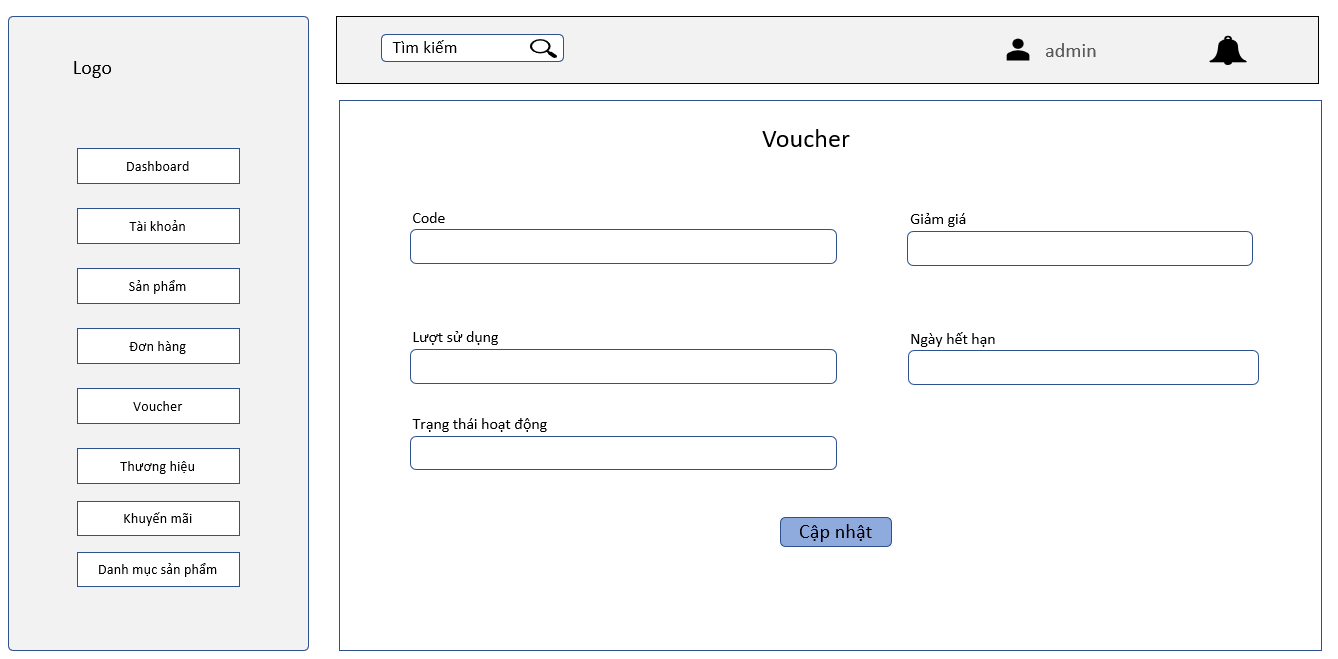
Hình 2. Màn hình chi tiết đơn hàng đã đặt của khách hàng



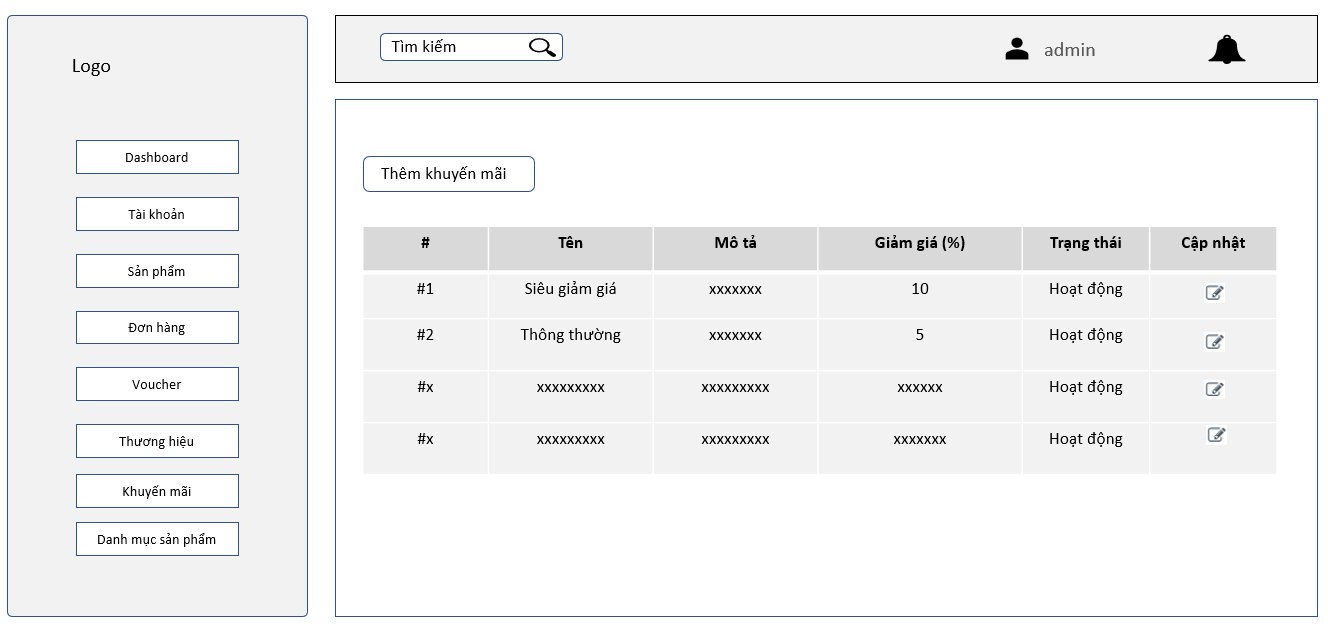
Hình 2. Màn hình quản lý voucher



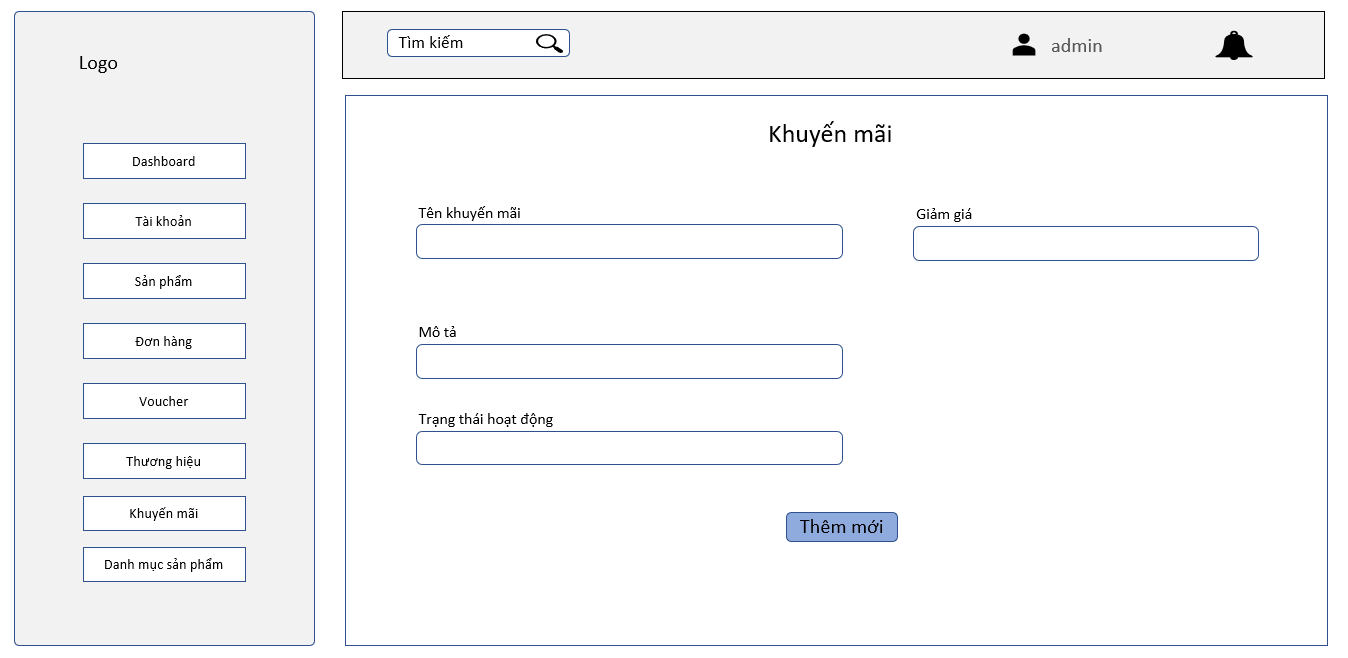
Hình 2. Màn hình thêm mới Voucher



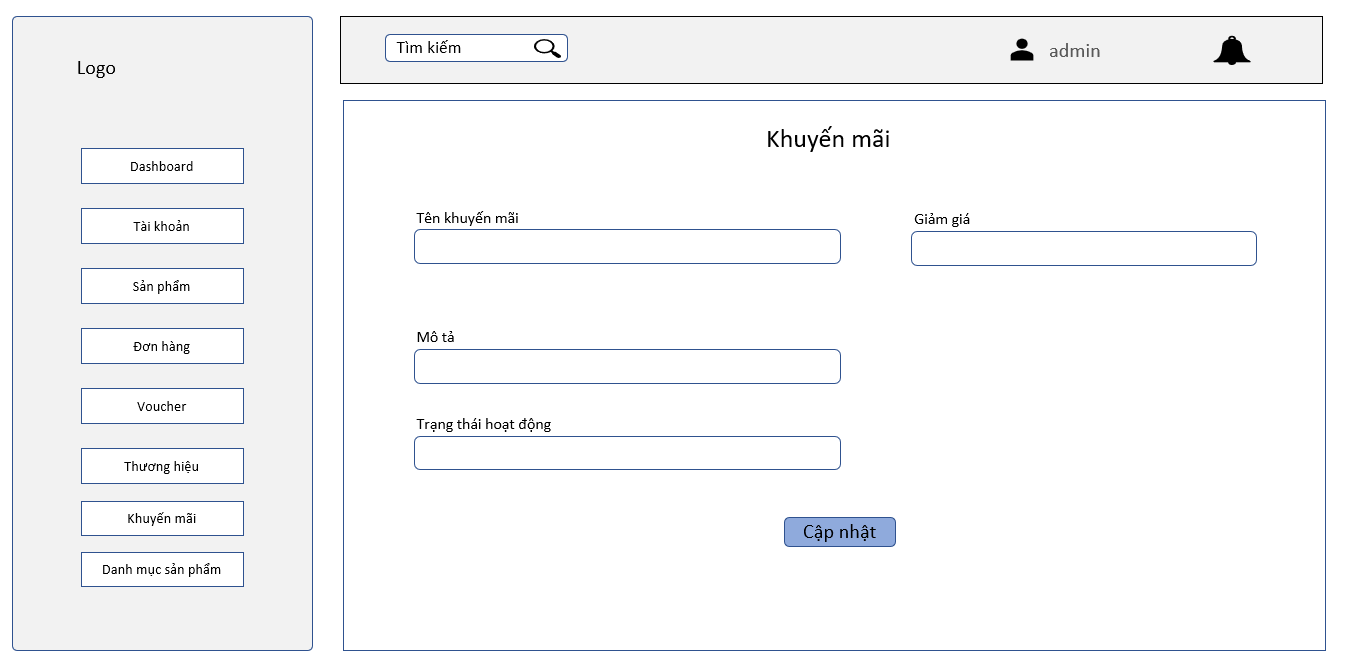
Hình 2. Màn hình cập nhật Voucher



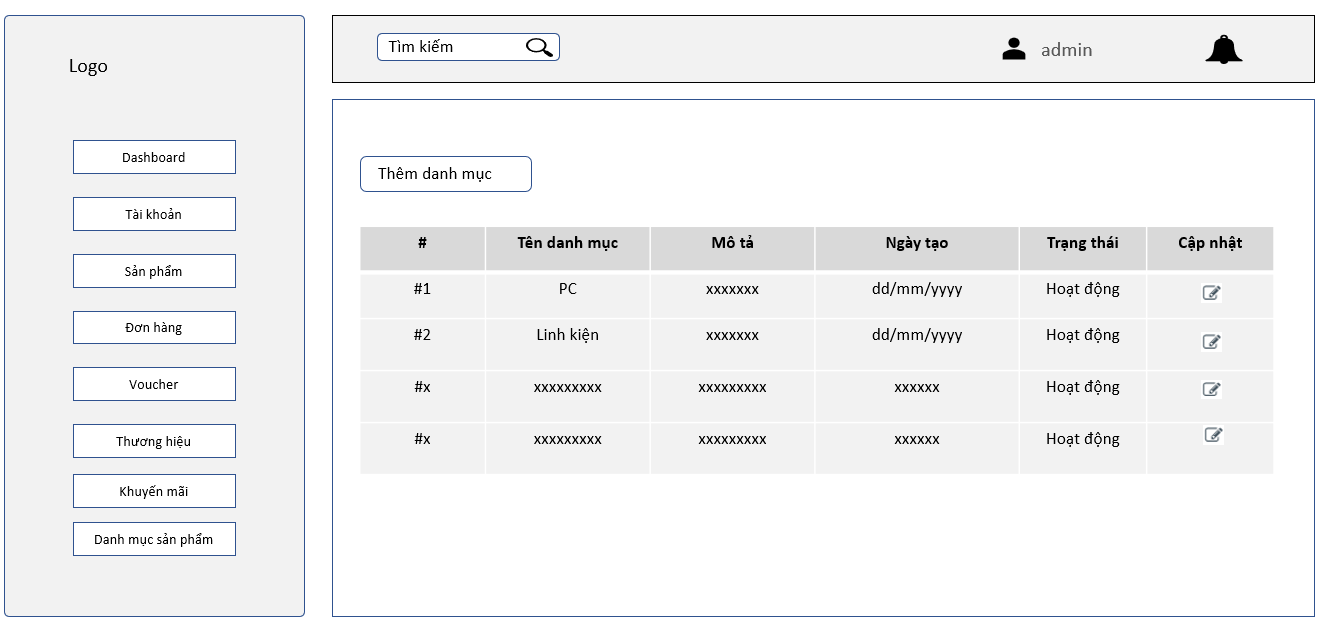
Hình 2. Màn hình quản lý khuyến mãi



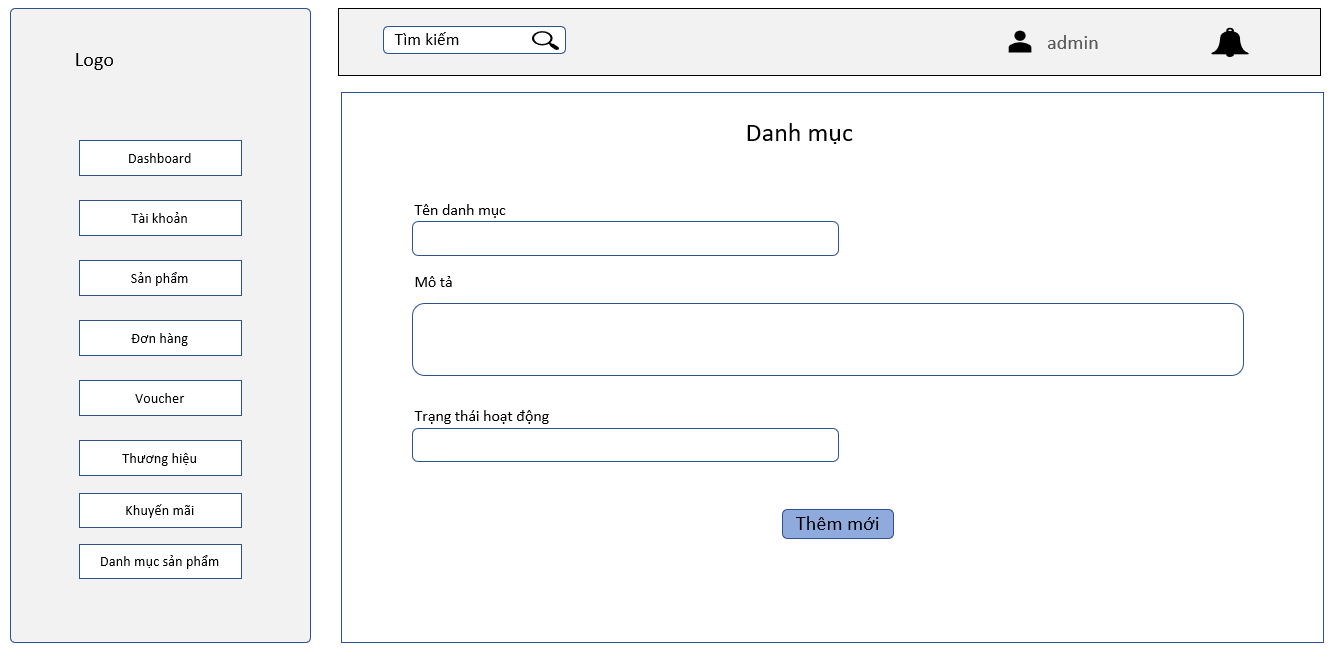
Hình 2. Màn hình thêm mới khuyến mãi



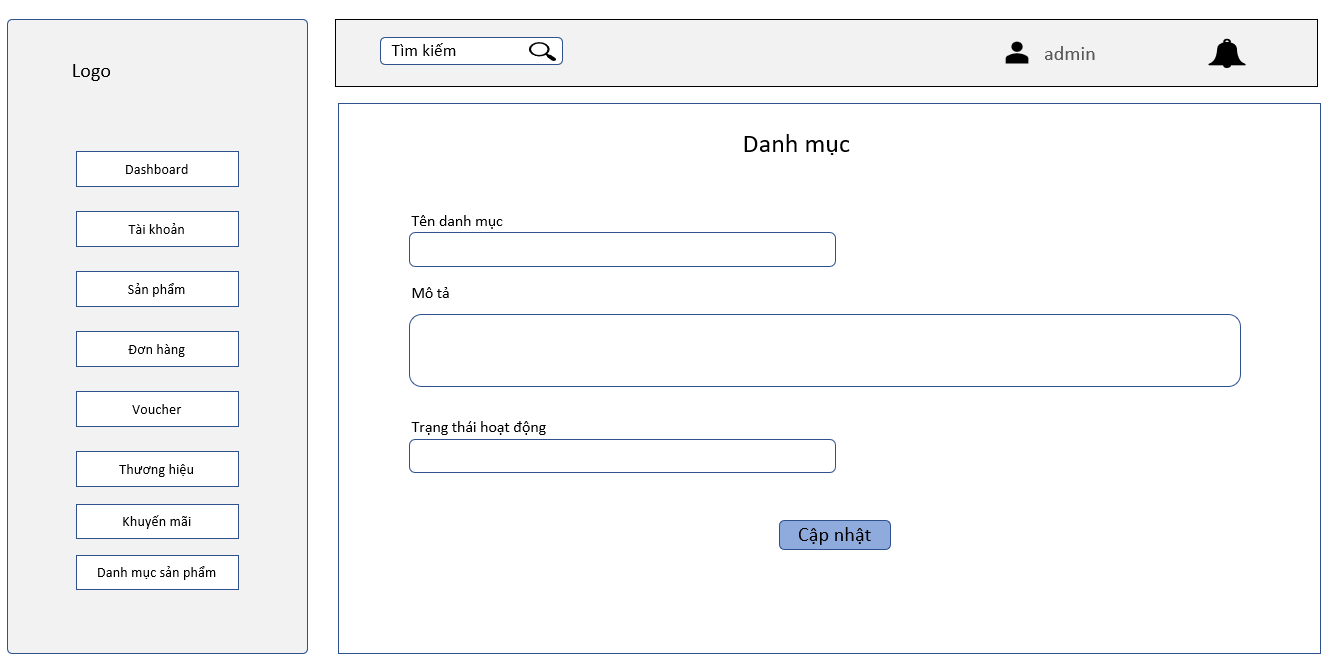
Hình 2. Màn hình cập nhật khuyến mãi



Hình 2. Màn hình quản lý danh mục



Hình 2. Màn hình thêm mới danh mục



Hình 2. Màn hình cập nhật danh mục

# Chương 3. Cài Đặt và kiểm thử chương trình

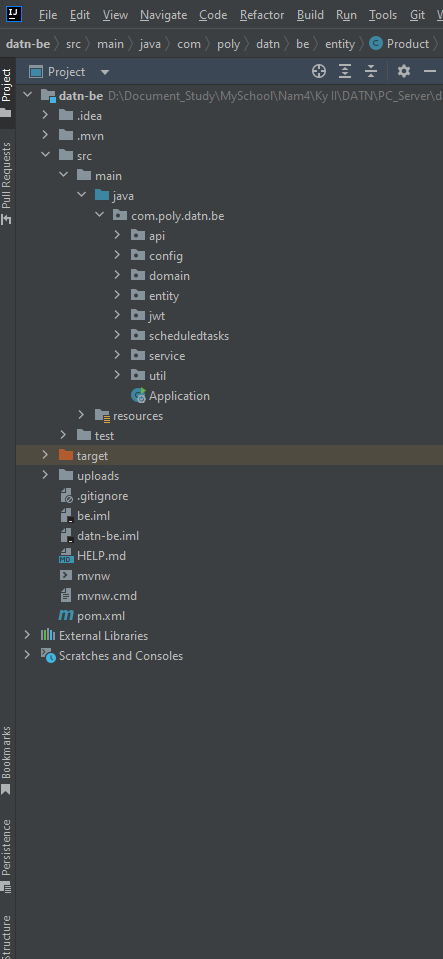
## 3.1 Cài đặt chương trình

### 3.1.1 Môi trường cài đặt

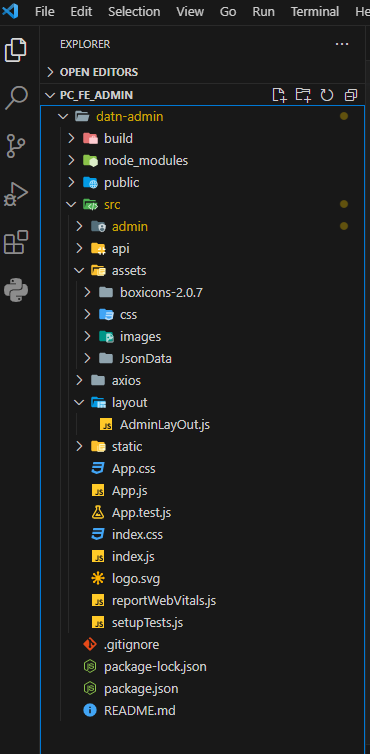
Môi trường cài đặt

* IntelliJ IDEA 2022.3.3
* Visual studio code
* Mysql Workbench 8.0CE
* NodeJs

Một số hình ảnh cài đặt môi trường



Hình 3. Cấu trúc mã nguồn phía Backend



Hình 3. Cấu trúc mã nguồn phía Frontend

### 3.1.2 Cài đặt chương trình phía Backend

Trong hệ thống mà em đã xây dựng thì em đã sử dụng framework Spring boot của ngôn ngữ java. Và dựa trên khảo sát và phân tích thiết kế hệ thống em chọn mô hình Client-Server để dễ dàng quản lý và cập nhật bảo trì hệ thống. Chính vì vậy Spring boot đã giúp em thực hiện cấu hình các chức năng của hệ thống một cách tốt nhất. Spring là toàn bộ hệ sinh thái để phát triển Java bao gồm một số lượng lớn các mô-đun có sẵn như Spring MVC, Spring JDBC, Spring Security và các mô-đun khác.

* Một số ưu điểm của framework.
* Tự động cấu hình

Tự động cấu hình là một cách tiếp cận Spring Boot cốt lõi để giảm số lượng hành động mà nhà em phải thực hiện. Nó tự động cấu hình một ứng dụng Spring dựa trên các phụ thuộc bạn đã thêm trước đó.

* Opinionated Dependencies

Opinionated có nghĩa là Spring Boot tự quyết định tập hợp các bean được định cấu hình mặc định mà bạn có thể ghi đè nếu cần. Hơn nữa, framework này chọn các gói để cài đặt cho các phần dependencies mà chúng ta có thể cần.

* Độc lập

Spring Boot cho phép các nhà phát triển thiết lập và chạy các ứng dụng Spring độc lập, cấp sản xuất mà không cần triển khai chúng lên máy chủ web. Để chạy một ứng dụng Java, bạn cần phải đóng gói ứng dụng đó, chọn, tải xuống và định cấu hình máy chủ web cũng như tổ chức quá trình triển khai.

* Một số kỹ thuật em đã xây cho hệ thống backend dựa vào framework này.
* @Repository  
  public interface AccountRepo extends JpaRepository<Account, Long> {  
    
   @Query("SELECT distinct new com.poly.datn.be.domain.dto.RespAccountDto( a.id, a.username, a.createDate, a.modifyDate, a.isActive , r.name," +  
   " ad.fullname, ad.gender, ad.phone, ad.email, ad.address, ad.birthDate ) FROM Account a " +  
   "inner join AccountDetail ad on a.id = ad.account.id " +  
   "inner join Role r on a.role.id = r.id")  
   List<RespAccountDto> findAllSecond(Pageable pageable);

Sử dụng phương thức @Query để có thể dễ dàng truy vấn tới cơ sở dữ liệu

@Entity  
@Table(name = "accounts")  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
@ToString  
public class Account {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "id", nullable = false)  
 private Long id;  
 @Column(name = "username", nullable = false, unique = true, length = 50)  
 private String username;  
 @Column(name = "password",nullable = false, length = 255)  
 private String password;  
 @Column(name = "create\_date", nullable = false)  
 private LocalDate createDate;  
 @Column(name = "modify\_date", nullable = false)  
 private LocalDate modifyDate;  
 @Column(name = "is\_active", nullable = false)  
 private Boolean isActive;  
 @ManyToOne  
 @JoinColumn(name = "role\_id")  
 private Role role;  
 @OneToMany(mappedBy = "account")  
 @JsonIgnore  
 private Collection<Order> orders;

Sử dụng Anotations để kết nối các bảng với nhau một cách dễ dàng và ngắn gọn

@Component  
@Slf4j  
public class JwtTokenProvider {  
 // Đoạn JWT\_SECRET này là bí mật, chỉ có phía server biết  
 private final String JWT\_SECRET = "secret";  
  
 //Thời gian có hiệu lực của chuỗi jwt  
 private final long JWT\_EXPIRATION = 604800000L;  
  
 // Tạo ra jwt từ thông tin user  
 public String generateToken(CustomUserDetails userDetails) {  
 Date now = new Date();  
 Date expiryDate = new Date(now.getTime() + JWT\_EXPIRATION);  
 // Tạo chuỗi json web token từ id của user.  
 return Jwts.*builder*()  
 .setSubject(userDetails.getAccount().getUsername())  
 .setIssuedAt(now)  
 .setExpiration(expiryDate)  
 .signWith(SignatureAlgorithm.*HS512*, JWT\_SECRET)  
 .compact();  
 }

Java còn rất nổi bật về độ an toàn và bảo mật khi người dùng truy cập vào hệ thống, yêu cầu gửi mã token xác minh đó có phải chính là bạn hay không.

@RestController  
@CrossOrigin("\*")  
public class AccountApi {  
 @Autowired  
 private AccountService accountService;  
  
 @Autowired  
 private AccountDetailService accountDetailService;  
  
  
 @GetMapping(AccountConst.*API\_ACCOUNT\_FIND\_ALL*)  
 public ResponseEntity<?> findAll(@RequestParam("page") Optional<Integer> page,  
 @RequestParam("size") Optional<Integer> size) {  
 Pageable pageable = PageRequest.*of*(page.orElse(1) - 1, size.orElse(9), Sort.Direction.*DESC*, "modifyDate");  
 return new ResponseEntity<>(this.accountService.findAllSecond(pageable), HttpStatus.*OK*);  
 }

Tạo ra các API dễ dàng và ngắn gọn để phục vụ cho các chức năng của hệ thống.

### 3.1.3 Cài đặt chương trình phía Frontend

Trong hệ thống phần frontend này em đã sử dụng framework ReactJs để xây dựng lên giao diện người dùng.

* Một số ưu điểm của framework.
* Hiệu suất cao: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM cho phép ReactJS cập nhật các thay đổi trên trang web một cách nhanh chóng và hiệu quả hơn so với cách truyền thống, giúp tăng tốc độ và hiệu suất của ứng dụng.
* Tái sử dụng: ReactJS cho phép tái sử dụng các thành phần UI, giúp giảm thiểu thời gian và chi phí phát triển. Các thành phần UI có thể được sử dụng lại trong nhiều phần khác nhau của ứng dụng, giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.
* Dễ dàng quản lý trạng thái: ReactJS giúp quản lý trạng thái của ứng dụng một cách dễ dàng. Sử dụng State và Props, ReactJS cho phép các nhà phát triển quản lý trạng thái của các thành phần UI một cách chính xác và dễ dàng.
* Hỗ trợ đa nền tảng: ReactJS không chỉ được sử dụng để phát triển các ứng dụng web, mà còn được sử dụng để phát triển các ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

Dưới đây là một số kỹ thuật em đã sử dụng trong hệ thống.

  const [brand, setBrand] = useState();

  const [page, setPage] = useState(1);

  const [total, setTotal] = useState();

  useEffect(() => {

    onLoad();

  }, [page]);

Sử dụng các state và props để dễ dàng quản lý trạng thái của các component

KẾT LUẬN

TÀI LIỆU THAM KHẢO

1. <https://en.wikipedia.org/wiki/JavaScript>
2. <https://developer.mozilla.org/>
3. <https://en.wikipedia.org/wiki/Java_%28programming_language%29>
4. <https://en.wikipedia.org/wiki/MySQL_Workbench>