|  |
| --- |
|  |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC KHOA HỌC MÁY TÍNH  **XÂY DỰNG WEBSITE BÁN THIẾT BỊ CHĂM SÓC SỨC KHỎE BẰNG SPRING BOOT VÀ REACTJS** |
| |  |  | | --- | --- | | **CBHD:** | **TS. Nguyễn Mạnh Cường** | | **Sinh viên:** | **Nguyễn Ngọc Bình** | | **Mã sinh viên:** | **2020608698** |   Hà Nội – Năm 2024 |
|

# MỤC LỤC

[MỤC LỤC ii](#_Toc174913170)

[MỤC LỤC HÌNH ẢNH iv](#_Toc174913171)

[DANH SÁCH BẢNG BIỂU vi](#_Toc174913172)

[LỜI CẢM ƠN vii](#_Toc174913173)

[LỜI MỞ ĐẦU 1](#_Toc174913174)

[2.1. Lý do lựa chọn đề tài 1](#_Toc174913175)

[2.2. Mục tiêu của đề tài: 1](#_Toc174913176)

[2.3. Bố cục đề tài: 2](#_Toc174913177)

[CHƯƠNG 1: KHẢO SÁT HỆ THỐNG 3](#_Toc174913178)

[1.1. Giới thiệu đề tài 3](#_Toc174913179)

[1.2. Khảo sát hiện trạng 3](#_Toc174913180)

[1.2.1. Cơ cấu tổ chức của khách hàng 3](#_Toc174913181)

[1.2.2. Các nghiệp vụ hiện tại 4](#_Toc174913182)

[1.3. Yêu cầu về chức năng 4](#_Toc174913183)

[1.4. Yêu cầu phi chức năng 5](#_Toc174913184)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 6](#_Toc174913185)

[2.1. Yêu cầu hệ thống 6](#_Toc174913186)

[2.1.1. Phân tích yêu cầu 6](#_Toc174913187)

[2.1.2. Phân tích yêu cầu phi chức năng 7](#_Toc174913188)

[2.2. Xây dựng biểu đồ usecase 8](#_Toc174913189)

[2..3. Biểu đồ usecase hệ thống: 8](#_Toc174913190)

[2..4. Đặc tả chi tiết usecase 8](#_Toc174913191)

[2..5. Phân tích biểu đồ hoạt động usecase 18](#_Toc174913192)

[2..3. Thiết kế cơ sở dữ liệu 41](#_Toc174913193)

[2.3.1. Biểu đồ Entity relationship diagram 41](#_Toc174913194)

[2.3.2. Chi tiết các bảng trong cơ sở dữ liệu 41](#_Toc174913195)

[CHƯƠNG 3: CÀI ĐẶT HỆ THỐNG VÀ MỘT SỐ KẾT QUẢ 50](#_Toc174913196)

[3.1. Công nghệ sử dụng 50](#_Toc174913197)

[3.1.1. Java 50](#_Toc174913198)

[3.1.2. ReactJS 52](#_Toc174913199)

[3.2. Môi trường đã sử dụng 54](#_Toc174913200)

[3.2.1. Eclipse 54](#_Toc174913201)

[3.2.2. Visual Studio Code 54](#_Toc174913202)

[3.3. Một số kết quả đạt được 56](#_Toc174913203)

# MỤC LỤC HÌNH ẢNH

[Hình 2. 1: Biểu đồ usecase hệ thống 8](#_Toc174901249)

[Hình 2. 2: Biểu đồ lớp đăng ký 19](#_Toc174901250)

[Hình 2. 3: Biểu đồ trình tự đăng ký 19](#_Toc174901251)

[Hình 2. 4: Biểu đồ lớp kích hoạt tài khoản 20](#_Toc174901252)

[Hình 2. 5: Biểu đồ trình tự kích hoạt tài khoản 20](#_Toc174901253)

[Hình 2. 6: Biểu đồ lớp chức năng đăng nhập 21](#_Toc174901254)

[Hình 2. 7: Biểu đồ trình tự đăng nhập 22](#_Toc174901255)

[Hình 2. 8: Biểu đồ lớp đổi mật khẩu 22](#_Toc174901256)

[Hình 2. 9: Biểu đồ trình tự đổi mật khẩu 23](#_Toc174901257)

[Hình 2. 10: Biểu đồ lớp cập nhật người dùng 23](#_Toc174901258)

[Hình 2. 11: Biểu đồ trình tự cập nhật thông tin người dùng 24](#_Toc174901259)

[Hình 2. 12: Biểu đồ lớp xem sản phẩm 25](#_Toc174901260)

[Hình 2. 13: Biểu đồ trình tự xem sản phẩm 25](#_Toc174901261)

[Hình 2. 14: Biểu đồ lớp tìm kiếm sản phẩm 26](#_Toc174901262)

[Hình 2. 15: Biểu đồ trình tự chức năng tìm kiếm sản phẩm 27](#_Toc174901263)

[Hình 2. 16: Biểu đồ lớp chức năng thao tác giỏ hàng 27](#_Toc174901264)

[Hình 2. 17: Biểu đồ trình tự thao tác giỏ hàng 28](#_Toc174901265)

[Hình 2. 18: Biểu đồ trình tự chức năng thao tác giỏ hàng 28](#_Toc174901266)

[Hình 2. 19: Biểu đồ lớp chức năng thanh toán 29](#_Toc174901267)

[Hình 2. 20: Biểu đồ trình tự chức năng thanh toán 30](#_Toc174901268)

[Hình 2. 21: Biểu đồ lớp chức năng thao tác đơn hàng 31](#_Toc174901269)

[Hình 2. 22: Biểu đồ trình tự chức năng thao tác đơn hàng 31](#_Toc174901270)

[Hình 2. 23: Biểu đồ trình tự danh sách đơn hàng 32](#_Toc174901271)

[Hình 2. 24: Biểu đồ lớp chức năng quản lý sản phẩm 33](#_Toc174901272)

[Hình 2. 25: Biểu đồ trình tự chức năng quản lý sản phẩm 34](#_Toc174901273)

[Hình 2. 26: Biểu đồ lớp quản lý tài khoản 35](#_Toc174901274)

[Hình 2. 27: Biểu đồ trình tự quản lý tài khoản 36](#_Toc174901275)

[Hình 2. 28: Biểu đồ lớp quản lý đơn hàng 37](#_Toc174901276)

[Hình 2. 29: Biểu đồ trình tự chức năng quản lý đơn hàng 37](#_Toc174901277)

[Hình 2. 30: Biểu đồ lớp quản lý type 38](#_Toc174901278)

[Hình 2. 31: Biểu đồ trình tự quản lý type 39](#_Toc174901279)

[Hình 2. 32: Biểu đồ lớp quản lý Category 39](#_Toc174901280)

[Hình 2. 33: Biểu đồ trình tự chức năng quản lý category 40](#_Toc174901281)

[Hình 2. 34: Biểu đồ Entity relationship diagram 41](#_Toc174901282)

[Hình 2. 35: Bảng User 41](#_Toc174901283)

[Hình 2. 36: Bảng Product 42](#_Toc174901284)

[Hình 2. 37: Bảng Category 43](#_Toc174901285)

[Hình 2. 38: Bảng Type 44](#_Toc174901286)

[Hình 2. 39: Bảng ShoppingCart 44](#_Toc174901287)

[Hình 2. 40: Bảng ShoppingCartItem 45](#_Toc174901288)

[Hình 2. 41: Bảng ShopOrderItem 46](#_Toc174901289)

[Hình 2. 42: Bảng ShopOrder 46](#_Toc174901290)

[Hình 2. 43: Bảng Payment 47](#_Toc174901291)

[Hình 2. 44: Bảng Reset\_password\_token 48](#_Toc174901292)

[Hình 2. 45: Bảng registration\_user\_token 49](#_Toc174901293)

# DANH SÁCH BẢNG BIỂU

[Bảng 1. 1: Yêu cầu chức năng của hệ thống 8](#_Toc174146956)

[Bảng 2. 1: Mô tả bảng User 42](#_Toc174901238)

[Bảng 2. 2: Mô tả bảng Product 43](#_Toc174901239)

[Bảng 2. 3: Mô tả bảng Category 43](#_Toc174901240)

[Bảng 2. 4: Mô tả bảng Type 44](#_Toc174901241)

[Bảng 2. 5: Mô tả bảng ShoppingCart 45](#_Toc174901242)

[Bảng 2. 6: Mô tả bảng ShoppingCartItem 45](#_Toc174901243)

[Bảng 2. 7: Mô tả bảng ShopOrderItem 46](#_Toc174901244)

[Bảng 2. 8: Mô tả bảng ShopOrder 47](#_Toc174901245)

[Bảng 2. 9: Mô tả bảng Payment 48](#_Toc174901246)

[Bảng 2. 10: Mô tả bảng Reset\_password\_token 48](#_Toc174901247)

[Bảng 2. 11: Mô tả bảng registration\_user\_token 49](#_Toc174901248)

# LỜI CẢM ƠN

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép em được bày tỏ lòng biết ơn đến các thầy cô của trường Đại học Công Nghiệp Hà Nội, đặc biệt là các thầy cô khoa Công Nghệ Thông Tin của trường đã tạo điều kiện cho em có một kỳ đồ án tốt nghiệp.

Và em cũng xin chân thành cảm ơn thầy giáo Ts. Nguyễn Mạnh Cường người đã tận tâm hướng dẫn em qua từng buổi nói chuyện, hướng dẫn, thảo luận về các lĩnh vực trong đề tài. Em rất biết ơn các góp ý, sự chỉ dẫn và kiến thức chuyên môn mà thầy giáo đã chia sẻ với em trong quá trình thực hiện báo cáo. Nhờ những lời khuyên và hướng dẫn của thầy đã giúp em đã có cơ hội học hỏi và phát triển kỹ năng của mình.

Không chỉ hướng dẫn và hỗ trợ em hoàn thành đồ án tốt nghiệp một cách thành công, sự đóng góp của thầy giáo còn giúp em hiểu rõ hơn về lĩnh vực chuyên môn và củng cố kiến thức hoàn thành đồ án tốt nghiệp. Nhờ thầy giáo mà em đã có cơ hội nắm bắt những khía cạnh quan trọng và phát triển tư duy sáng tạo. Đồ án này không thể hoàn thành một cách xuất sắc như vậy nếu không có sự giúp đỡ của thầy. Em rất cảm ơn sự tâm huyết và nhiệt tình giúp đỡ trong xuyên suốt quá trình thực hiện báo cáo này.

Trong quá trình thực hiện báo cáo, cũng như là trong quá trình thực hiện đề tài đã chọn, em cũng đã rất cố gắng xây dựng và hoàn thiện sản phẩm. Tuy nhiên trong quá trình thực hiện sẽ khó tránh khỏi những thiết hụt và sai sót. Em rất mong nhận được ý kiến đóng góp từ thầy cô để học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt bài báo cáo tốt nghiệp. Em xin chúc thầy luôn mạnh khỏe, tràn đầy may mắn và thành công trong cuộc sống.

Em xin chân thành cảm ơn.

**Sinh viên thực hiện**

Nguyễn Ngọc Bình

# LỜI MỞ ĐẦU

## Lý do lựa chọn đề tài

Ngày nay, công nghệ thông tin đã có những bước tiến mạnh mẽ theo cả chiều rộng lẫn chiều sâu, cùng với xu thế toàn cầu hóa và kinh doanh quốc tế đang phát triển một cách mạnh mẽ, hoạt động thương mại điện tử ngày càng trở nên đa dạng và phong phú và có sức ảnh hưởng lớn với nền kinh tế của mỗi quốc gia trên toàn cầu. Vì vậy, có thể coi việc phát triển thương mại điện tử là xu hướng phát triển kinh tế hiện nay. Thương mại điện tử giúp người bán mở rộng thị trường, và tiếp cận với nhiều khách hàng tiềm năng, cũng như các khách hàng có nhu cầu dễ dàng tìm kiếm sản phẩm mình mong muốn. Trong thế kỉ 21, khi đời sống vật chất tinh thần của người Việt Nam được nâng cao, nhu cầu về sức khỏe và chăm sóc bản thân cũng gia tăng. Thiết bị chăm sóc sức khỏe trở thành những vật dụng không thể thiếu trong mỗi gia đình để đảm bảo sức khỏe và nâng cao chất lượng cuộc sống. Tuỳ theo nhu cầu và điều kiện của mỗi cá nhân, việc chọn lựa thiết bị chăm sóc sức khỏe như máy đo huyết áp, máy massage, hay các thiết bị hỗ trợ vận động cần được thực hiện một cách dễ dàng và thuận tiện. Chính vì vậy, với mong muốn giúp đỡ người dùng có thể tiếp cận và lựa chọn dễ dàng với các thiết bị chăm sóc sức khỏe, đơn giản hóa việc mua bán và cũng qua đó được củng cố và học hỏi thêm được nhiều kiến thức về lập trình Java với framework Spring Boot và ReactJs, em đã chọn được đề tài cho mình là “Xây dựng website bán thiết bị chăm sóc sức khỏe sử dụng Java Spring Boot và ReactJS” nhằm kết nối giữa người tiêu dùng đang có nhu cầu mua những mặt hàng về thiết bị chăm sóc sức khỏe và người bán.

## Mục tiêu của đề tài:

Mục tiêu của đề tài là tạo ra một website hoàn chỉnh giúp người tiêu dùng có thể tìm đến các mặt hàng liên quan tới thiết bị chăm sóc sức khỏe. Các chức năng của website như việc đăng nhập, mua bán và quản lý sẽ được phát triển qua các tài liệu phân tích và thiết kế, từ đó có thể ứng dụng và tạo ra các chức năng đó trong hệ thống Backend. Sau khi tạo ra thành công và kiểm tra các chức năng đã hoạt động một cách hoàn chỉnh, phần giao diện sẽ được thiết kế để sử dụng và tương tác với các chức năng đó.

## Bố cục đề tài:

Nội dung quyển báo cáo đồ án tốt nghiệp sẽ bao gồm các chương sau: **Chương 1: Khảo sát hệ thống**

Trong chương 1, em sẽ nêu ra lý thuyết cơ sở, các kiến thức liên quan tới các ngôn ngữ chính được sử dụng trong nghiên cứu của đề tài.

**Chương 2: Phân tích thiết kế hệ thống**

Sau khi đưa ra được phát biểu, nêu rõ được công nghệ, cơ sở lý thuyết và xác định được yêu cầu của bài toán, em sẽ đi phân tích và thiết kế các chức năng của bài toán qua các tài liệu.

**Chương 3: Thực nghiệm và kết quả**

Ở chương này, em sẽ tiến hành thực nghiệm thiết kế hệ thống và giao diện từ các kết quả đạt được với phương pháp và kỹ thuật được đưa ra ở chương 2.

**Kết luận:** Cuối cùng là kết luận, tổng hợp các kết quả, các hướng phát triển và mở rộng trong tương lai.

Qua đề tài này em sẽ đúc kết được các kinh nghiệm về việc phân tích thiết kế cũng như xây dựng 1 website. Em hi vọng đề tài này có thể mang lại lợi ích và kết nối người dùng đến với các thiết bị chăm sóc sức khỏe hơn.

# CHƯƠNG 1: KHẢO SÁT HỆ THỐNG

## Giới thiệu đề tài

Với sự phát triển không ngừng của công nghệ thông tin và thương mại điện tử, việc ứng dụng công nghệ vào các hoạt động kinh doanh đang trở thành xu hướng tất yếu. Đặc biệt, trong lĩnh vực chăm sóc sức khỏe, nhu cầu về các thiết bị y tế chất lượng đang ngày càng tăng cao. Việc xây dựng một website bán thiết bị chăm sóc sức khỏe không chỉ giúp các doanh nghiệp mở rộng kênh bán hàng mà còn mang lại nhiều tiện ích cho người tiêu dùng, giúp họ dễ dàng tiếp cận và mua sắm các sản phẩm cần thiết cho sức khỏe.

Đề tài "Xây dựng website bán thiết bị chăm sóc sức khỏe bằng Spring Boot và ReactJS" được lựa chọn với mục tiêu tạo ra một nền tảng thương mại điện tử chuyên cung cấp các thiết bị chăm sóc, hỗ trợ khách hàng tìm kiếm, lựa chọn và mua sắm một cách nhanh chóng và tiện lợi. Công nghệ Spring Boot và ReactJS được chọn để xây dựng hệ thống nhằm đảm bảo tính linh hoạt, hiệu suất cao và dễ dàng bảo trì trong tương lai.

## Khảo sát hiện trạng

### Cơ cấu tổ chức của khách hàng

Khách hàng của hệ thống này là một doanh nghiệp chuyên cung cấp các thiết bị chăm sóc sức khỏe và y tế, bao gồm các sản phẩm như máy massage, máy chạy bộ, máy máy đo huyết áp, và các thiết bị hỗ trợ chăm sóc sức khỏe tại nhà khác. Hiện tại, doanh nghiệp này có một hệ thống quản lý kho và bán hàng truyền thống, chủ yếu dựa vào các phần mềm quản lý cục bộ và các phương pháp bán hàng qua điện thoại, trực tiếp tại cửa hàng.

* **Chủ doanh nghiệp:** Quản lý toàn bộ hoạt động kinh doanh, đưa ra các quyết định chiến lược, và giám sát các hoạt động bán hàng và marketing.
* **Nhân viên bán hàng:** Phụ trách tiếp nhận đơn hàng từ khách hàng qua điện thoại hoặc tại cửa hàng, tư vấn sản phẩm, và thực hiện các giao dịch bán hàng.
* **Nhân viên kho:** Quản lý kho hàng, thực hiện các nghiệp vụ nhập, xuất hàng, và kiểm kê hàng tồn kho.
* **Kế toán:** Phụ trách các công việc liên quan đến tài chính, bao gồm theo dõi doanh thu, lập báo cáo tài chính, và quản lý các chứng từ.

### Các nghiệp vụ hiện tại

Hiện tại, doanh nghiệp gặp phải một số vấn đề khi sử dụng các phương pháp quản lý và bán hàng truyền thống:

* **Nghiệp vụ bán hàng:** Nhân viên bán hàng thường phải nhập liệu thủ công các đơn hàng vào hệ thống quản lý, điều này dễ dẫn đến sai sót và mất nhiều thời gian.
* **Quản lý kho:** Việc quản lý kho hàng chưa được tối ưu hóa, dẫn đến khó khăn trong việc kiểm soát hàng tồn kho, đôi khi xảy ra tình trạng thiếu hoặc thừa hàng hóa.

## Yêu cầu về chức năng

Dựa trên khảo sát hiện trạng, hệ thống website cần đáp ứng các yêu cầu chức năng sau:

Bảng 1. 1: Yêu cầu chức năng của hệ thống

|  |  |
| --- | --- |
| **Yêu cầu** | **Mô tả** |
| Quản lý sản phẩm | Hệ thống phải cho phép quản lý thông tin chi tiết của các sản phẩm, bao gồm tên, mô tả, giá, số lượng tồn kho, hình ảnh, v.v. Hệ thống cho phép tài khoản có quyền để quản lý sản phẩm như:   * Thêm mới sản phẩm: Người dùng cần nhập các thông tin mà hệ thống yêu cầu để thêm một sản phẩm mới. * Sửa sản phẩm: Sản phẩm có thể sửa một hoặc nhiều trường như phần thêm mới sản phẩm. * Xóa sản phẩm: Sản phẩm có thể xóa 1 hoặc nhiều bởi tài khoản có quyền tương ứng. * Xem sản phẩm: Các sản phẩm sau khi tạo hoặc sửa thì sẽ được hiển thị ra danh sách. Tùy thuộc vào mỗi quyền thì danh sách sản phẩn sẽ hiển thị theo một cách khác nhau để có thể quản lý hay để tiếp cận với khách hàng. |
| Giỏ hàng và thanh toán | Khách hàng có thể thêm sản phẩm vào giỏ hàng, xem sản phẩm của mình khi chưa thanh toán, khách hàng có thể thanh toán lại hoặc có thể hủy đơn hàng.  Khách hàng thực hiện thanh toán qua các phương thức khác nhau như có thể thanh toán khi nhận hàng hoặc có thể sử dụng phương thức thanh toán trực tuyến thông qua VNPay. |
| Tìm kiếm sản phẩm | Khách hàng cần có thể tìm kiếm sản phẩm theo nhiều tiêu chí khác nhau như tên sản phẩm, loại sản phẩm, giá cả, v.v. |
| Quản lý tài khoản người dùng | Hệ thống cần cho phép người dùng đăng ký, đăng nhập, và quản lý thông tin tài khoản của mình như tải ảnh đại diện của người dùng, cập nhật lại họ và tên và các trường thông tin khác trừ username và email. |

## Yêu cầu phi chức năng

Ngoài các yêu cầu về chức năng, hệ thống cần đáp ứng các yêu cầu phi chức năng sau:

* **Giao diện người dùng thân thiện:** Website phải có giao diện trực quan, dễ sử dụng, phù hợp với nhiều đối tượng khách hàng khác nhau.
* **Hiệu suất cao:** Hệ thống phải đảm bảo khả năng xử lý đồng thời nhiều giao dịch mà không làm giảm tốc độ truy cập của người dùng.
* **Bảo mật:** Hệ thống cần phải bảo vệ thông tin cá nhân và tài khoản của khách hàng, đảm bảo an toàn trong quá trình giao dịch.
* **Tuân thủ các quy định về báo cáo tài chính:** Các báo cáo tài chính được xuất ra từ hệ thống phải tuân thủ các quy định hiện hành của Bộ Tài Chính.

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Yêu cầu hệ thống

### Phân tích yêu cầu

Yêu cầu hệ thống đối với Khách vãng lai (Customer):

* Xem sản phẩm: Khách vãng lai có thể xem danh sách hoặc chi tiết các sản phẩm được đăng bày bán ở trên website.
* Đăng ký: Khách vãng lai có thể đăng ký tài khoản
* Đăng nhập: Khách vãng lai có thể đăng nhập bằng tài khoản đã đăng ký.
* Kích hoạt tài khoản: Khách vãng lai kích hoạt tài khoản thông qua gửi email.
* Đổi mật khẩu: Khách vãng lai có thể lấy lại và thay đổi mật khẩu thông qua Email đã đăng ký tài khoản.
* Tìm kiếm sản phẩm: Khách vãng lai có thể tìm kiếm sản phẩm theo tên sản phẩm.

Yêu cầu hệ thống đối với Người dùng (User):

* Tất cả các quyền hạn giống với quyền hạn của Khách vãng lai.
* Cập nhật thông tin cá nhân: Người dùng có thể cập nhật các thông tin cá nhân của mình bao gồm Hình ảnh, Số điện thoại, Địa chỉ, Họ tên, Tên.
* Giỏ hàng: Người dùng có thể thêm hàng vào giỏ và có thể thêm sửa xóa hàng hay số lượng hàng trong giỏ hàng của mình.
* Thanh toán: Người dùng có thể chọn sản phẩm đơn lẻ hoặc các sản phẩm trong giỏ hàng để tiến hành thanh toán theo một trong hai phương thức thanh toán, hoặc có thể vào đơn hàng chưa thanh toán để tiến hành thanh toán.
* Đơn hàng: Người dùng có thể theo dõi, thao tác như thanh toán đơn hàng, hủy đơn hàng và xem chi tiết đơn hàng của mình.

Yêu cầu hệ thống đối với Người quản trị (Admin):

* Tất cả các quyền hạn của Người dùng
* Quản lý sản phẩm: Quản trị viên có thể xem, thêm, sửa, xóa tất cả sản phẩm.
* Quản lý tài khoản: Quản trị viên có thể xem, thêm, sửa, thay đổi quyền và xóa tài khoản.
* Quản lý đơn hàng: Quản trị viên có thể xem các đơn hàng, xóa đơn hàng và xem tổng doanh thu các đơn hàng, lọc các đơn hàng theo trạng thái, ngày tạo đơn hàng.
* Quản lý type của sản phẩm: Quản trị viên có thể xem, thêm và xóa tất cả các type của sản phẩm.
* Quản lý category của sản phẩm: Quản trị viên có thể xem, thêm và xóa tất cả các category của sản phẩm.

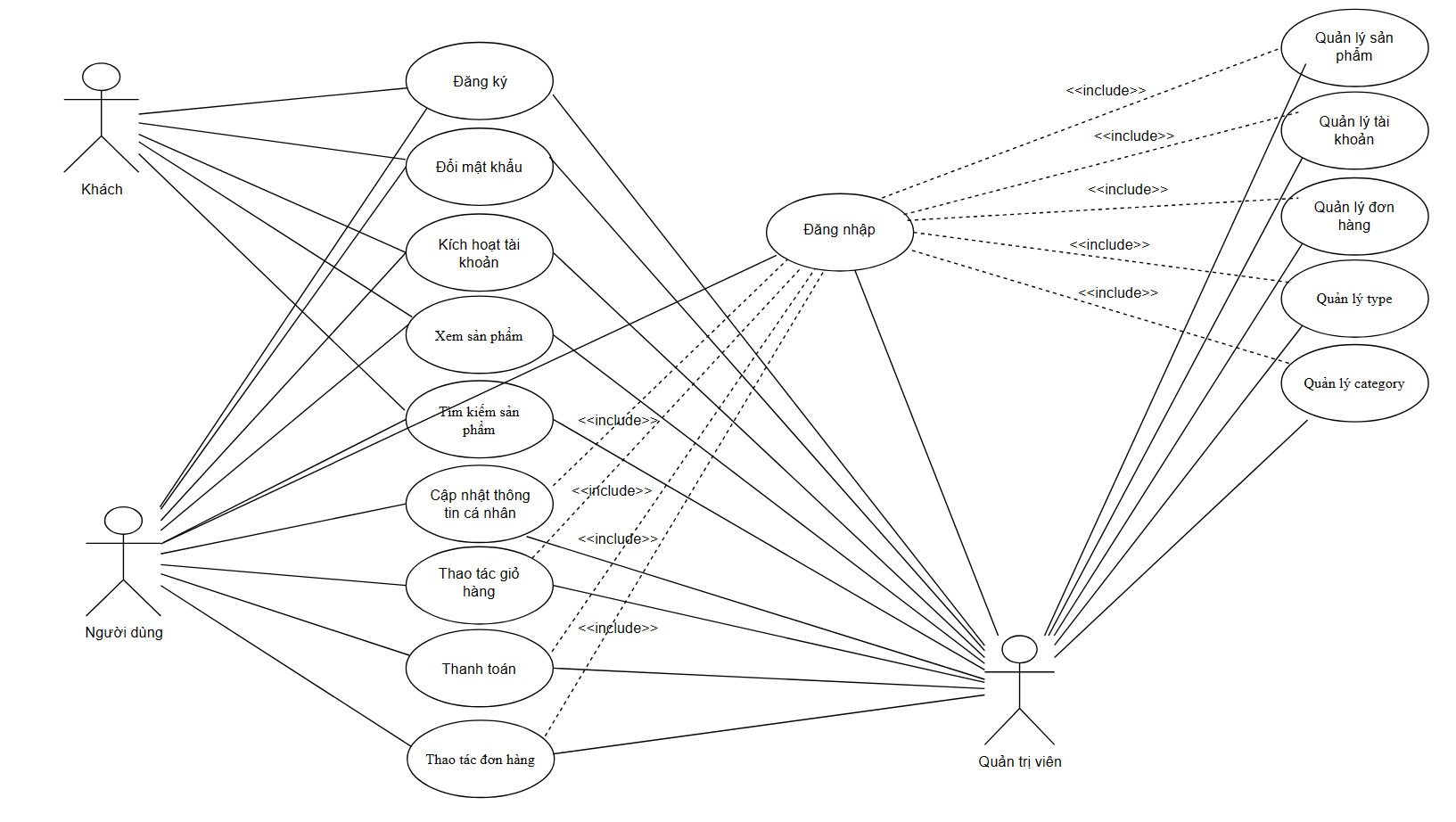
### Phân tích yêu cầu phi chức năng

Các yêu cầu phi chức năng bao gồm:

* ***Hiệu năng***: một hệ thống nên có một hiệu năng tốt và ổn định. Tránh những lỗi logic quá rườm rà hay tích hợp quá nhiều thứ vào trong một chức năng mà em sẽ tách chúng ra riêng biệt để hệ thống có thể hoạt động một cách nhanh chóng.
* ***Bảo mật****:* Về vấn đề bảo mật của người dùng, em sẽ mã hóa mật khẩu của người dùng khi đăng kí và sử dụng bảo mật cors trong Backend đồng thời xóa thông tin của người dùng ở bộ nhớ trang web sau khi đăng xuất. Những tính năng của người dùng và admin chỉ có thể thông qua đăng nhập để lấy token mới có thể truy cập vào.
* ***Phân quyền***: Hệ thống phân quyền cho User và Admin, những trang quản lý của Admin người dùng sẽ không thể truy cập bằng bất cứ cách nào, đồng thời trong hệ thống Backend cũng sẽ phân quyền sử dụng thư viện security để tránh User có thể gọi và lấy dữ liệu từ API ra.
* ***Mô hình***: Hệ thống Backend sẽ được em thiết kế theo mô hình 3 tầng [5] theo thứ tự từ cao xuốn thấp: Presentation layer – Business logic Layer – Data access layer. Tầng trên chứa các controller sẽ tạo ra các Restfull API dùng để giao tiếp với Frontend, tầng giữa sẽ sử dụng các service để xử lý logic được nhận data từ tầng dưới, tầng dưới cùng sẽ dùng để giao tiếp với dữ liệu.
* ***Nguyên tắc****:* Tuân thủ theo nguyên tắc SOLID [4], bao gồm:
  + Single responsibility: 1 class chỉ giữ một trách nhiệm duy nhất.
  + Open/closed: Có thể thoải mái mở rộng 1 class bằng nhiều phiên bản, nhưng không được sửa đổi bên trong class đó.
  + Liskov substitution: Trong một chương trình, các object của class con có thể thay thế class cha mà không làm thay đổi tính đúng đắn của chương trình.
  + Interface segregation: Thay vì dùng 1 interface lớn, em sẽ tách thành nhiều interface nhỏ, với nhiều mục đích cụ thể.
  + Dependency inversion: Các class cấp cha không phụ thuộc vào các class lớp con, chúng sẽ giao tiếp thông qua 1 Interface.

## Xây dựng biểu đồ usecase

### Biểu đồ usecase hệ thống:



Hình 2. 1: Biểu đồ usecase hệ thống

### Đặc tả chi tiết usecase

1. Usecase đăng ký tài khoản

* Mô tả vắn tắt: Use case này cho phép người dùng đăng tài khoản.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu khi người dùng kích vào nút “Đăng ký” trên màn đăng nhập. Hệ thống sẽ yêu cầu nhập các thông tin (firstname, lastname, username, email, password, address, phone) để tạo tài khoản.

Người dùng kích vào nút “Đăng ký”. Hệ thống sẽ lưu thông tin của khách hàng vào bảng User và mở hộp thoại đã gửi email kích hoạt tài khoản.

Use case kết thúc.

* + Luồng rẽ nhánh:

Tại bất cứ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và usecase kết thúc.

Tại bước 2 trong luồng cơ bản khi người dùng nhập thông tin không hợp lệ. Hệ thống sẽ hiển thị thông báo lỗi ra ngoài và cần người dùng nhập lại. Người dùng có thể nhập lại để tiếp tục thao tác.

* + Các yêu cầu đặc biệt: Người dùng cần nhập đúng định dạng các thông tin yêu cầu để tạo tài khoản.
  + Tiền điều kiện: Không có.
  + Hậu điều kiện: Đăng ký thành công sẽ có một bản ghi mới trong bảng USERS.
  + Điểm mở rộng: Không có.

1. Usecase kích hoạt tài khoản

* Mô tả vắn tắt: Use case này cho phép người dùng kích hoạt tài khoản.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi người dùng đăng kí thành công tài khoản hoặc đăng nhập với tài khoản chưa kích hoạt. Hệ thống sẽ hiển thị một hộp thoại kích hoạt tài khoản và khách hàng sẽ vào gmail kích hoạt tài khoản.

Use case kết thúc.

* + Luồng rẽ nhánh:

Tại bước 1 khi hiện hộp thoại, người dùng có thể ấn vào nút “Resend” để gửi lại thông tin kích hoạt tài khoản qua email.

* + Các yêu cầu đặc biệt: Không có.
  + Tiền điều kiện: Người dùng đã đăng kí thành công tài khoản và chưa kích hoạt tài khoản.
  + Hậu điều kiện: Kích hoạt tài khoản thành công sẽ cập nhật trạng thái của tài khoản status là “Active” vào dữ liệu tài khoản với id tương ứng bảng User.
  + Điểm mở rộng: Không có.

1. Usecase đăng nhập

* Mô tả vắn tắt: Use case này cho phép người dùng đăng nhập.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi người dùng ấn vào nút “Đăng nhập”. Giao diện đăng nhập hiện ra

Người dùng nhập các thông tin đăng nhập gồm Username và Password rồi ấn nút “Đăng nhập” để tiến hành đăng nhập.

Use case kết thúc.

* + Luồng rẽ nhánh:

Tại bước 2, nếu người dùng nhập sai thông tin tài khoản sẽ hiện ra thông báo lỗi và yêu cầu nhập lại.

Sau khi đăng nhập xong nếu tài khoản chưa kích hoạt sẽ hiện ra hộp thoại kích hoạt tài khoản.

* + Các yêu cầu đặc biệt: Không có.
  + Tiền điều kiện: Khách hàng đã đăng kí thành công tài khoản.
  + Hậu điều kiện: Tạo ra token đăng nhập qua JWT và lưu thông tin đăng nhập.
  + Điểm mở rộng: Không có.

1. Usecase đổi mật khẩu

* Mô tả vắn tắt: Use case này cho phép người dùng đổi mật khẩu.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi khách hàng ấn vào nút “Quên mật khẩu” ở giao diện đăng nhập. Trang gửi email kích hoạt để đổi mật khẩu hiện ra.

Người dùng vào email ấn vào link chuyển trang tới trang đổi mật khẩu. Nhập thông tin mật khẩu mới vào ô input.

Ấn Reset password để tiến hành đổi mật khẩu.

Use case kết thúc.

* + Luồng rẽ nhánh:

Tại bước 1, người dùng có thể ấn nút Resend để gửi lại email.

* + Các yêu cầu đặc biệt: Không có.
  + Tiền điều kiện: Khách hàng đã có tài khoản.
  + Hậu điều kiện: Lưu mật khẩu mới vào database User.
  + Điểm mở rộng: Không có.

1. Usecase cập nhật thông tin người dùng

* Mô tả vắn tắt: Use case này cho phép người dùng thay đổi thông tin tài khoản cá nhân.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi khách hàng ấn vào nút “Update” ở trang thông tin tài khoản. Hộp thoại cập nhật hiện ra bao gồm các thông tin tài khoản trước đó.

Người dùng nhập thông tin muốn thay đổi hoặc có thể không thay đổi gì. Người dùng ấn nút cập nhật để lưu thông tin sau khi thay đổi.

Use case kết thúc.

* + Luồng rẽ nhánh:

Tại bước 2, người dùng có thể nhấn nút Close thoát khỏi hộp thoại cập nhật.

* + Các yêu cầu đặc biệt: Người dùng không được xóa để trống 1 vài thông tin không được null khi ấn cập nhật.
  + Tiền điều kiện: Người dùng đã đăng nhập.
  + Hậu điều kiện: Lưu thông tin cá nhân mới của người dùng vào data User trong bảng.
  + Điểm mở rộng: Không có.

1. Usecase xem sản phẩm

* Mô tả vắn tắt: Use case này cho phép người dùng xem thông tin sản phẩm.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi khách hàng ấn vào nút một trong các danh mục trên thanh navbar. Danh sách sản phấm sẽ hiển thị ra.

Người dùng ấn vào sản phẩm, trang chi tiết sản phẩm hiện ra.

Use case kết thúc.

* + Luồng rẽ nhánh: Không có.
  + Các yêu cầu đặc biệt: Không có.
  + Tiền điều kiện: Không có.
  + Hậu điều kiện: Không có.
  + Điểm mở rộng: Không có.

1. Usecase tìm kiếm sản phẩm

* Mô tả vắn tắt: Use case này cho phép người dùng tìm kiếm thông tin sản phẩm.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi khách hàng nhập thông tin vào ô tìm kiếm và ấn vào nút tìm kiếm.

Danh sách các sản phẩm được tìm kiếm hiện ra.

Use case kết thúc.

* + Luồng rẽ nhánh:

Tại bước 1, khi người dùng nhập từ khóa vào ô input, 1 danh sách gợi ý các sản phẩm hiện ra. Người dùng có thể click vào một trong các sản phẩm đó để xem chi tiết sản phẩm.

* + Các yêu cầu đặc biệt: Không.
  + Tiền điều kiện: Không.
  + Hậu điều kiện: Không.
  + Điểm mở rộng: Không có.

1. Usecase thao tác giỏ hàng

* Mô tả vắn tắt: Use case này cho phép người dùng xem và thao tác như thêm, sửa, xóa giỏ hàng của mình.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi khách hàng ấn vào nút “Giỏ hàng” ở thanh navbar. Hộp thoại giỏ hàng hiện ra.

Người dùng xem và chỉnh sửa các thông tin liên quan tới giỏ hàng bao gồm xóa từng sản phẩm, xóa tất cả sản phẩm, tăng giảm số lượng sản phẩm.

Use case kết thúc.

* + Luồng rẽ nhánh: Không có.
  + Các yêu cầu đặc biệt: Không có.
  + Tiền điều kiện: Người dùng đã đăng nhập.
  + Hậu điều kiện: Lưu thông tin các sản phẩm vào bảng ShoppingCart, ShoppingCartItem, Product.
  + Điểm mở rộng: Không có.

1. Usecase thanh toán

* Mô tả vắn tắt: Use case này cho phép người dùng thanh toán các sản phẩm muốn mua.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi người dùng ấn vào nút ‘Thanh toán” ở trong giỏ hàng hoặc nhất vào “Mua ngay” trong trang thông tin chi tiết sản phẩm. Trang thanh toán hiện ra.

Người dùng điều thay đổi các thông tin thanh toán bao gồm số lượng sản phẩm, phương thức thanh toán và ấn vào nút thanh toán.

Use case kết thúc.

* + Luồng rẽ nhánh: không có.
  + Các yêu cầu đặc biệt: Không.
  + Tiền điều kiện: Người dùng đã đăng nhập.
  + Hậu điều kiện: Lưu thông tin đơn hàng vào trong bảng ShopOrder.
  + Điểm mở rộng: Không có.

1. Usecase thao tác đơn hàng

* Mô tả vắn tắt: Use case này cho phép người dùng xem vào thay đổi thông tin đơn hàng.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi người dùng ấn vào nút “Đơn hàng” ở trên thanh header. Trang đơn hàng của người dùng hiện ra.

Người dùng thay đổi thông tin đơn hàng với đơn hàng trong trạng thái NOT\_PAY như ấn vào nút “Đã nhận hàng” hoặc với các đơn hàng trong trạng thái PROCESSING với thao tác hủy đơn hàng hoặc thanh toán cho đơn hàng.

Use case kết thúc.

* + Luồng rẽ nhánh: Không có.
  + Các yêu cầu đặc biệt: Không có.
  + Tiền điều kiện: Người dùng đã thanh toán trước đó.
  + Hậu điều kiện: Lưu thông tin đơn hàng vào bảng ShopOrder
  + Điểm mở rộng: Không có.

1. Usecase quản lý sản phẩm

* Mô tả vắn tắt: Use case này cho phép người quản trị thao tác và thay đổi thông tin của tất cả sản phẩm.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi người quản trị ấn vào nút “Tới trang quản trị” ở trên thanh header, ngay sau đó trang quản lý hiện ra và người quản trị chọn vào mục “Quản lý sản phẩm”.

Quản trị viên xem và thay đổi các thông tin của sản phẩm bao gồm cập nhật sản phẩm, xóa nhiều sản phẩm, tạo sản phẩm mới, lọc danh sách các sản phẩm.

Use case kết thúc.

* + Luồng rẽ nhánh: Không có.
  + Các yêu cầu đặc biệt: Quản trị viên không được xóa để trống 1 vài thông tin không được trống khi ấn cập nhật hoặc tạo mới cho sản phẩm.
  + Tiền điều kiện: Quản trị viên đăng nhập với vai trò Admin.
  + Hậu điều kiện: Lưu thông tin các sản phẩm tương ứng vào bảng Product.
  + Điểm mở rộng: Không có.

1. Usecase quản lý tài khoản

* Mô tả vắn tắt: Use case này cho phép người quản trị thao tác và thay đổi thông tin của tất cả tài khoản người dùng.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi người quản trị ấn vào nút “Tới trang quản trị” ở trên thanh header, ngay sau đó trang quản lý hiện ra và người quản trị chọn vào mục “Quản lý tài khoản”.

Quản trị viên xem và thay đổi các thông tin của tài khoản bao gồm cập nhật tài khoản, xóa nhiều tài khoản, tạo tài khoản mới, lọc danh sách các tài khoản theo quyền.

Use case kết thúc.

* + Luồng rẽ nhánh: Không có.
  + Các yêu cầu đặc biệt: Người dùng không được xóa để trống 1 vài thông tin không được trống khi ấn cập nhật hoặc tạo mới cho tài khoản.
  + Tiền điều kiện: Quản trị viên đăng nhập với vai trò Admin.
  + Hậu điều kiện: Lưu thông tin các tài khoản tương ứng vào bảng User.
  + Điểm mở rộng: Không có.

1. Usecase quản lý đơn hàng

* Mô tả vắn tắt: Use case này cho phép người quản trị thao tác và thay đổi thông tin của tất cả tài khoản người dùng.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi người quản trị ấn vào nút “Tới trang quản trị” ở trên thanh header, ngay sau đó trang quản lý hiện ra và người quản trị chọn vào mục “Quản lý tài khoản”.

Quản trị viên xem và thay đổi các thông tin của tài khoản bao gồm cập nhật tài khoản, xóa nhiều tài khoản, tạo tài khoản mới, lọc danh sách các tài khoản theo quyền.

Use case kết thúc.

* + Luồng rẽ nhánh: Không có.
  + Các yêu cầu đặc biệt: Người dùng không được xóa để trống 1 vài thông tin không được trống khi ấn cập nhật hoặc tạo mới cho tài khoản.
  + Tiền điều kiện: Quản trị viên đăng nhập với vai trò Admin.
  + Hậu điều kiện: Lưu thông tin các tài khoản tương ứng vào bảng User.
  + Điểm mở rộng: Không có.

1. Usecase quản lý type của sản phẩm

* Mô tả vắn tắt: Use case này cho phép người quản trị thao tác và thay đổi thông tin type của tất cả sản phẩm.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi người quản trị ấn vào nút “Tới trang quản trị” ở trên thanh header, ngay sau đó trang quản lý hiện ra và người quản trị chọn vào mục “Quản lý type”.

Quản trị viên xem và thay đổi các thông tin của type bao gồm xóa nhiều type, tạo type mới.

Use case kết thúc.

* + Luồng rẽ nhánh: Không có.
  + Các yêu cầu đặc biệt: Không.
  + Tiền điều kiện: Quản trị viên đăng nhập với vai trò Admin.
  + Hậu điều kiện: Lưu thông tin các type tương ứng vào bảng Product.
  + Điểm mở rộng: Không có.

1. Usecase quản lý category của sản phẩm

* Mô tả vắn tắt: Use case này cho phép người quản trị thao tác và thay đổi thông tin category của tất cả sản phẩm.
* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu sau khi người quản trị ấn vào nút “Tới trang quản trị” ở trên thanh header, ngay sau đó trang quản lý hiện ra và người quản trị chọn vào mục “Quản lý category”.

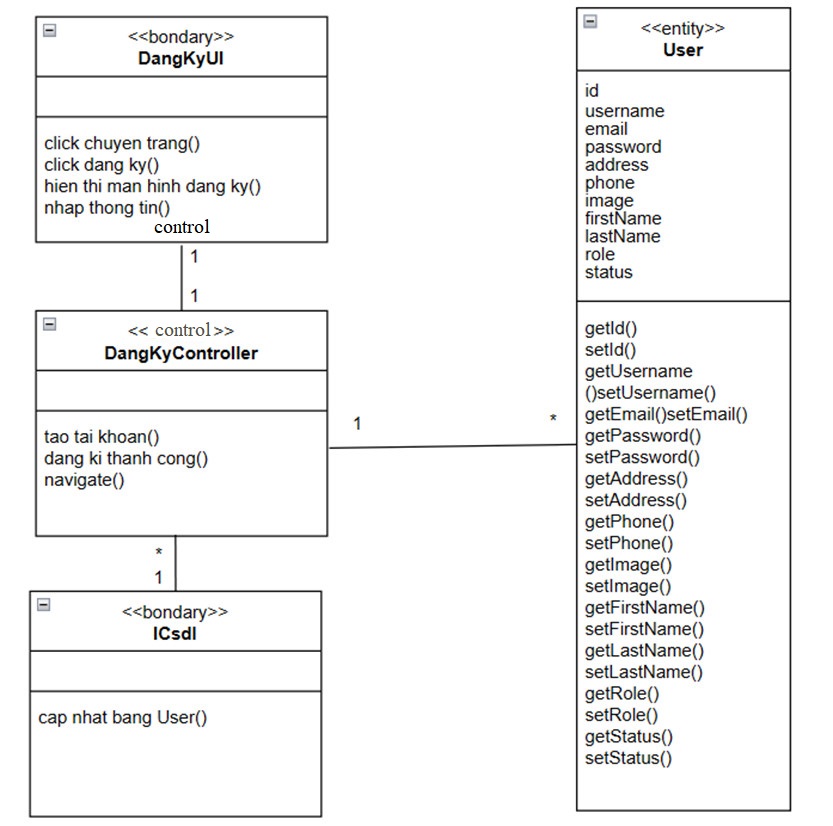
Quản trị viên xem và thay đổi các thông tin của category bao gồm xóa nhiều category, tạo category mới.

Use case kết thúc.

* + Luồng rẽ nhánh: Không có.
  + Các yêu cầu đặc biệt: Không.
  + Tiền điều kiện: Quản trị viên đăng nhập với vai trò Admin.
  + Hậu điều kiện: Lưu thông tin các category tương ứng vào bảng Product.
  + Điểm mở rộng: Không có.

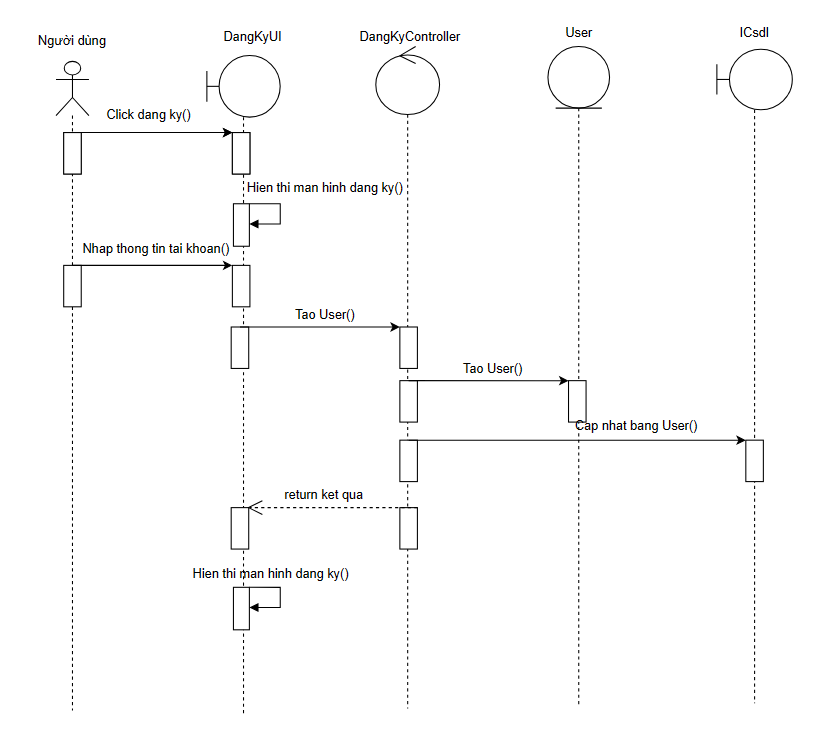
### Phân tích biểu đồ hoạt động usecase

* + 1. Biểu đồ hoạt động chức năng đăng kí tài khoản
* Biểu đồ lớp:



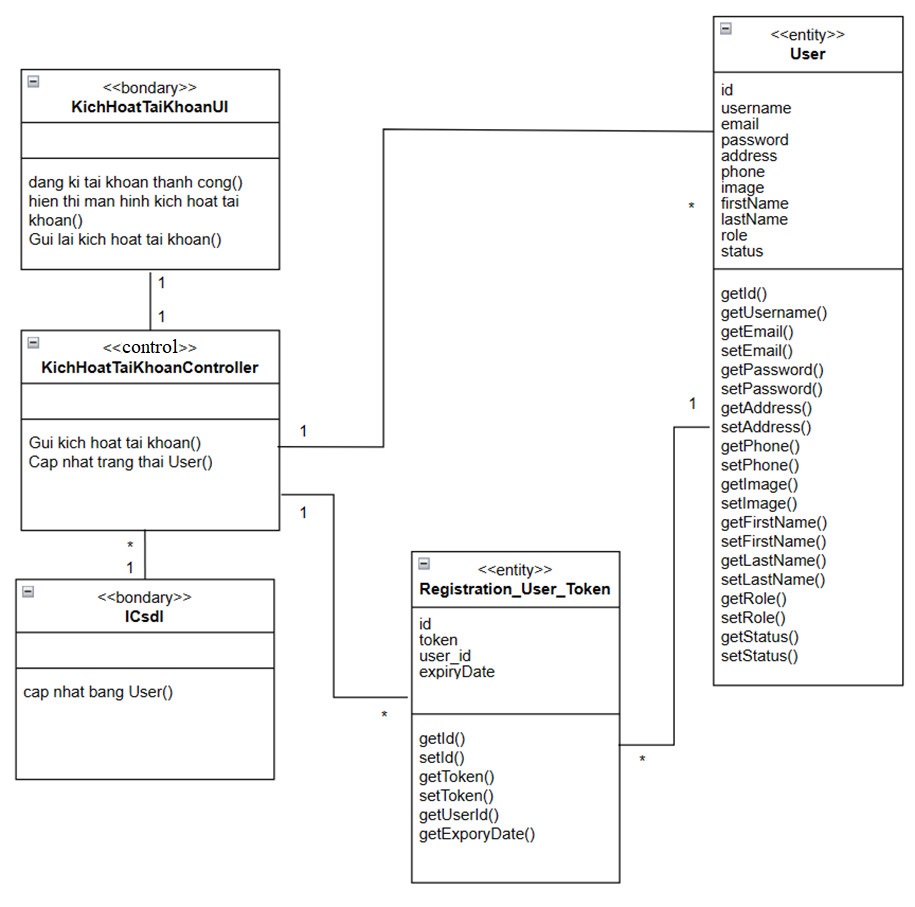
Hình 2. 2: Biểu đồ lớp đăng ký

* Biểu đồ trình tự:



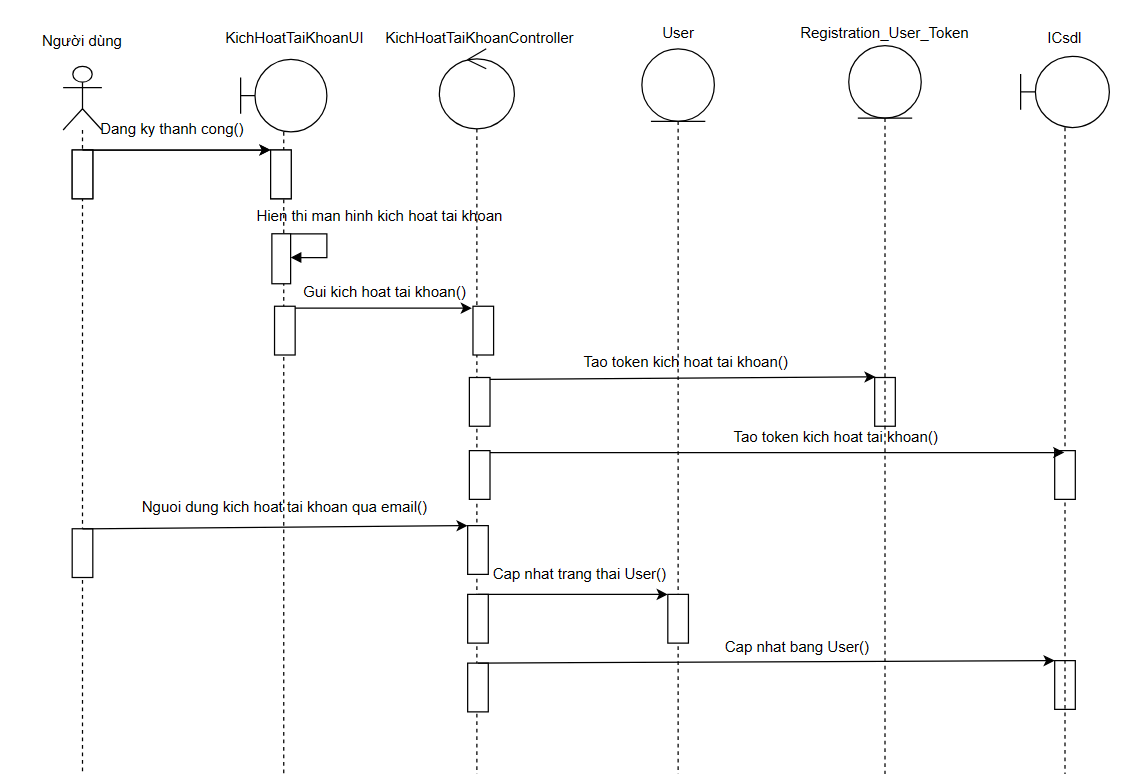
Hình 2. 3: Biểu đồ trình tự đăng ký

* + 1. Biểu đồ hoạt động chức năng kích hoạt tài khoản
* Biểu đồ lớp:



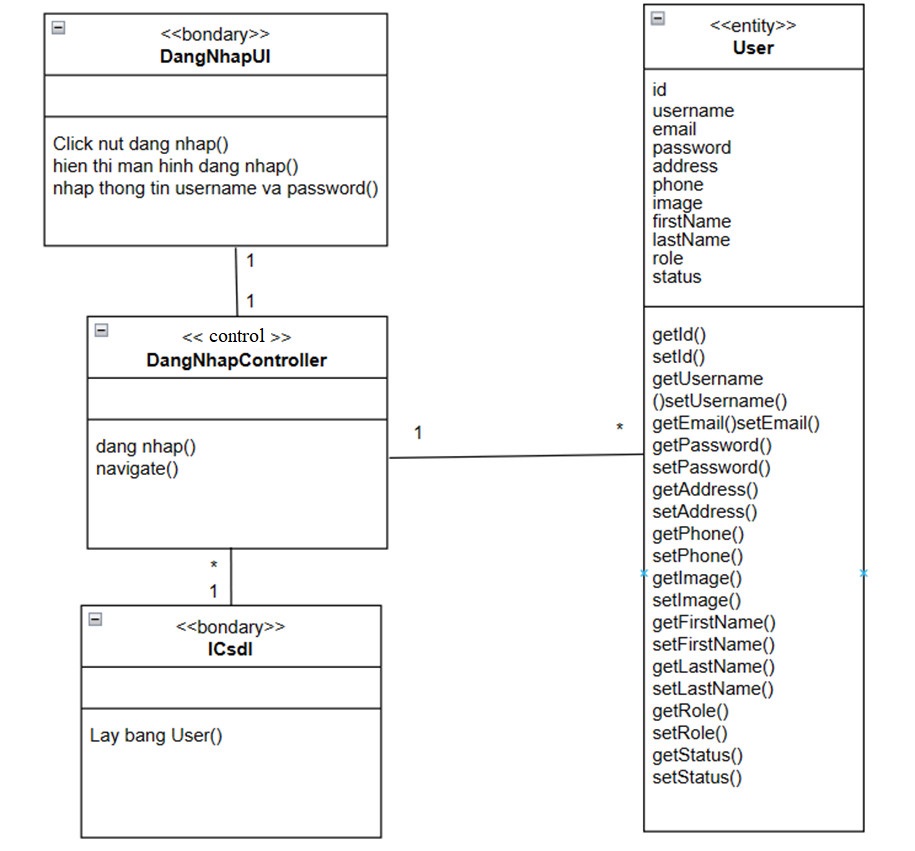
Hình 2. 4: Biểu đồ lớp kích hoạt tài khoản

* Biểu đồ trình tự:



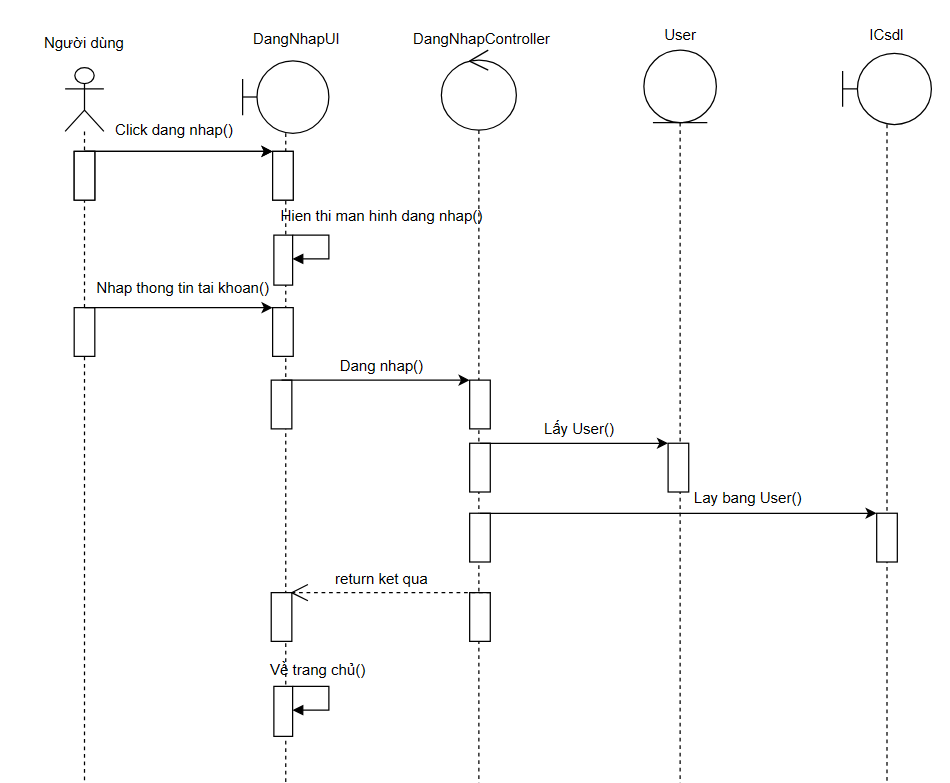
Hình 2. 5: Biểu đồ trình tự kích hoạt tài khoản

* + 1. Biểu đồ hoạt động chức năng đăng nhập
* Biểu đồ lớp:



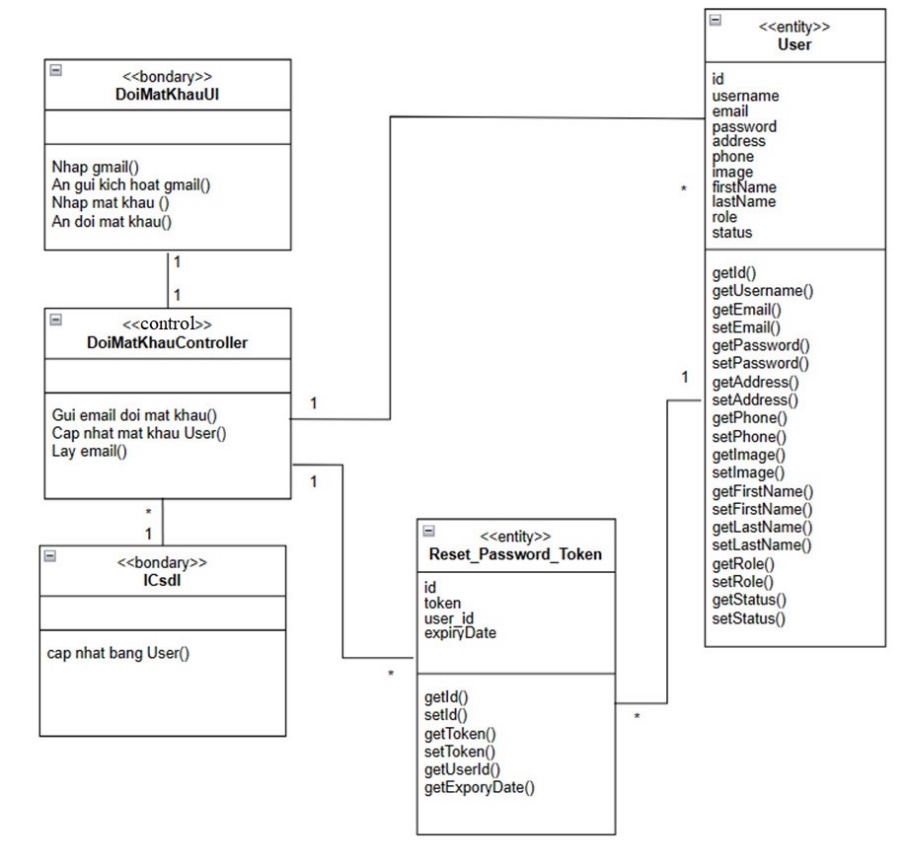
Hình 2. 6: Biểu đồ lớp chức năng đăng nhập

* Biểu đồ trình tự:



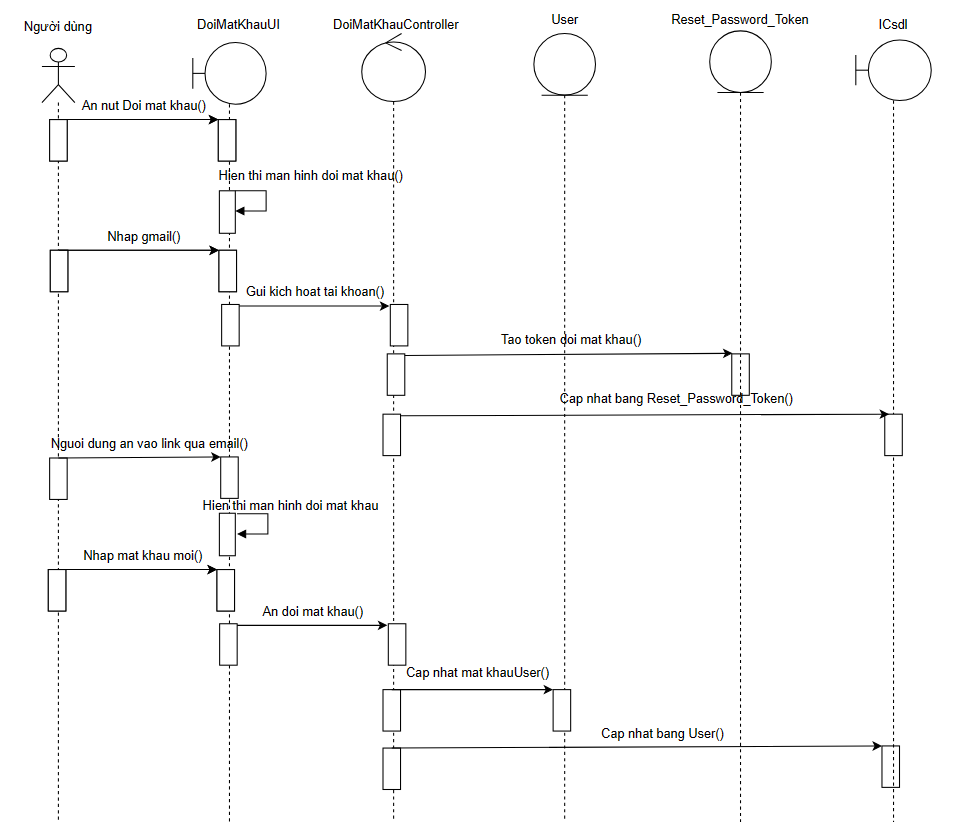
Hình 2. 7: Biểu đồ trình tự đăng nhập

* + 1. Biểu đồ hoạt động chức năng đổi mật khẩu
* Biểu đồ lớp:



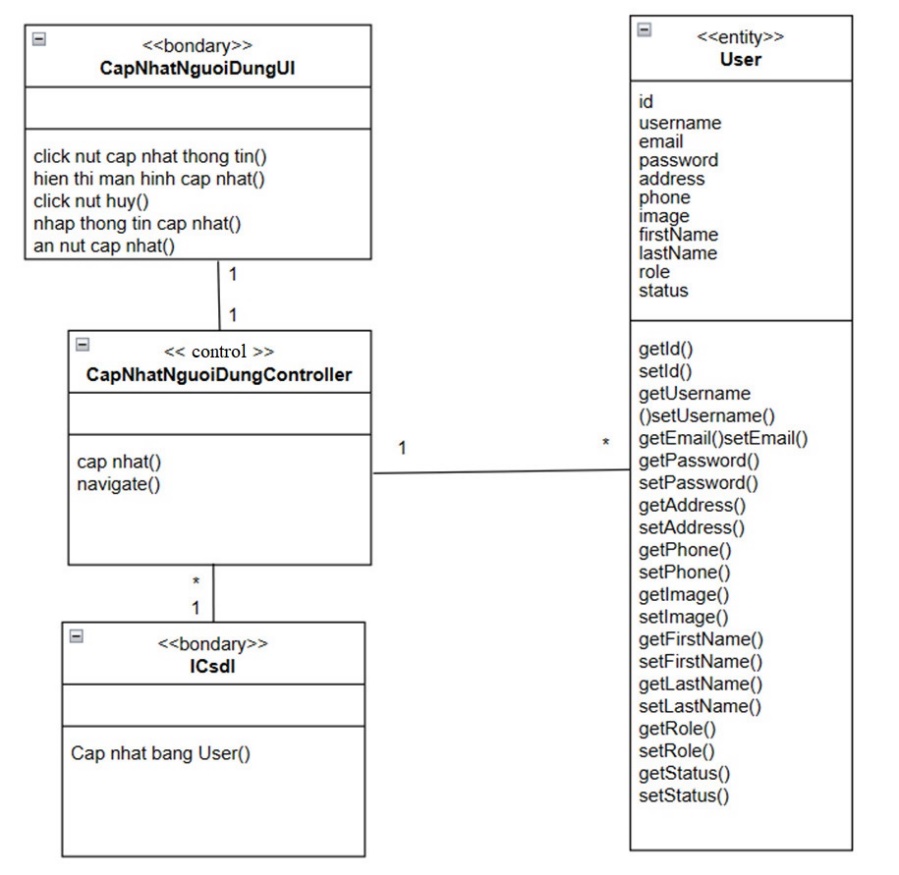
Hình 2. 8: Biểu đồ lớp đổi mật khẩu

* Biểu đồ trình tự:



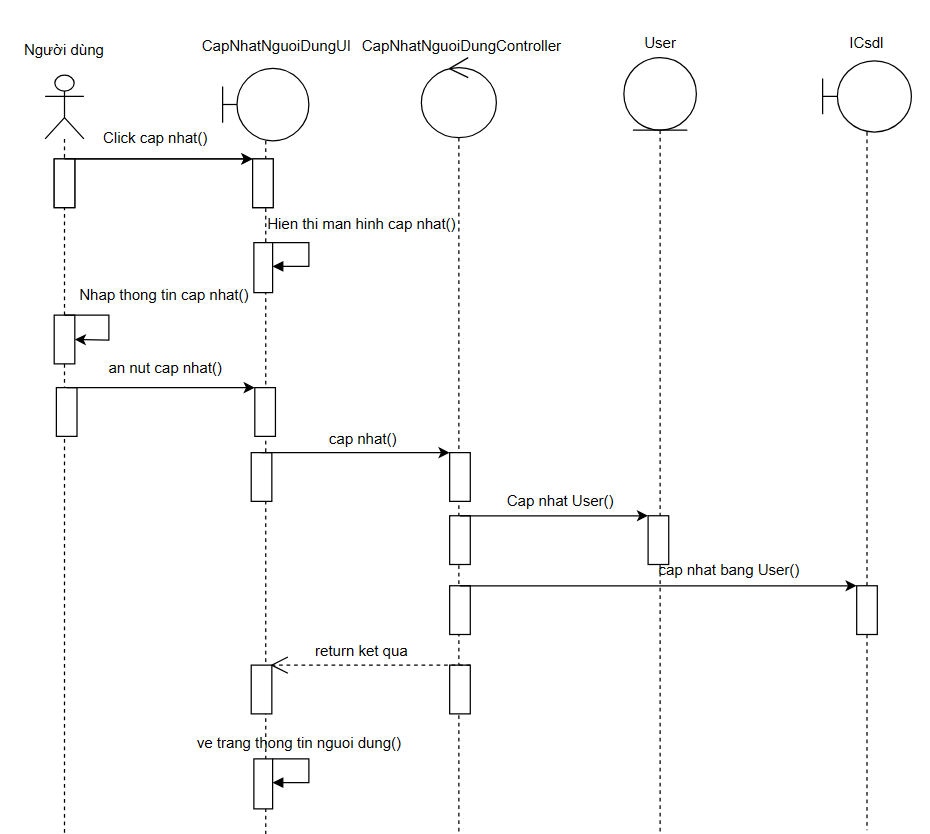
Hình 2. 9: Biểu đồ trình tự đổi mật khẩu

* + 1. Biểu đồ hoạt động chức năng cập nhật thông tin người dùng
* Biểu đồ lớp:



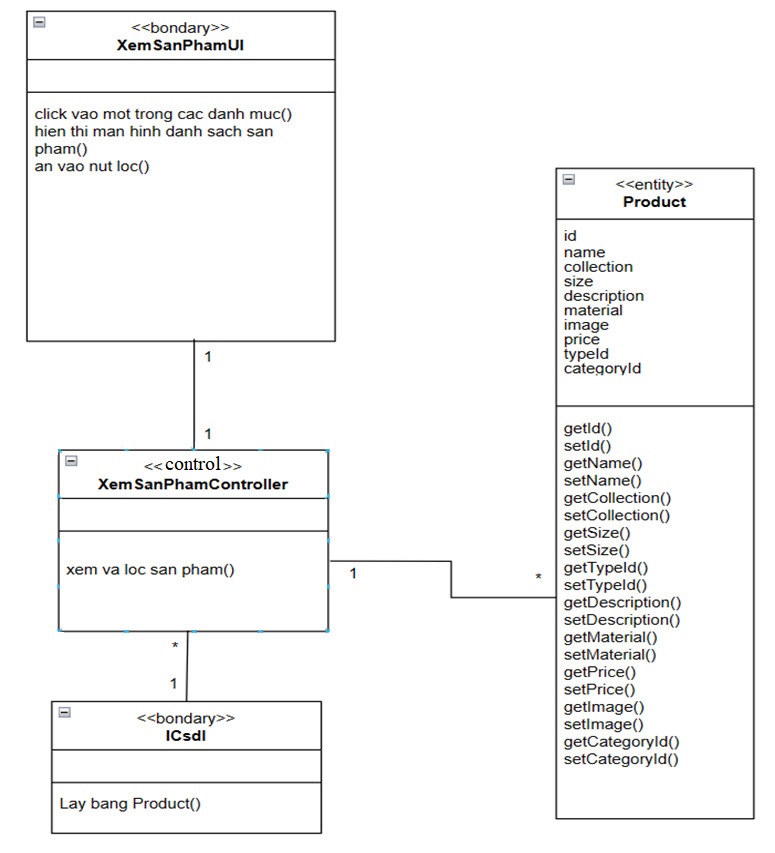
Hình 2. 10: Biểu đồ lớp cập nhật người dùng

* Biểu đồ trình tự:



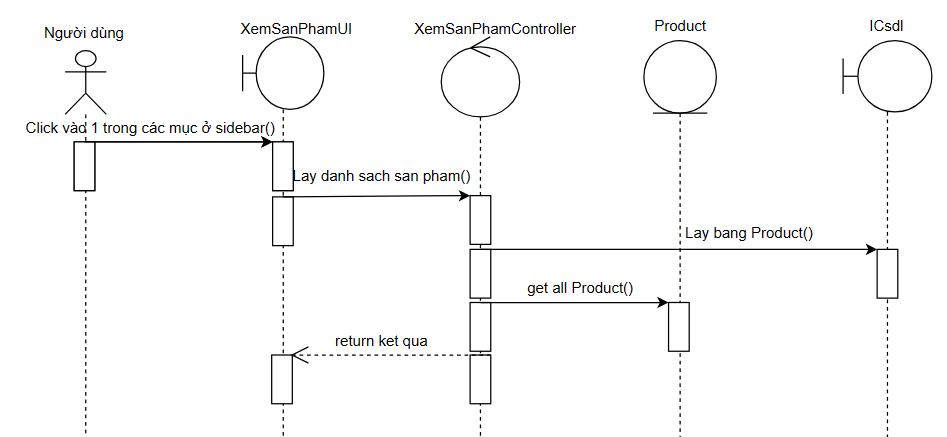
Hình 2. 11: Biểu đồ trình tự cập nhật thông tin người dùng

* + 1. Biểu đồ hoạt động chức năng xem sản phẩm
* Biểu đồ lớp:



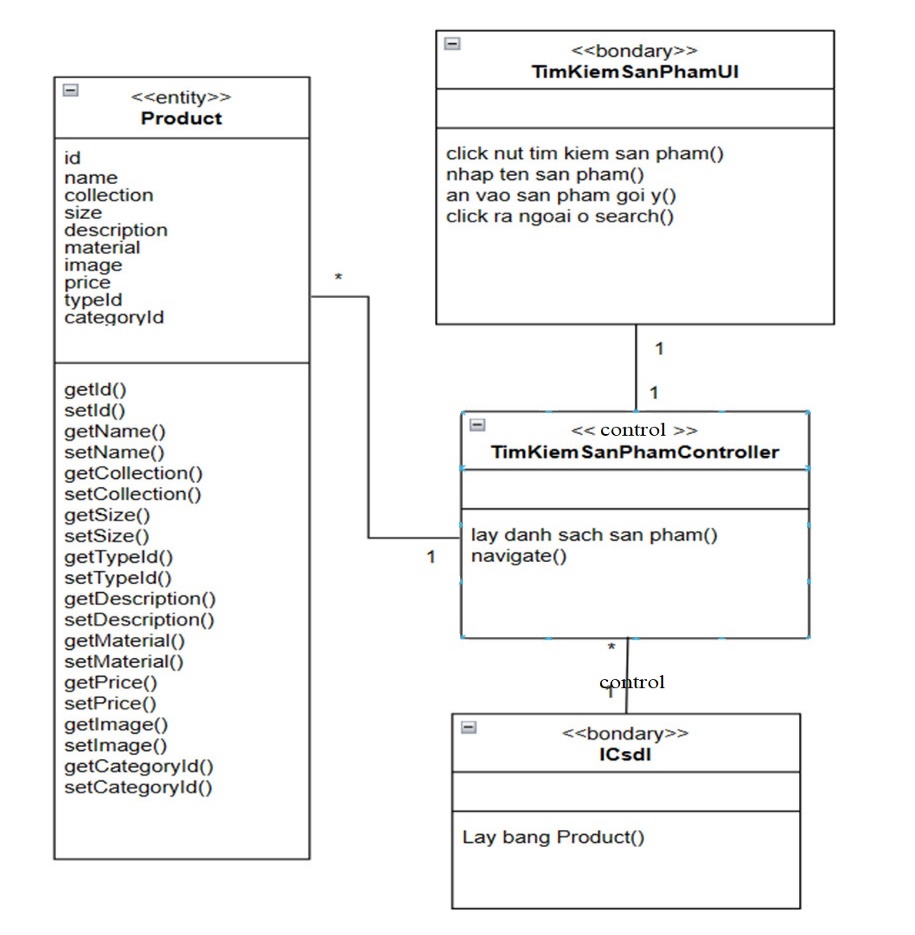
Hình 2. 12: Biểu đồ lớp xem sản phẩm

* Biểu đồ trình tự:



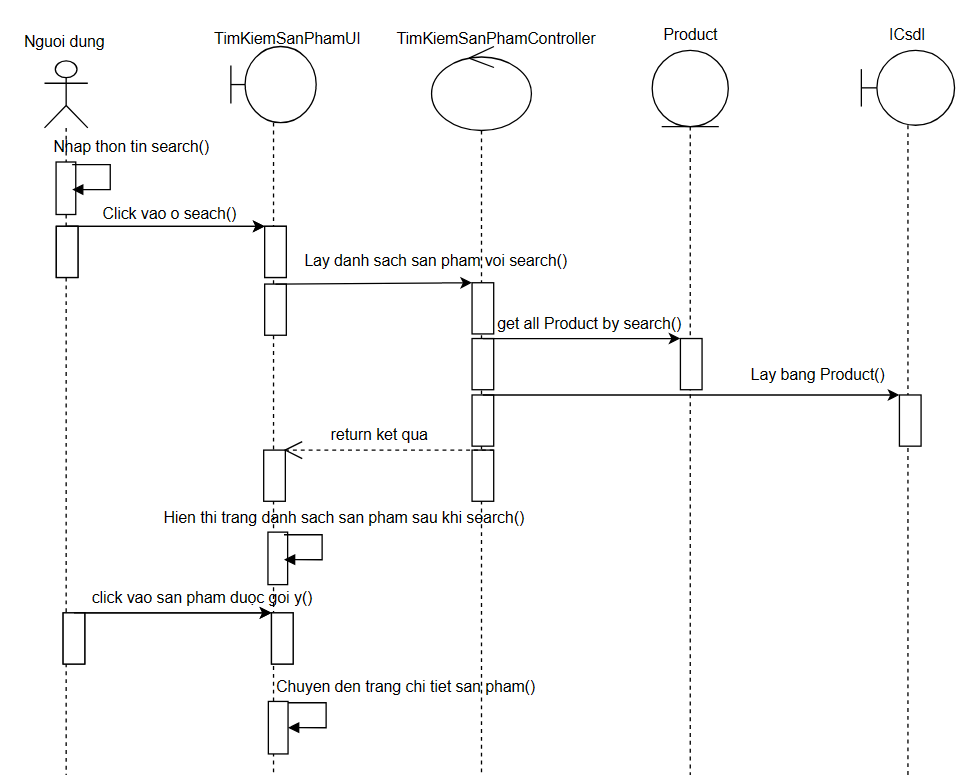
Hình 2. 13: Biểu đồ trình tự xem sản phẩm

* + 1. Biểu đồ hoạt động chức năng tìm kiếm sản phẩm
* Biểu đồ lớp:



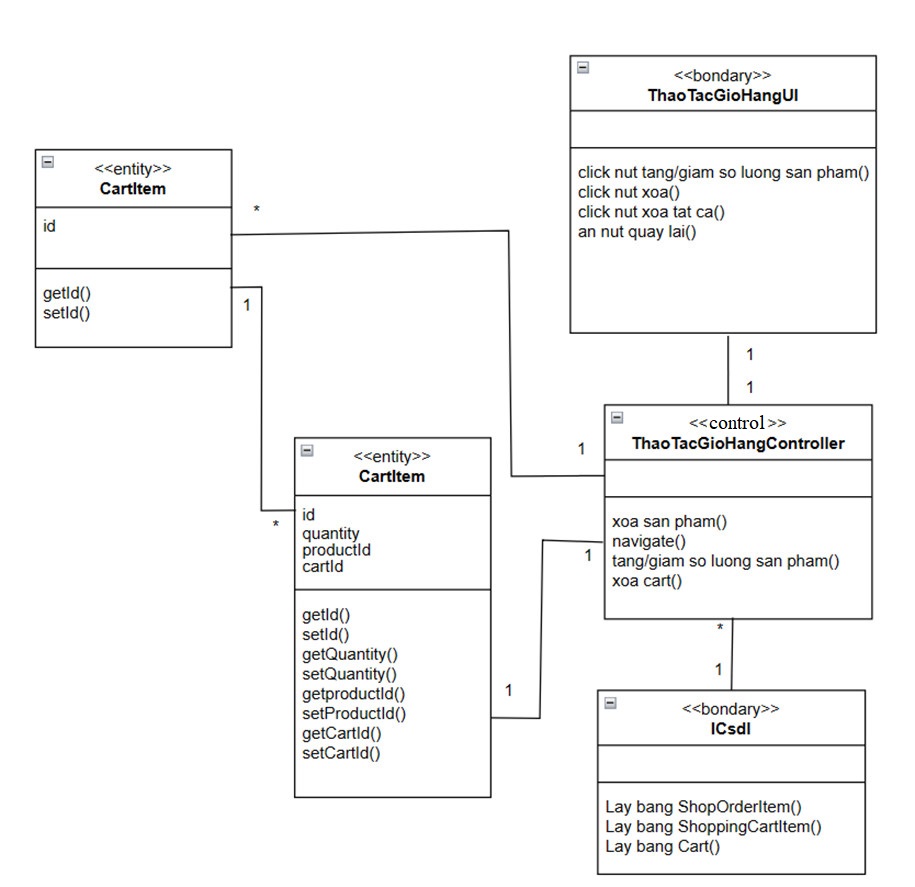
Hình 2. 14: Biểu đồ lớp tìm kiếm sản phẩm

Biểu đồ trình tự:



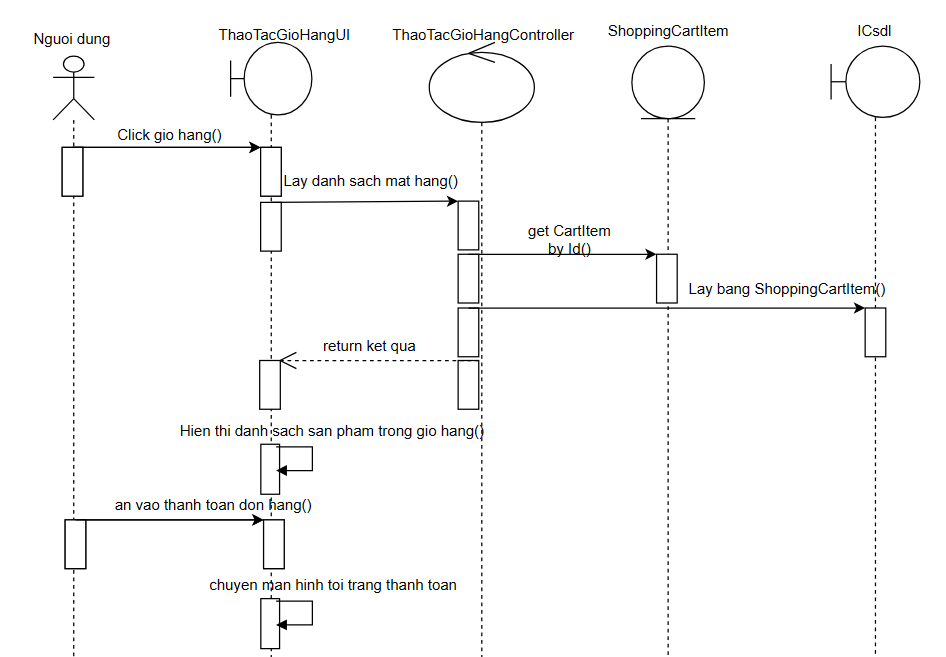
Hình 2. 15: Biểu đồ trình tự chức năng tìm kiếm sản phẩm

* + 1. Biểu đồ hoạt động chức năng thao tác giỏ hàng
* Biểu đồ lớp:



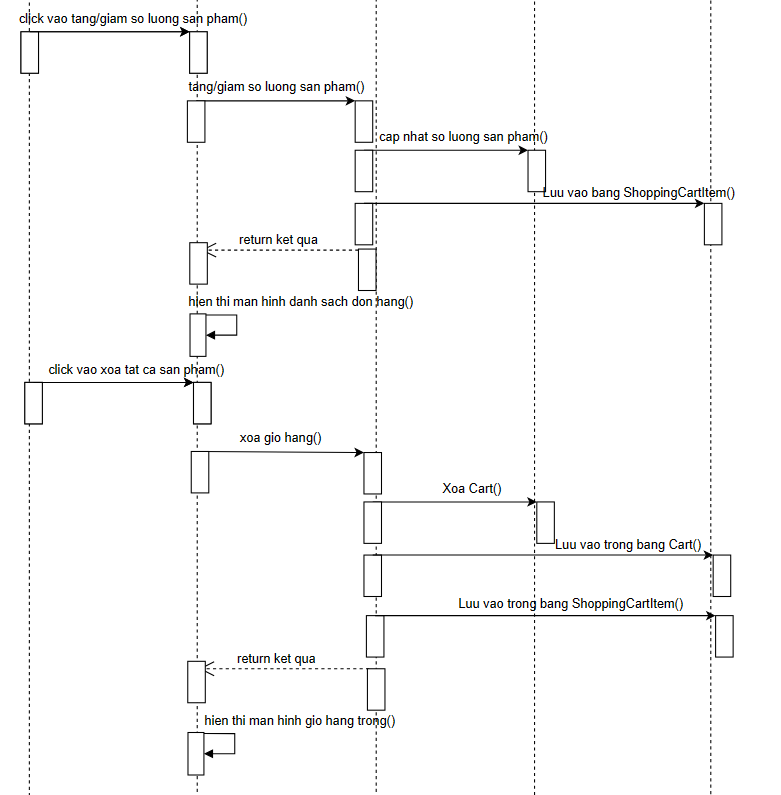
Hình 2. 16: Biểu đồ lớp chức năng thao tác giỏ hàng

* Biểu đồ trình tự 1:



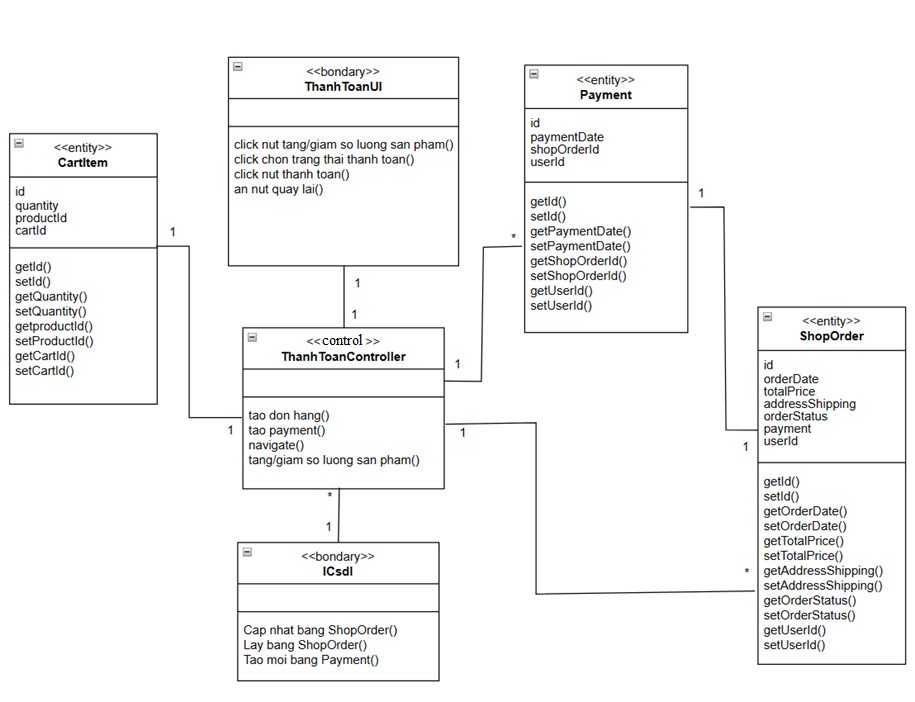
Hình 2. 17: Biểu đồ trình tự thao tác giỏ hàng

* Biều đồ trình tự 2:



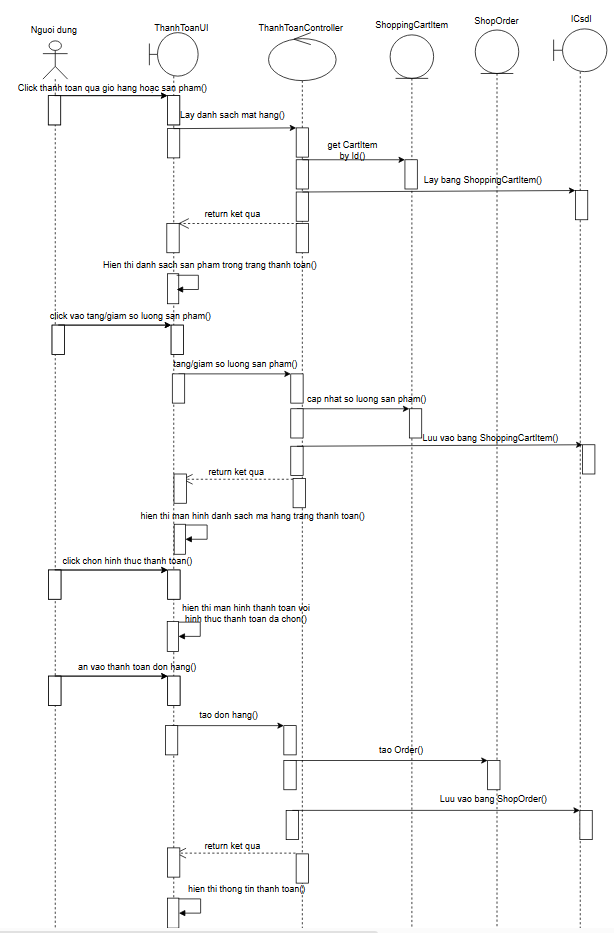
Hình 2. 18: Biểu đồ trình tự chức năng thao tác giỏ hàng

* + 1. Biểu đồ hoạt động chức năng thanh toán
* Biểu đồ lớp:

**

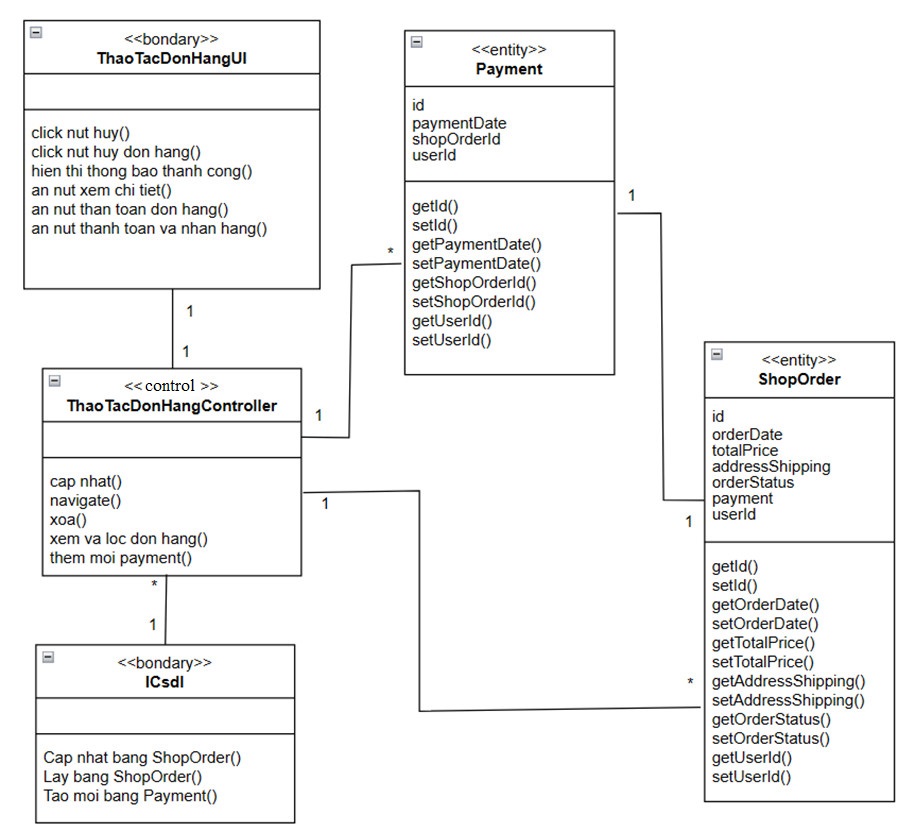
Hình 2. 19: Biểu đồ lớp chức năng thanh toán

* Biểu đồ trình tự:



Hình 2. 20: Biểu đồ trình tự chức năng thanh toán

* + 1. Biểu đồ hoạt động chức năng thao tác đơn hàng
* Biểu đồ lớp:



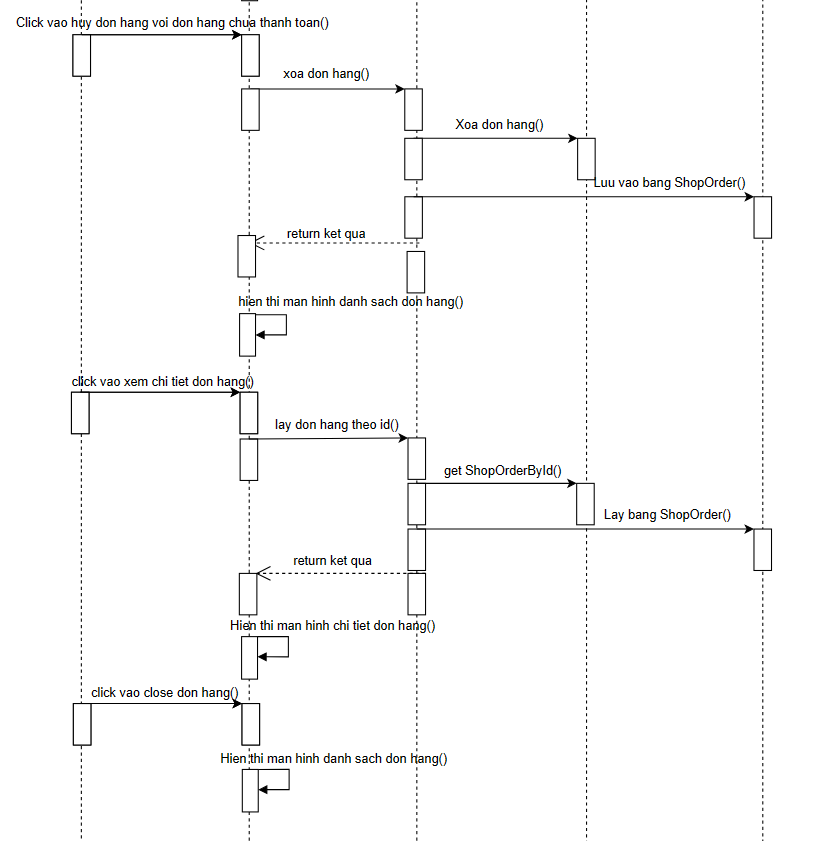
Hình 2. 21: Biểu đồ lớp chức năng thao tác đơn hàng

* Biểu đồ trình tự:



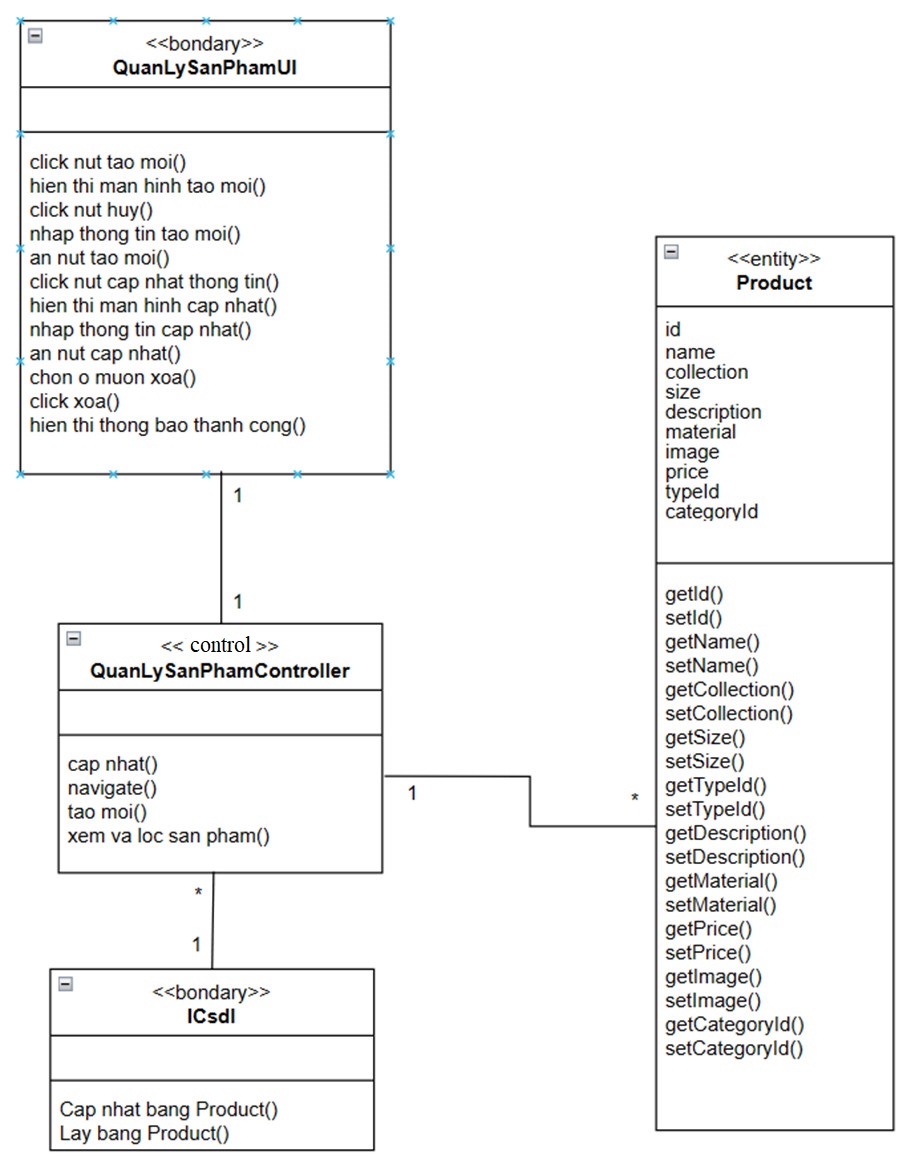
Hình 2. 22: Biểu đồ trình tự chức năng thao tác đơn hàng

* Biểu đồ trình tự:



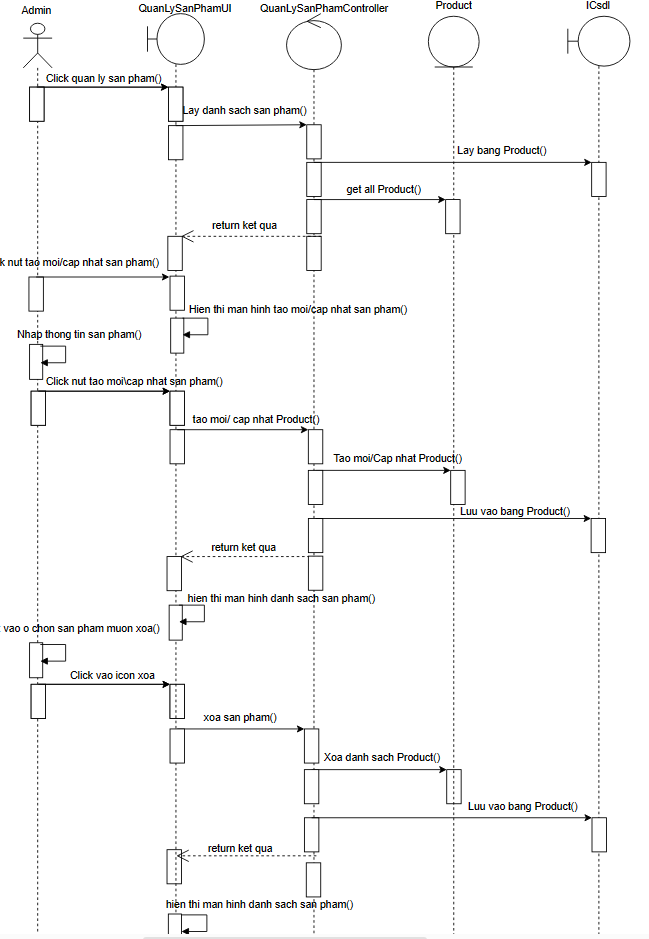
Hình 2. 23: Biểu đồ trình tự danh sách đơn hàng

* + 1. Biểu đồ hoạt động chức năng quản lý sản phẩm
* Biểu đồ lớp:



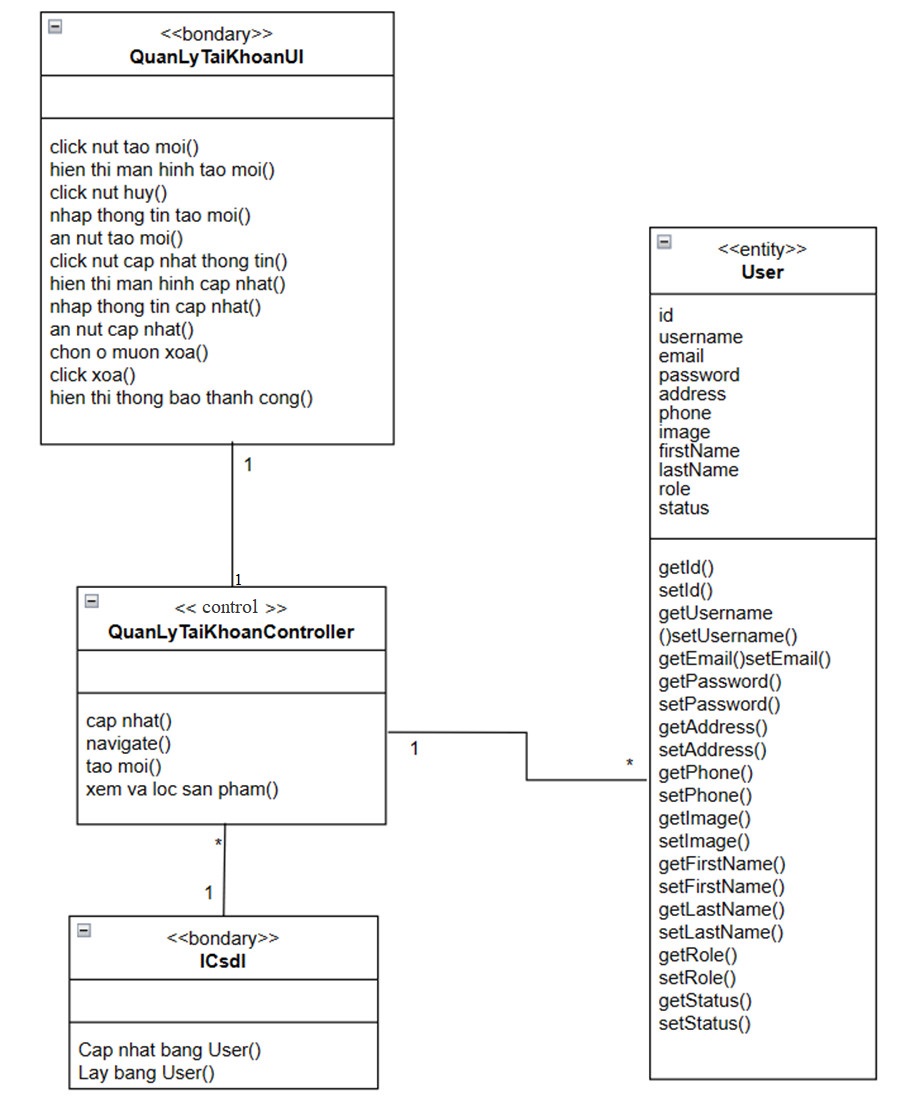
Hình 2. 24: Biểu đồ lớp chức năng quản lý sản phẩm

* Biểu đồ trình tự:



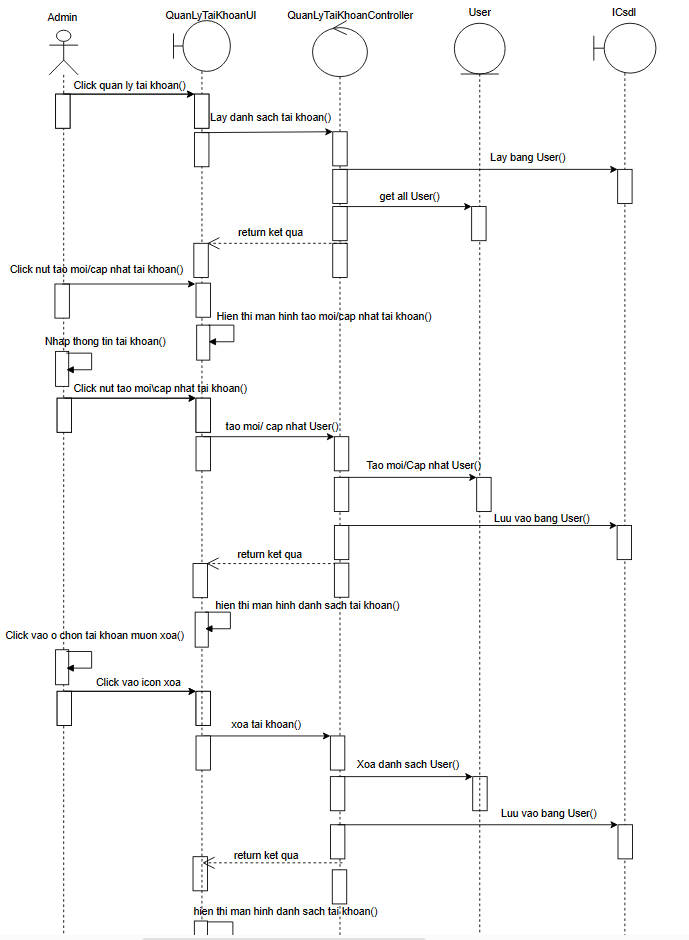
Hình 2. 25: Biểu đồ trình tự chức năng quản lý sản phẩm

* + 1. Biểu đồ hoạt động chức năng quản lý tài khoản
* Biểu đồ lớp:



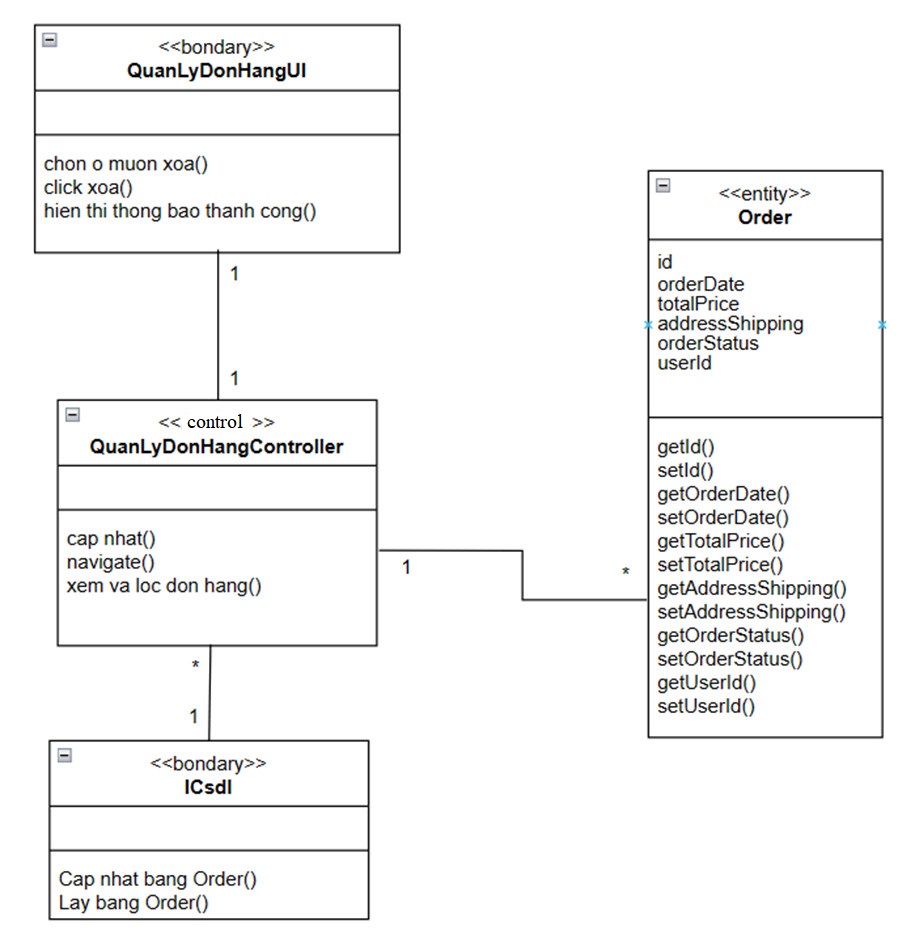
Hình 2. 26: Biểu đồ lớp quản lý tài khoản

* Biểu đồ trình tự:



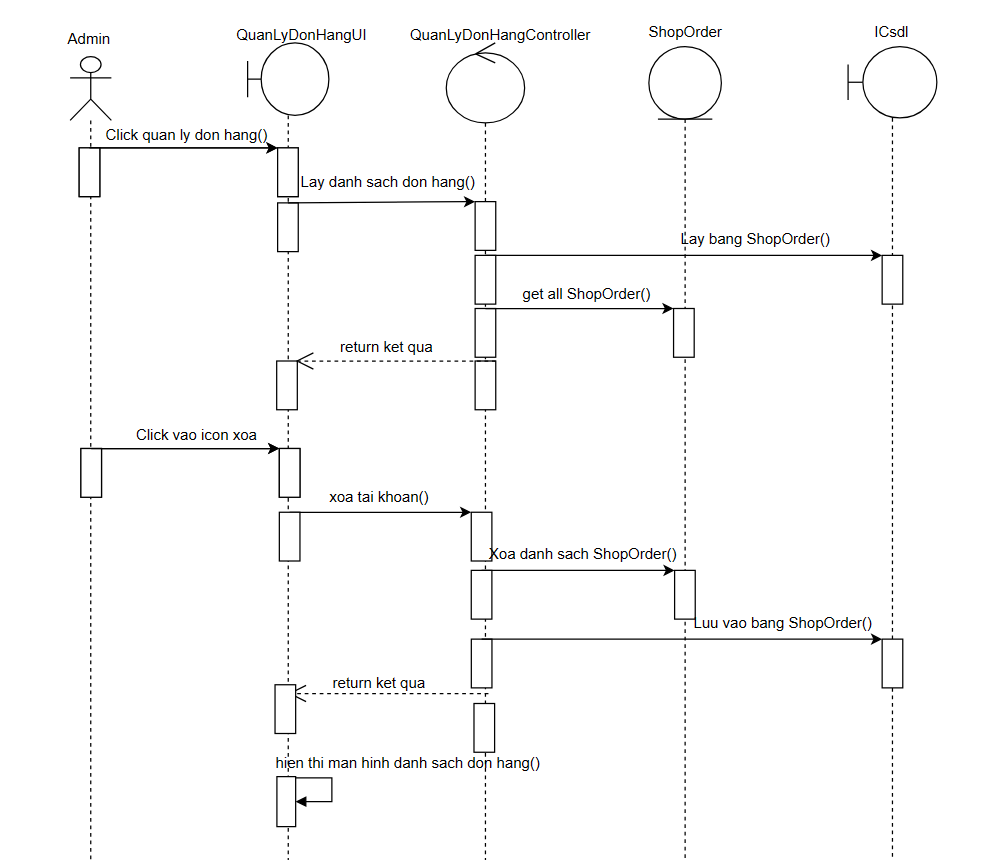
Hình 2. 27: Biểu đồ trình tự quản lý tài khoản

* + 1. Biểu đồ hoạt động chức năng quản lý đơn hàng
* Biểu đồ lớp:



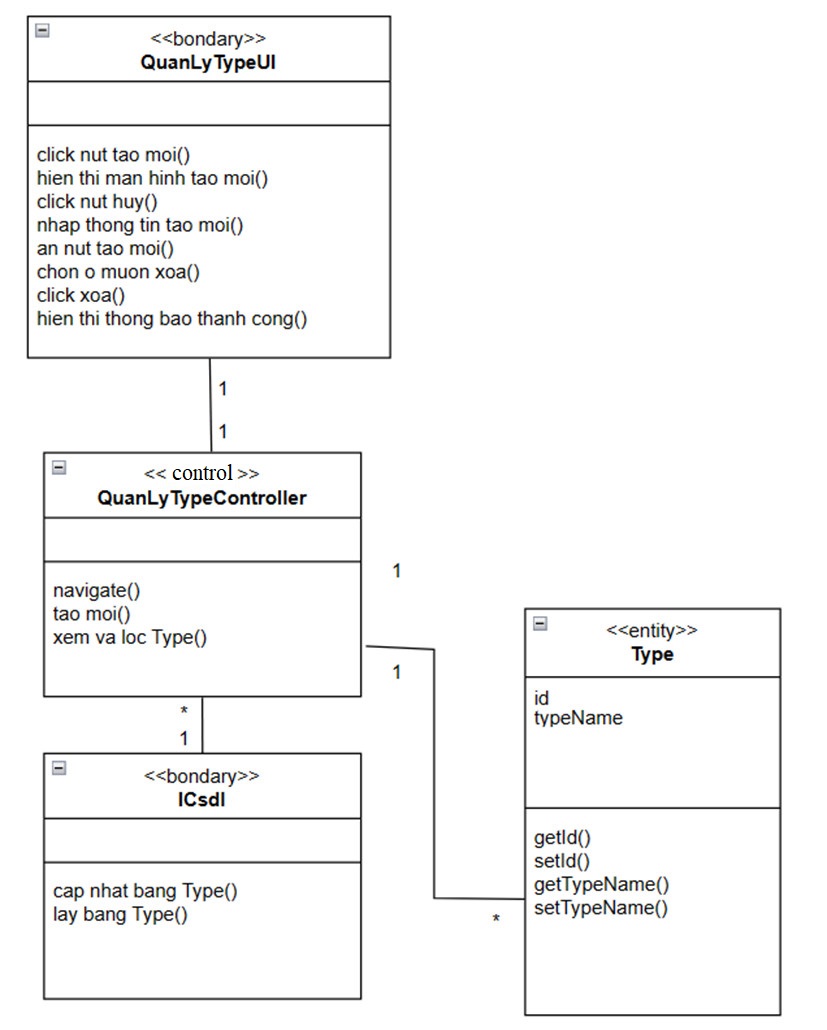
Hình 2. 28: Biểu đồ lớp quản lý đơn hàng

- Biểu đồ trình tự:



Hình 2. 29: Biểu đồ trình tự chức năng quản lý đơn hàng

* + 1. Biểu đồ hoạt động chức năng quản lý type của sản phẩm
* Biểu đồ lớp:



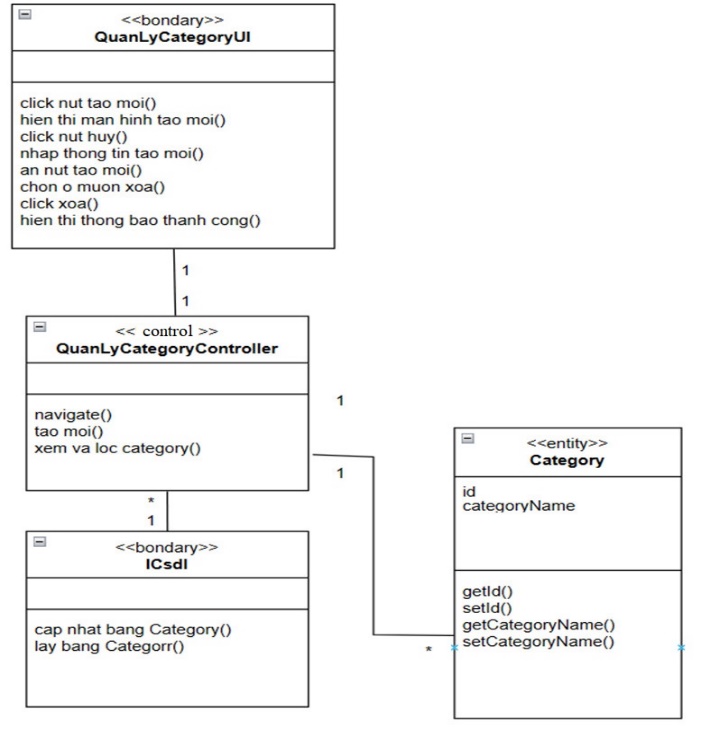
Hình 2. 30: Biểu đồ lớp quản lý type

* Biểu đồ trình tự:



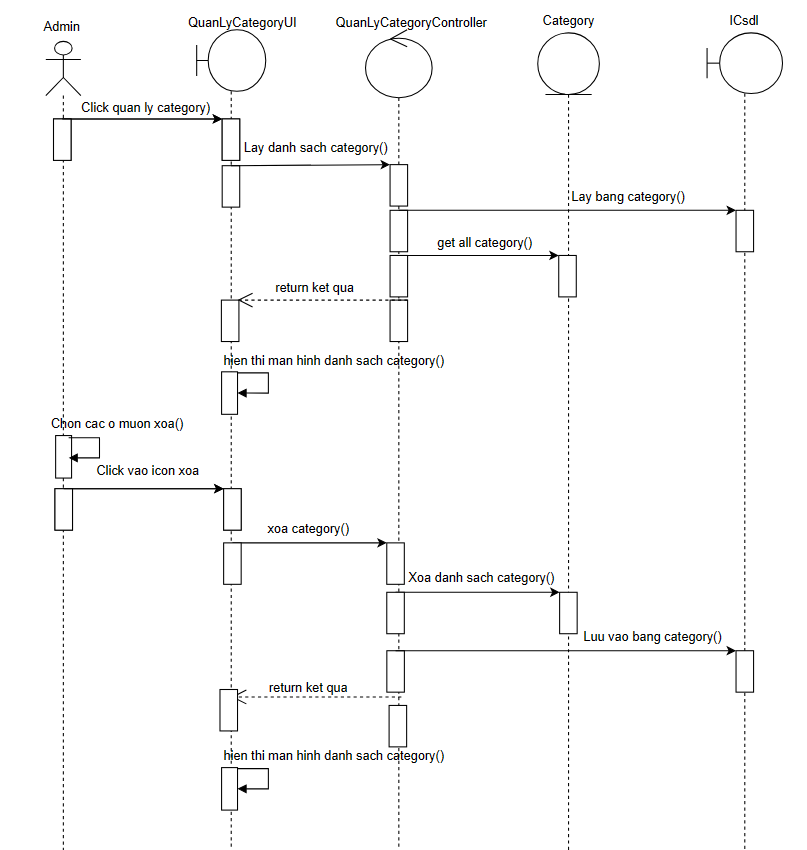
Hình 2. 31: Biểu đồ trình tự quản lý type

* + 1. Biểu đồ hoạt động chức năng quản lý category của sản phẩm
* Biểu đồ lớp:



Hình 2. 32: Biểu đồ lớp quản lý Category

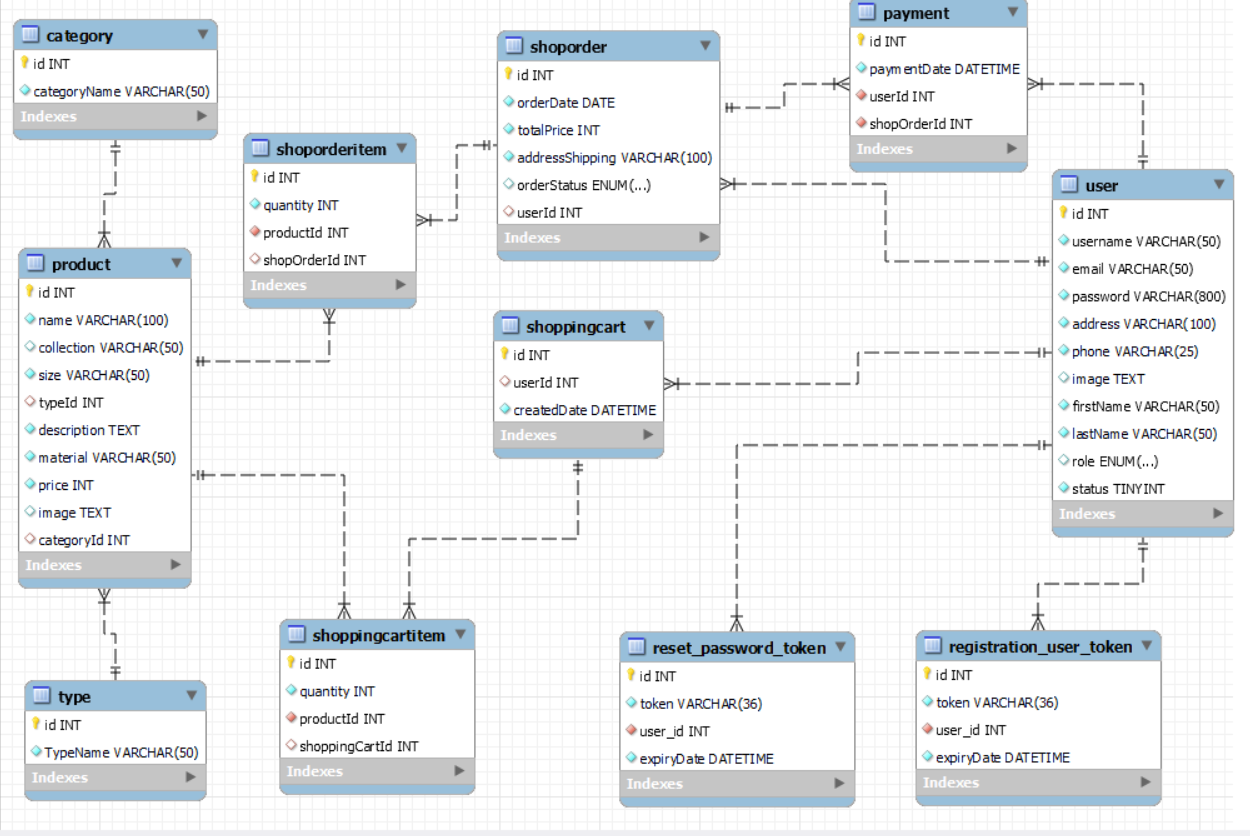
* Biểu đồ trình tự:



Hình 2. 33: Biểu đồ trình tự chức năng quản lý category

## Thiết kế cơ sở dữ liệu

### Biểu đồ Entity relationship diagram



Hình 2. 34: Biểu đồ Entity relationship diagram

Hình 3.1 Biểu đồ Entity relationship diagram

### Chi tiết các bảng trong cơ sở dữ liệu

1. Bảng User



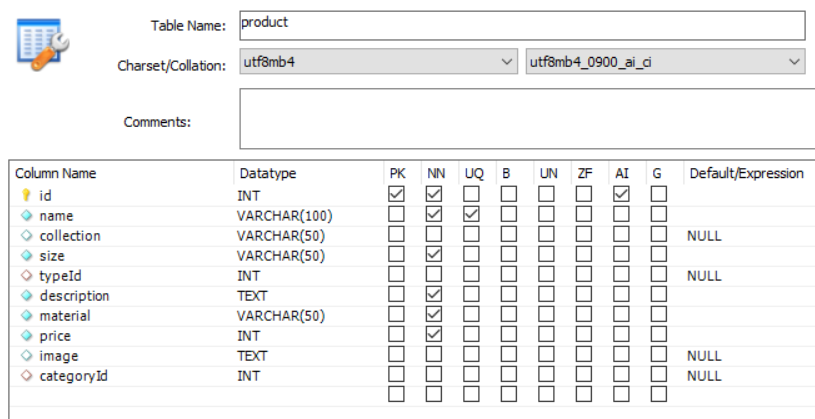
Hình 2. 35: Bảng User

Chi tiết bảng:

Bảng 2. 1: Mô tả bảng User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| username | varchar | 50 | not null, unique key | tên đăng nhập |
| email | varchar | 50 | not null, unique key | email |
| password | varchar | 800 | not null | mật khẩu |
| address | varchar | 100 | not null | địa chỉ |
| phone | varchar | 25 | not null | số điện thoại |
| image | text |  |  | hình ảnh |
| firstName | varchar | 50 | not null | Tên |
| lastName | varchar | 50 | not null | Họ |
| role | enum (Admin, User) |  | not null, default(‘User’) | quyền |
| status | TinyInt |  | not null, default (1) | trạng thái |

1. Bảng Product



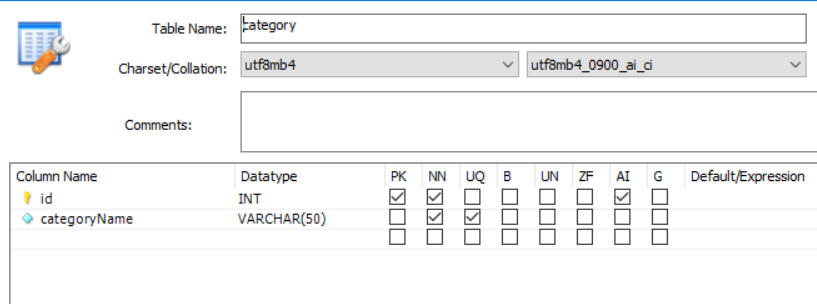
Hình 2. 36: Bảng Product

Chi tiết bảng:

Bảng 2. 2: Mô tả bảng Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| name | varchar | 100 | not null, unique key | tên sản phẩm |
| collection | varchar | 50 |  | bộ sưu tập |
| size | varchar | 800 | not null | kích thước |
| description | text |  | not null | mô tả |
| material | varchar | 50 | not null | chất liệu |
| price | text |  | not null | giá tiền |
| image | text |  |  | hình ảnh |
| typeId | int, foreign key |  | on delete set null | typeId |
| categoryId | int, foreign key |  | on delete set null | categoryId |

1. Bảng Category



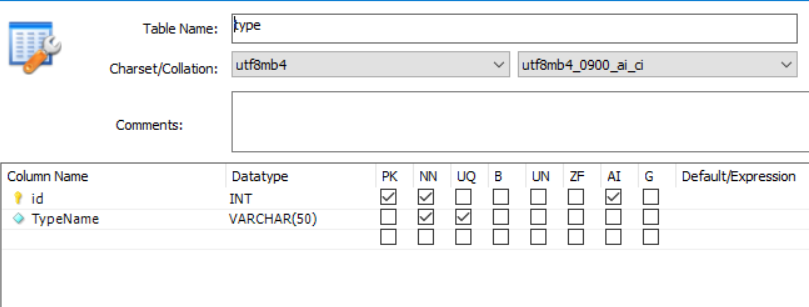
Hình 2. 37: Bảng Category

Chi tiết bảng:

Bảng 2. 3: Mô tả bảng Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| categoryName | varchar | 50 | not null, unique key | tên Category |

1. Bảng Type



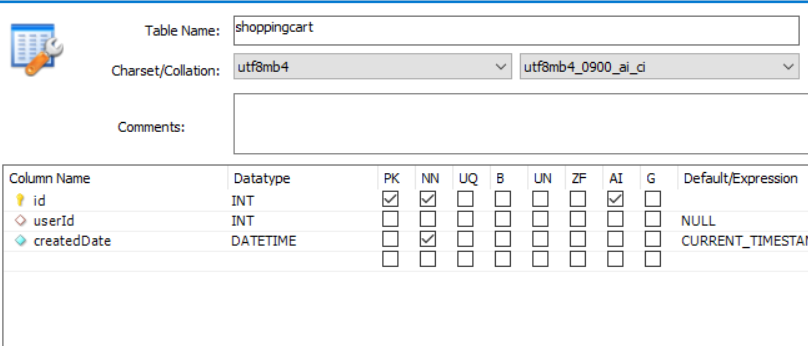
Hình 2. 38: Bảng Type

Chi tiết bảng:

Bảng 2. 4: Mô tả bảng Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| typeName | varchar | 50 | not null, unique key | tên type |

1. Bảng ShoppingCart



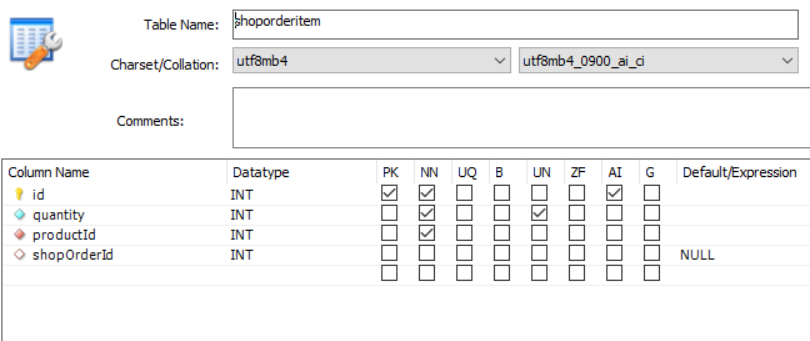
Hình 2. 39: Bảng ShoppingCart

Chi tiết bảng:

Bảng 2. 5: Mô tả bảng ShoppingCart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| userId | int |  | unique key | userId |
| createdDate | Datetime |  | not null, current timestamp | ngày tạo |

1. Bảng ShoppingCartItem



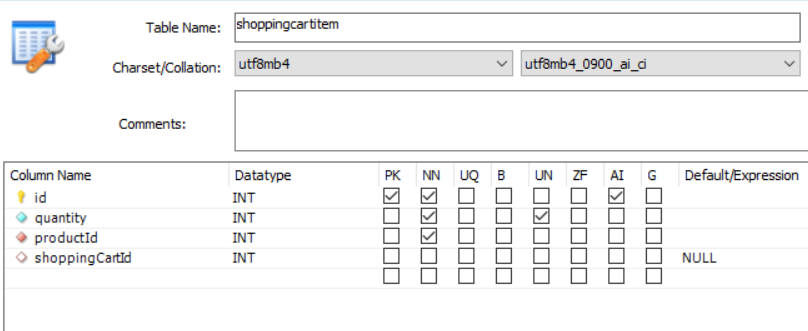
Hình 2. 40: Bảng ShoppingCartItem

Chi tiết bảng:

Bảng 2. 6: Mô tả bảng ShoppingCartItem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| quantity | int |  | not null, unsigned | số lượng |
| productId | int, foreign key |  | on delete cascase | productId |
| shopOrderId | int, foreign key |  | on delete cascase | shopOrderId |

1. Bảng ShopOrderItem



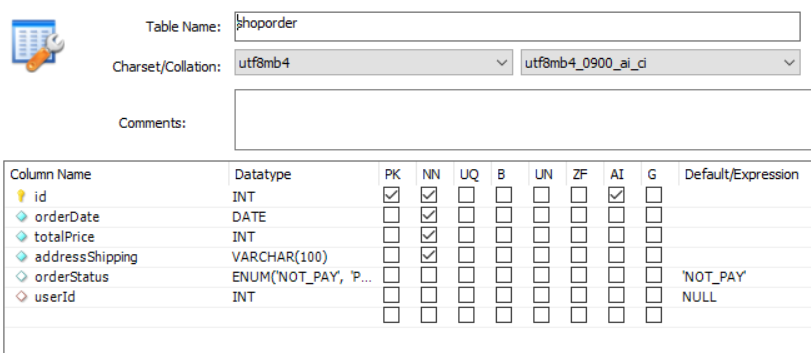
Hình 2. 41: Bảng ShopOrderItem

Chi tiết bảng:

Bảng 2. 7: Mô tả bảng ShopOrderItem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| quantity | int |  | not null, unsigned | số lượng |
| productId | int, foreign key |  | on delete cascase | productId |
| shoppingCartId | int, foreign key |  | on delete cascase | shoppingCartId |

1. Bảng ShopOrder



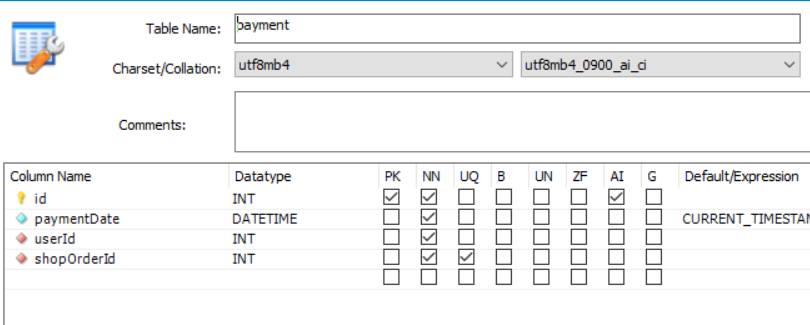
Hình 2. 42: Bảng ShopOrder

Chi tiết bảng:

Bảng 2. 8: Mô tả bảng ShopOrder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| orderDate | date |  | not null | Ngày đặt |
| totalPrice | int, foreign key |  | not null | Tổng tiền |
| addressShipping | varchar | 100 | not null | Địa chỉ |
| orderStatus | Enum(‘NOT\_PAY’, ‘PROCESSING’, ‘PAY’) |  | default(‘NOT\_PAY), not null | Trạng thái đơn hàng |
| userId | int, foreign key |  | on delete set null | userId |

1. Bảng Payment



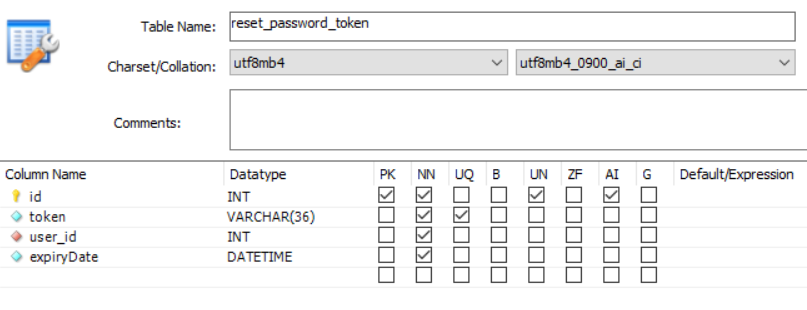
Hình 2. 43: Bảng Payment

Chi tiết bảng:

Bảng 2. 9: Mô tả bảng Payment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| paymentDate | datetime |  | not null | ngày thanh toán |
| userId | int, foreign key |  | on delete cascase | userId |
| shopOrderId | int, foreign key |  | on delete cascase | shopOrderId |

1. Bảng Reset\_password\_token



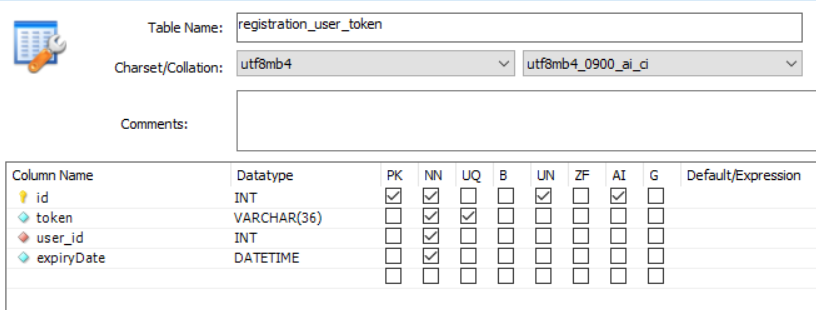
Hình 2. 44: Bảng Reset\_password\_token

Chi tiết bảng:

Bảng 2. 10: Mô tả bảng Reset\_password\_token

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| token | varchar | 36 | not null | token |
| userId | int, foreign key |  | on delete cascase | userId |
| expriryDate | datetime |  | not null | thời gian hết hạn |

1. Bang registration\_user\_token



Hình 2. 45: Bảng registration\_user\_token

Chi tiết bảng:

Bảng 2. 11: Mô tả bảng registration\_user\_token

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Kích thước** | **Ràng buộc** | **Mô tả** |
| id | int |  | not null, primarykey | id |
| token | varchar | 36 | not null | token |
| userId | int, foreign key |  | on delete cascase | userId |
| expriryDate | datetime |  | not null | thời gian hết hạn |

# CHƯƠNG 3: CÀI ĐẶT HỆ THỐNG VÀ MỘT SỐ KẾT QUẢ

## Công nghệ sử dụng

## Java

Hiện nay có rất nhiều công nghệ được sử dụng để phát triển và thiết kế website (PHP, ASP, ASP.NET, JSP...).

Java là một loại mã nguồn mở, hiện đang được khá nhiều người dùng lựa chọn để thiết kế website nhờ những ưu điểm nổi bật như sau.

**Khả năng bảo mật cao**

Java được đánh giá là có độ bảo mật cao hơn khá nhiều so với các ngôn ngữ lập trình khác. Do đó khi sử dụng mã nguồn này thì bạn sẽ không phải lo ngại nhiều về vấn đề bảo mật hay tình trạng bị hack thông tin trên web.

**Hoạt động mạnh mẽ**

Ưu điểm tiếp theo khiến cho Java được yêu thích đó chính là có khả năng mang lại một website hoạt động mạnh mẽ trên bất kỳ nền tảng nào. Đây là ưu điểm mà không phải bất kỳ ngôn ngữ nào cũng có thể đảm bảo được.

**Thông dịch**

Ngôn ngữ lập trình của Java vừa có khả năng biên dịch, vừa có khả năng thông dịch. Theo đó, chương trình nguồn viết bằng Java có đôi \*.java đầu tiên được biên dịch thành tập tin có đuôi \*.class, sau đó sẽ tiếp tục được thông dịch thành mã máy.

**Độc lập nền, khả năng chuyển**

Các chương trình được viết bởi ngôn ngữ Java có khả năng chạy trên nhiều máy tính, nhiều hệ điều hành khác nhau, chỉ cần ở đó có cài đặt máy ảo Java Virtual Machine. Tóm lại, ưu điểm cực lớn của Java chính là “Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere).

**Hướng đối tượng**

Hướng đối tượng trong ngôn ngữ lập trình Java khá giống với C++, tuy nhiên lại là ngôn ngữ lập trình hướng đối tượng hoàn toàn.

**Đa nhiệm - đa luồng (MultiTasking - Multithreading)**

Ngôn ngữ lập trình này hỗ trợ lập trình đa nhiệm, đa luồng nên cho phép chạy song song nhiều tiến trình, tiểu trình trong cùng một thời điểm và tạo tương tác với nhau.

**Hỗ trợ mạnh cho việc phát triển ứng dụng**

Một trong những ưu điểm không thể không kể đến của Java chính là cung cấp nhiều công cụ, thư viện lập trình, từ đó hỗ trợ phong phú cho việc phát triển nhiều loại hình ứng dụng khác nhau như:

*J2SE (Java 2 Standard Edition)*: hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-server.

*J2EE (Java 2 Enterprise Edition)*: hỗ trợ phát triển các ứng dụng thương mại.

*J2ME (Java 2 Micro Edition)*: hỗ trợ phát triển các ứng dụng trên các thiếtbị di động, không dây,..



Hình 3. 1: Ngôn ngữ Java

Chính vì những ưu điểm của việc phát triển ứng dụng web bằng Java như vậy nên em quyết định sử dụng công nghệ Java: Spring MVC, Spring JPA, SpringSecurity, Spring Boot để ứng dụng vào đề tài thực tập tốt nghiệp lần này.



Hình 3. 2: Framework Spring

## ReactJS

Ngoài các công nghệ Java mạnh mẽ đã được đề cập, ReactJS cũng là một công cụ quan trọng được sử dụng phổ biến trong phát triển giao diện người dùng (UI) cho các ứng dụng web hiện đại. Được phát triển bởi Facebook, ReactJS không chỉ giúp tạo ra giao diện người dùng mượt mà, tương tác mà còn cung cấp những ưu điểm nổi bật như sau:

**Thư viện JavaScript mạnh mẽ:**

ReactJS là một thư viện JavaScript được sử dụng để xây dựng giao diện người dùng. Thay vì làm việc với toàn bộ trang web, ReactJS tập trung vào các thành phần nhỏ (components) giúp tái sử dụng mã nguồn, cải thiện hiệu suất và dễ dàng bảo trì.

**Khả năng tương tác cao:**

ReactJS nổi bật với khả năng tạo ra các UI động, tương tác tốt với người dùng. Điều này có được nhờ vào hệ thống Virtual DOM (Document Object Model ảo) của React, giúp cập nhật và hiển thị giao diện một cách hiệu quả mà không cần phải tải lại toàn bộ trang web.

**Cộng đồng hỗ trợ rộng lớn:**

ReactJS có một cộng đồng phát triển lớn mạnh, điều này đồng nghĩa với việc có nhiều tài nguyên, thư viện và công cụ hỗ trợ, giúp giảm bớt khối lượng công việc cho các lập trình viên. Các thư viện như React Router, Redux, và nhiều công cụ khác hỗ trợ tích hợp dễ dàng với ReactJS để xây dựng các ứng dụng phức tạp.

**Tính linh hoạt và mở rộng:**

ReactJS không giới hạn chỉ trong việc phát triển web. Nhờ vào sự kết hợp với các framework như React Native, nó còn có thể được sử dụng để phát triển ứng dụng di động đa nền tảng, mang lại tính linh hoạt cao cho các dự án lập trình.

**Hỗ trợ tốt cho SEO:**

Một trong những điểm yếu của các ứng dụng web đơn trang (Single Page Applications - SPA) là việc tối ưu hóa công cụ tìm kiếm (SEO). Tuy nhiên, ReactJS có thể dễ dàng tích hợp với các công cụ render phía server (server-side rendering) hoặc sử dụng các kỹ thuật như Next.js để cải thiện SEO cho các ứng dụng web.

**Kết hợp với Java trong các dự án:**

ReactJS có thể dễ dàng kết hợp với các công nghệ Java như Spring Boot trong các dự án full-stack, nơi mà backend được xử lý bằng Java và frontend được xây dựng bằng ReactJS. Sự kết hợp này mang lại một giải pháp toàn diện cho việc phát triển các ứng dụng web hiện đại, đáp ứng nhu cầu của người dùng về cả hiệu suất và trải nghiệm.



Hình 3. 3: Thư viện ReactJS

Tóm lại, việc sử dụng ReactJS trong phát triển giao diện người dùng mang lại nhiều lợi ích cho các dự án web hiện đại, đặc biệt khi kết hợp với các công nghệ backend mạnh mẽ như Java. Trong dự án thực tập này, ReactJS có thể là một lựa chọn tối ưu để phát triển phần giao diện, tạo nên một ứng dụng web mạnh mẽ và thân thiện với người dùng.

## Môi trường đã sử dụng

### Eclipse

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 Eclipsechúng ta có thể mở rộng hơn mã nguồn bằng cách chèn thêm các plugins choproject (PDE- Plug-in Development Environment). Mặc dù Eclipse được viết bằngngôn 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 cácngôn 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 library hỗ trợ phát triển phần mềm. Chính vì những lý do đó nên nhóm em quyết định sử dụng Eclipse IDE với môi trường là Maven.



Hình 3. 4: Eclipse IDE

### Visual Studio Code

**Visual Studio Code** là một công cụ hỗ trợ lập trình mã nguồn mở được phát triển bởi Microsoft. Đây là một trong những môi trường phát triển tích hợp (IDE) được ưa chuộng nhất cho việc phát triển ứng dụng web, đặc biệt là khi làm việc với các framework và thư viện JavaScript như ReactJS.

**Visual Studio Code** nổi bật với các tính năng như:

* **Hỗ trợ JavaScript/TypeScript mạnh mẽ:** Visual Studio Code tích hợp sẵn nhiều công cụ và tính năng hỗ trợ cho việc phát triển với JavaScript và TypeScript, hai ngôn ngữ chính được sử dụng trong phát triển với ReactJS.
* **Extensions phong phú:** Visual Studio Code có một kho mở rộng (extension) khổng lồ, cho phép người dùng tùy biến môi trường lập trình theo nhu cầu. Các extension như ESLint, Prettier, và React snippets giúp tối ưu hóa quá trình phát triển ReactJS.
* **Hỗ trợ Git:** Tích hợp Git trong Visual Studio Code giúp quản lý mã nguồn dễ dàng, theo dõi các thay đổi và quản lý các nhánh trong dự án một cách hiệu quả.
* **Live Server:** Một trong những extension phổ biến nhất khi làm việc với ReactJS là Live Server, cho phép xem trước các thay đổi trên trang web ngay lập tức mà không cần phải làm mới thủ công.
* **Terminal tích hợp:** Visual Studio Code có tích hợp sẵn terminal, giúp lập trình viên có thể chạy các lệnh npm hoặc yarn, cài đặt các gói, và khởi động ứng dụng ReactJS mà không cần phải rời khỏi IDE.
* **Code Intelligence:** Với các tính năng như IntelliSense, tự động hoàn thành mã, và gợi ý mã, Visual Studio Code giúp tăng tốc quá trình lập trình, giảm thiểu lỗi cú pháp, và cải thiện năng suất làm việc.

**Node.js** và **npm** (Node Package Manager) là những công cụ quan trọng trong việc thiết lập môi trường phát triển cho ReactJS. Node.js cung cấp runtime để chạy JavaScript phía server, trong khi npm quản lý các gói (packages) và phụ thuộc cần thiết cho các dự án ReactJS.

**Create React App** là một công cụ phổ biến được sử dụng để tạo dự án ReactJS một cách nhanh chóng mà không cần phải cấu hình phức tạp. Create React App thiết lập sẵn các cấu hình cần thiết, giúp lập trình viên có thể tập trung vào việc phát triển mà không phải lo lắng về thiết lập môi trường.



Hình 3. 5: Visual Studio Code

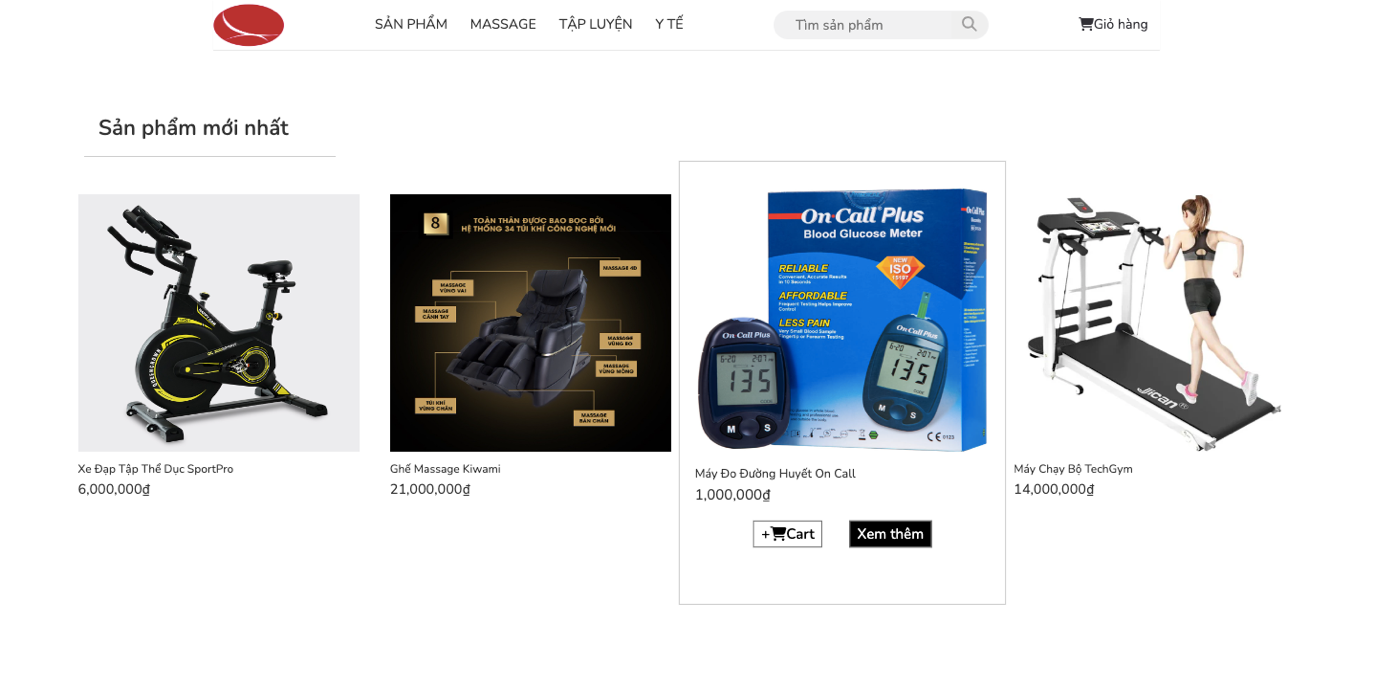
Chính vì những ưu điểm và tính linh hoạt trong việc phát triển ứng dụng web hiện đại, nhóm em đã quyết định sử dụng **Visual Studio Code** làm môi trường phát triển chính khi làm việc với ReactJS trong dự án này.

## Một số kết quả đạt được



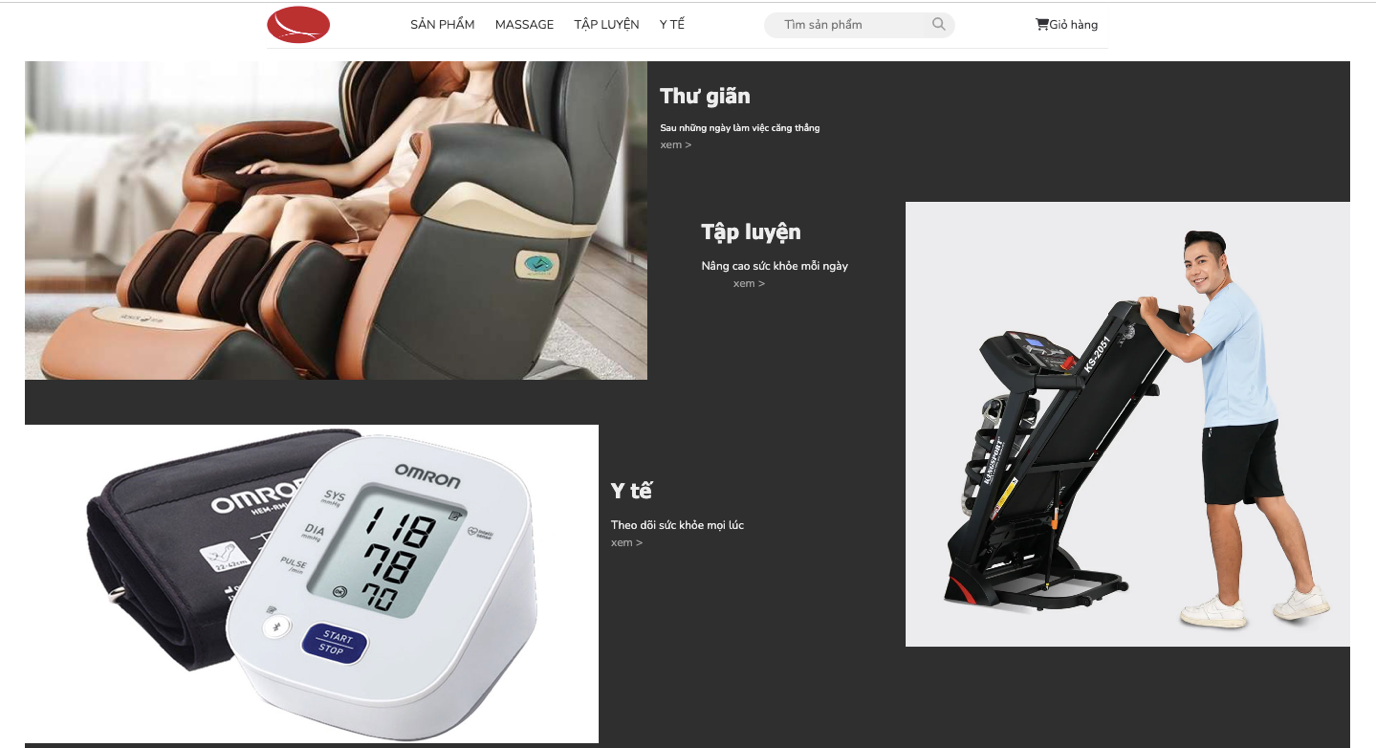
Hình 3. 6: Màn hình trang chủ - Thanh điều hướng và slide của website

Khi truy cập vào website bán thiết bị chăm sóc sức khỏe, người dùng vãn lai (Chưa đăng nhập) được điều hướng đến trang chủ của website. Khi được điều hướng, người dùng sẽ được nhìn thấy phần đầu trang chưa các thông tin của website như số điện thoại liên hệ, thanh điều hướng đến mục đăng ký - đăng nhập, tiếp đến là thanh điều hướng chính bao gồm logo của website, các nút điều hướng đi tới các sản phẩm của website, bên cạnh đó còn có mục tìm kiếm sản phẩm và giỏ hàng. Và bên dưới thanh điều hướng chính là phần quảng cáo các sản phẩm của website dưới dạng "slide" để hiển thị cho khách hàng các sản phẩm nổi bật nhất tháng hoặc năm.



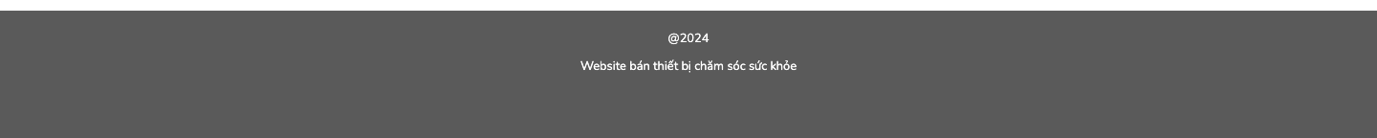
Hình 3. 7: Màn hình trang chủ - Sản phẩm mới nhất

Tại phần sản phẩm mới nhất của website bán thiết bị chăm sóc sức khỏe, người dùng chưa đăng nhập, hoặc đã có tài khoản và đăng nhập sẽ có thể thấy các sản phẩm được cập nhật mới nhất của cửa hàng.



Hình 3. 8: Màn hình trang chủ - Điều hướng có hình ảnh minh họa

Tại phần này, là nơi đưa ra các hình ảnh minh họa của các loại sản phẩm, từ các mục này, người dùng có thể tiếp cận dễ dàng hơn tới các sản phẩm mong muốn khi đến với website.



Hình 3. 9: Màn hình trang chủ - footer

Và cuối cùng, phần chân trang của website, nó sẽ cung cấp một vài thông tin về