**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

-----🙞🙜🕮🙞🙜-----

Logo

Description automatically generated

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

**XÂY DỰNG WEBSITE BÁN SÁCH COMIC STORE**

**Giảng viên hướng dẫn: TS. Nguyễn Bá Nghiễn**

**Sinh viên: Phan Quang Huy**

**Mã sinh viên: 2019603750**

**Lớp: CNTT03 – K14**

**Hà Nội, Năm 2023**

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

-----🙞🙜🕮🙞🙜-----

Logo

Description automatically generated

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

**XÂY DỰNG WEBSITE BÁN SÁCH COMIC STORE**

**Giảng viên hướng dẫn: TS. Nguyễn Bá Nghiễn**

**Sinh viên: Phan Quang Huy**

**Mã sinh viên: 2019603750**

**Lớp: CNTT03 – K14**

**Hà Nội, Năm 2023**

**MỤC LỤC**

[DANH MỤC TỪ VIẾT TẮT 1](#_Toc147319539)

[LỜI CẢM ƠN 2](#_Toc147319540)

[MỞ ĐẦU 3](#_Toc147319541)

[CHƯƠNG 1. TỔNG QUAN 4](#_Toc147319542)

[1.1. Tổng quan về đề tài 4](#_Toc147319543)

[1.1.1. Tên đề tài 4](#_Toc147319544)

[1.1.2. Lý do chọn đề tài 4](#_Toc147319545)

[1.2. Tổng quan về lập trình Web 5](#_Toc147319546)

[1.2.1. Web tĩnh 5](#_Toc147319547)

[1.2.2. Web động 5](#_Toc147319548)

[1.2.3. Khác nhau giữa web tĩnh và web động 6](#_Toc147319549)

[1.3. Công nghệ sử dụng trong dự án 7](#_Toc147319550)

[1.3.1. Mô hình MVC 7](#_Toc147319551)

[1.3.2. MERN STACK 10](#_Toc147319552)

[1.3.3. Nền tảng Nodejs 10](#_Toc147319553)

[1.3.4. Thư viện Reactjs 12](#_Toc147319554)

[1.3.5. Framework Expressjs 13](#_Toc147319555)

[1.3.6. Dịch vụ lưu trữ dữ liệu Mongodb 15](#_Toc147319556)

[1.3.7. Thư viện thiết kế Ant Design 16](#_Toc147319557)

[1.4. Các kiến thức cần nắm được trong quá trình thực hiện đồ án 17](#_Toc147319558)

[CHƯƠNG 2. THIẾT KẾ HỆ THỐNG 18](#_Toc147319559)

[1.1. Khảo sát hệ thống 18](#_Toc147319560)

[1.1.1. Đối tượng khảo sát 18](#_Toc147319561)

[1.1.2. Quy trình tác nghiệp 18](#_Toc147319562)

[1.1.3. Quy trình thực hiện mua hàng của khách hàng 18](#_Toc147319563)

[1.1.4. Quy trình thực hiện quản trị hệ thống của nhân viên quản trị 20](#_Toc147319564)

[1.1.5. Các thông tin vào ra của hệ thống 21](#_Toc147319565)

[1.1.6. Giới thiệu chung về website bán sách Comic Store 22](#_Toc147319566)

[1.1.7. Yêu cầu hệ thống 22](#_Toc147319567)

[1.2. Phân tích thiết kế hệ thống 24](#_Toc147319568)

[1.2.1. Người quản trị 24](#_Toc147319569)

[1.2.2. Khách hàng 25](#_Toc147319570)

[1.2.3. Mô tả use case 26](#_Toc147319571)

[1.3. Thiết kế giao diện và cơ sở dữ liệu 61](#_Toc147319572)

[1.3.1. Thiết kế CSDL 61](#_Toc147319573)

[1.3.2. Xây dựng giao diện 63](#_Toc147319574)

[CHƯƠNG 3. WEBSITE BÁN SÁCH COMIC STORE 68](#_Toc147319575)

[1.1. Khách hàng 68](#_Toc147319576)

[1.1.1. Màn hình Đăng nhập đăng ký 68](#_Toc147319577)

# 

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Stt** | **Từ viết tắt** | **Giải nghĩa** |
|  |  |  |
|  |  |  |

# LỜI CẢM ƠN

Lời đâu tiên em xin chân thành cảm ơn trường Đại học Công nghiệp Hà Nội, thầy cô giáo khoa Công nghệ thông tin đã tạo điều kiện thuận lợi và truyền dạy kiến thức cho em trong thời gian bốn năm qua để em có thể hoàn thành đề tài đồ án tốt nghiệp một cách tốt nhất. Trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp em luôn nhận được sự hướng dẫn nhiệt tình từ phía nhà trường và các thầy cô giáo . Và hơn hết em xin bày tỏ lòng biết ơn sâu sắc nhất tới thầy giáo hướng dẫn **TS. Nguyễn Bá Nghiễn** đã tận tình giúp đỡ, chỉ bảo em rất nhiều trong quá trình tìm hiểu nghiên cứu và hoàn thành đồ án tốt nghiệp.

Mặc dù em đã cố gắng thực hiện, nghiên cứu kĩ đề tài và hoàn thành tốt nhất có thể, nhưng kiến thức chuyên ngành của bản thân còn nhiều hạn chế nên trong phần báo cáo không tránh khỏi còn những thiếu sót, em rất mong nhận được sự đóng góp của tất cả thầy cô để đồ án của em được đầy đủ và hoàn chỉnh hơn.

Em xin chân thành cảm ơn.

# MỞ ĐẦU

Với sự phát triển nhanh chóng của công nghệ thông tin như hiện nay việc áp dụng công nghệ vào đời sống là điều hiển nhiên. Công nghệ hay internet và các dịch vụ trức tuyến đã thay đổi hoàn toàn cách con người truyền tải và trao đổi thông tin trên toàn cầu cùng với đó là thay đổi cách thức mọi người sống, làm việc, học tập, giáo tiếp và tiếp cận thông tin.

Con người càng phát triển nhu cầu mua sắm càng cao và mua sắm trực tuyến đã mở ra một cách thức mới giúp việc mua sắm trở nên tiện lợi dễ dàng hơn rất nhiều. Thay vì phải ra nhà sách, lựa chọn trong hàng nghìn quyển sách thì chúng ta có thể ở nhà, lựa chọn quyển sách mình mong muốn chỉ với vài bước tìm kiếm, đặt hàng và đợi người ta giao hàng đến

Sách cung cấp cho chúng ta không chỉ kiến thức, mà còn các bài học cuộc sống, kinh nghiệm sống, đồng thời cũng là một phương tiện giải trí khá hiệu quả.

Xuất phát từ những lý do trên và ứng dụng các kiến thức đã học, em quyết định chọn đề tài “Xây dựng website bán sách COMIC STORE”. Bằng việc xây dựng website bán sách này sẽ đáp ứng như cầu mua sách trực tuyến của khách hàng và đem lại doanh thu cho cửa hàng.

.

# TỔNG QUAN

## Tổng quan về đề tài

### Tên đề tài

Website bán sách Cosmic Store.

### Lý do chọn đề tài

#### Dịch vụ bán hàng online đang là xu hướng

* Bán hàng online đem đến lợi ích cho cả khách hàng và doanh nghiệp bán hàng. Khách hàng sẽ nhận được sự tiện lợi và đa dạng mẫu mã sẩn phẩm trong việc mua hàng. Doanh nghiệp khi bán hàng online dễ mở ra thêm thị trường mà không cần phải đầu tư thêm nhiều về chi phí.

#### Nhu cầu đọc sách của con người

* Xã hội phát triển nhanh con người ai cũng cần học tập, bổ sung kiến thức để có thể bắt kịp. Đọc sách là một trong những phương pháp cổ điển nhất để làm điều đó và nó không bao giờ lỗi thời. Mặc dù có nhiều dịch vụ đọc sách online nhưng theo quan điểm của nhiều người đọc một quyển sách trên tay có hứng đọc hơn nhiều so với đọc qua màn hình điện thoại hoặc máy tính. Vì vậy họ vẫn sẽ vui vẻ bỏ tiền ra mua về những cuốn sách mà mình thích để đọc và sưu tầm

#### Tiềm năng kinh doanh

* Với sự tăng trưởng ổn định cùng với sự đổi mới về công nghệ ngành công nghiệp sách và đặc biệt là kinh doanh sách online có tiềm năng phát triển lớn. Các doanh nghiệp nếu biết tận dụng hình thức kinh doanh online kết hợp với quảng cáo online, cung cấp cho người dùng dịch vụ tốt thì chắc chắn có thể thu hút thêm khách hàng cũng như tăng doanh số bán hàng.

## Tổng quan về lập trình Web

[Website](https://vi.wikipedia.org/wiki/Website) còn gọi là trang web hoặc trang mạng, là một tập hợp trang web, thường chỉ nằm trong một tên miền hoặc tên miền phụ trên World Wide Web của Internet. Một [website](https://vi.wikipedia.org/wiki/Website) là tập tin HTML hoặc XHTML có thể truy nhập dùng giao thức HTTP.

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

Một website có thể được xây dựng từ nhiều ngôn ngữ lập trình khác nhau như Java, PHP, ASP.NET,…. W[ebsite](https://vi.wikipedia.org/wiki/Website)  có thể được xây dựng từ các tệp tin HTML (trang mạng tĩnh) hoặc vận hành bằng các CMS chạy trên máy chủ (trang mạng động).

### Web tĩnh

**Web tĩnh**(Static Website) chính là cách gọi khác của những website phiên bản đầu tiên. Đây là trang web đơn thuần chỉ sử dụng hoàn toàn ngôn ngữ [**HTML**](https://www.w3schools.com/html/). Sau khi tải trang **HTML** từ máy chủ xuống, trình duyệt sẽ hỗ trợ biên dịch mã và hiển thị nội dung của trang web. Và lúc này người dùng hầu như không thể tương tác được với website.

### Web động

**Website động** (Dynamic Website) là một thuật ngữ dùng để chỉ những trang web được hỗ trợ bởi một phần mềm cơ sở web. Nói đúng hơn đây là một chương trình chạy với giao thức [**http://**](https://developer.mozilla.org/vi/docs/Web/HTTP). Có thể xem nó là một tập hợp các dữ liệu số hóa được tổ chức thành cơ sở dữ liệu và trình diễn trên các trang web dưới dạng văn bản, âm thanh, hình ảnh. **Dynamic website** được tích hợp thêm các phần xử lý thông tin và truy xuất dữ liệu.

### Khác nhau giữa web tĩnh và web động

#### Khác nhau về chức năng

**Web động** được tích hợp thêm các phần xử lý thông tin và truy xuất dữ liệu. Còn **web tĩnh** thì không. Người sở hữu trang web có quyền điều hành, chỉnh sửa và cập nhật thông tin trên trang web một cách dễ dàng. Khách hàng có được quyền trao đổi thông tin với chủ website và những khách hàng khác.

Khác với **website tĩnh**, **web động** luôn có thông tin mới được cập nhật bởi phần mềm quản trị web do các công ty thiết kế website cung cấp. Các thông tin này được lưu vào cơ sở dữ liệu của trang web và đưa ra sử dụng dựa vào yêu cầu của người dùng.

#### Khác nhau về ngôn ngữ lập trình

**Web tĩnh** đơn thuần chỉ sử dụng hoàn toàn ngôn ngữ lập trình HTML. Trong khi đó, **website động**được các chuyên gia lập trình sử dụng các ngôn ngữ lập trình tạo mã nguồn dựa theo yêu cầu của trang web. Hỗ trợ thêm các ngôn ngữ lập trình đa dạng như [**ASP.NET**](https://dotnet.microsoft.com/apps/aspnet)**,**[**PHP**](https://php.net/)**,JavaScript**…Và cơ sở dữ liệu SQL server, [**MySQL**](https://www.mysql.com/)**,MongoDb**…

#### Khác nhau về khả năng tương tác với khách hàng

Đối với **website tĩnh**, người dùng hầu như không thể tương tác được với website. Không thể trò chuyện, giao tiếp với nó. Nội dung của **Website tĩnh**đã được xác định ngay từ đầu. Nếu muốn thay đổi hay thêm bớt nội dung, người quản lý phải làm lại khuôn để có thể in ra những tờ báo mới. Nhưng **web động** lại khác, với các chương trình ứng dụng, khách hàng có được quyền trao đổi thông tin với chủ website và những khách hàng khác một cách dễ dàng nhất.

#### Khác nhau về chi phí bảo trì, nâng cấp

Người ta thường sử dụng trang **web tĩnh** khi nội dung website ít khi cập nhật và để tiết kiệm chi phí. Bạn không phải trả nhiều tiền cho việc lập trình. Nếu bạn là doanh nghiệp muốn tự mình làm website thì bạn có thể học các kiến thức căn bản và tự thiết kế một web tĩnh cho mình. Với **trang web tĩnh**, bạn có thể có một giao diện được thiết kế tự do hơn, cuốn hút hơn. Ngoài ra **website tĩnh** còn có một lợi thế vô song là thân thiện với các cơ chế tìm kiếm hơn so với **web động**. Bởi vì địa chỉ [**URL**](https://vi.wikipedia.org/wiki/URL)của các html trong web không chứa dấu chấm hỏi như trong **website động**. Tuy nhiên, khó quản lý nội dung và khó nâng cấp, bảo trì.

Website động tuy có chi phí xây dựng cao hơn. Tuy nhiên, nó dễ dàng nâng cấp và bảo trì. Có thể xây dựng được những web lớn.

#### Khác nhau về ứng dụng

**Web tĩnh** chỉ được ứng dụng khi nội dung website ít khi cập nhật. Dành cho những website nhỏ. Trong khi đó, **web động** lại được sử dụng đối với những we có tầm cỡ lớn có thể phát triển mở rộng thêm. Làm web tin tức, blog, làm website thương mại điện tử bán hàng, làm web giới thiệu sản phẩm công ty…

## Công nghệ sử dụng trong dự án

### Mô hình MVC

#### Tổng quan về mô hình MVC

**MVC** là viết tắt của cụm từ “**Model-View-Controller**“. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

* **Model** (dữ liệu) : Quản lí xử lí các dữ liệu
* **View** (giao diện) : Nới hiển thị dữ liệu cho người dùng
* **Controller** (bộ điều khiển): Điều khiển sự tương tác của hai thành phần **Model** và **View**

Mô hình MVC (MVC pattern) thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động , cũng như là các ứng dụng web

Mô hình MVC gồm 3 loại chính là thành phần bên trong không thể thiếu khi áp dụng mô hình này:

* **Model**: Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là **View** và **Controller**. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản. Chẳng hạn như biểu tượng hay là một nhân vật trong game.
* **View**: Đây là phần giao diện (theme) dành cho người sử dụng. View là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được
* **Controller:**Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả **Model** lẫn **View**. Nó nhận input và thực hiện các update tương ứng.

#### Luồng xử lý trong của mô hình MVC

Luồng xử lý trong của mô hình MVC, bạn có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:

* + Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị **Controller** trong MVC chặn lại để xem đó là URL request hay sự kiện.
  + Sau đó, **Controller** xử lý input của user rồi giao tiếp với**Model** trong MVC
  + **Model** chuẩn bị data và gửi lại cho **Controller**
  + Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt
  + Ở đây, **View** không giao tiếp trực tiếp với **Model**. Sự tương tác giữa **View** và **Model** sẽ chỉ được xử lý bởi **Controller**

#### Ưu điểm mô hình MVC

* Đầu tiên, nhắc tới ưu điểm mô hình MVC thì đó là băng thông (Bandwith) nhẹ vì không sử dụng viewstate nên khá tiết kiệm băng thông. Việc giảm băng thông giúp website hoạt động ổn định hơn
* Kiểm tra đơn giản và dễ dàng, kiểm tra lỗi phần mềm trước khi bàn giao lại cho người dùng
* Một lợi thế chính của MVC là nó tách biệt các phần Model, Controller và View với nhau
* Sử dụng mô hình MVC chức năng Controller có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau
* Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau
* Có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau
* Hỗ trợ TTD (test-driven development). Chúng ta có thể tạo một ứng dụng với unit test và viết các won test case
* Phiên bản mới nhất của MVC hỗ trợ trợ thiết kế responsive website mặc định và các mẫu cho mobile. Chúng ta có thể tạo công cụ View của riêng mình với cú pháp đơn giản hơn nhiều so với công cụ truyền thống

#### Nhược điểm mô hình MVC

* MVC đa phần phù hợp với công ty chuyên về website hoặc các dự án lớn thì mô hình này phù hợp hơn so với với các dự án nhỏ, lẻ vì khá là cồng kềnh và mất thời gian
* Mô hình MVC không hỗ trợ Preview như ASP.NET
* Khó triển khai

### MERN STACK

MERN là một thuật ngữ rút gọn của MongoDB, Express, React và Node. MERN Stack là một stack Javascript được thiết kế để giúp phát triển ứng dụng web toàn ngăn xếp dễ dàng hơn và nhanh hơn.

Tất cả bốn công nghệ này cung cấp một khuôn khổ hoàn chỉnh cho các nhà phát triển để tạo ra bất kỳ ứng dụng web nào. MERN đang tuân theo kiến ​​trúc 3 tầng truyền thống, bao gồm tầng hiển thị front-end (React.js), tầng ứng dụng (Express.js và Node.js) và tầng cơ sở dữ liệu (MongoDB)

### Nền tảng Nodejs

Nodejs là một nền tảng (platform) phát triển độc lập dựa trên V8 JavaScript engine. Đây là một trình thông dịch thực thi mã JavaScript cho phép tạo các ứng dụng web như video clip và diễn đàn, đặc biệt có thể mở rộng nhanh chóng và dễ dàng cũng như thu hẹp phạm vi hoạt động của các trang mạng mạng xã hội.

Đây được xem là một lợi thế khi NodeJS có thể hoạt động trên nhiều nền tảng hệ điều hành khác nhau, từ Windows, Linux đến OS X. NodeJS cung cấp một thư viện phong phú dưới dạng các mô-đun Javascript khác nhau giúp đơn giản hóa việc lập trình và giảm thiểu thời gian cần thiết.

Ý tưởng chính của Node js là sử dụng non – blocking, nhanh chóng định tuyến đầu vào/ đầu ra dữ liệu thông qua các tác vụ thời gian thực. Bởi vì Node js có khả năng mở rộng nhanh chóng và có thể xử lý một số lượng lớn các kết nối đồng thời với thông lượng cao.

Hầu hết các ứng dụng web truyền thống sử dụng tài nguyên hệ thống không hiệu quả khi các yêu cầu tạo ra các luồng xử lý yêu cầu mới và tiêu tốn khá nhiều bộ nhớ hệ thống. Vì vậy giải pháp mà NodeJS đưa ra chính là sử dụng luồng đơn có tên tiếng anh là Single – Threaded, đây là sự kết hợp non – blocking I/ O để thực hiện các yêu cầu, điều này cho phép nó hỗ trợ hàng chục hàng nghìn kết nối đồng thời.

* **Ưu điểm**
* IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.
* Sử dụng JavaScript – một ngôn ngữ lập trình dễ học.
* Chia sẻ cùng code ở cả phía client và server.
* NPM(Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ.
* Cộng đồng hỗ trợ tích cực.
* Cho phép stream các file có kích thước lớn
* **Nhược điểm**
* Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.
* Khó thao tác với cơ sử dữ liệu quan hệ.
* Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác.
* Cần có kiến thức tốt về JavaScript.
* Không phù hợp với các tác vụ đòi hỏi nhiều CPU.

### Thư viện Reactjs

ReactJS được hiểu nôm na là một thư viện mã nguồn mở chứa nhiều JavaScript và người tạo ra ReactJS chính là ông trùm với cái tên quen thuộc Facebook, được ra đời vào năm 2013. Mục đích của việc tạo ReactJS là tạo ra các ứng dụng web nhanh, hiệu quả và hấp dẫn với nỗ lực viết mã tối thiểu. Mục tiêu chính của ReactJS là bất kỳ trang web nào sử dụng ReactJS phải mượt mà, nhanh chóng, khả năng mở rộng cao và dễ thực hiện.

Nhìn chung, các tính năng và điểm mạnh của ReactJS thường đến từ việc tập trung vào các phần riêng lẻ. Do đó, khi làm việc với web thay vì toàn bộ ứng dụng của một trang web bằng ReactJS, các nhà phát triển tính năng có thể tách rời và chuyển đổi giao diện người dùng từ những cách phức tạp và biến nó thành những phần đơn giản hơn. Điều này có nghĩa là kết xuất dữ liệu không chỉ ở phía máy chủ. Thực hiện với các vị trí, nhưng bạn cũng có thể thực hiện tại vị trí khách hàng khi sử dụng ReactJS.

Tạo cho chính bản thân ReactJS một dom ảo, đây là nơi các component được tồn tại trên đó. Tạo một dom như thế này sẽ cải thiện hiệu suất rất nhiều. Nếu bạn cần thay đổi hoặc cập nhật một phép tính trong dom, ReactJS sẽ tính toán nó trước và phần còn lại sẽ chạy trong dom để thực hiện công việc, vì vậy ReactJS có thể tránh các thao tác cần thiết trong dom mà không phải trả thêm phí.

Việc viết các đoạn mã JS sẽ trở nên dễ dàng sử dụng một cú pháp đặc biệt đó chính là cú pháp JSX, cú pháp này cho phép bạn kết hợp mã HTML và Javascript. Ngoài ra, bạn có thể thêm mã bổ sung vào hàm kết xuất mà không cần nối chuỗi. Đây được coi là một trong những tính năng thú vị của ReactJS, việc chuyển đổi các đoạn **HTML** sang các constructor đều được thực hiện bởi trình chuyển đổi chính là JSX.

Khi khởi chạy ReactJS, đừng quên cài đặt thêm ứng dụng mở rộng Chrome chuyên dành cho ReactJS. Điều này làm cho việc debug mã của bạn dễ dàng hơn. Sau khi cài đặt, bạn có thể nhìn thẳng vào mái Virtual Dom điều đó đồng nghĩa với việc bạn đang theo dõi một cây dom thông thường

Đây là một trong những điều đặc biệt và độc đáo của ReactJS. Đây cũng là một vấn đề lớn đối với các khuôn khổ JS vì mặc dù có nhiều cải tiến nhưng hầu hết các khuôn khổ JS không thân thiện với công cụ tìm kiếm. ReactJS hỗ trợ hiển thị và trả về trình duyệt dưới dạng trang web khi bạn chạy ReactJS trên máy chủ và Dom ảo, vì vậy chúng tôi rất tự hào nói rằng chúng tôi không thuộc nhóm không thân thiện với SEO. Bởi vì điều này, React rất phù hợp với tính cách thân thiện với SEO của nó.

Trên thực tế, React chỉ là một khung nhìn, vì vậy React nên được kết hợp với những thứ như Redux và Flux. Một luồng dữ liệu luôn cần thiết, Redux là một trong những nguồn dữ liệu được sử dụng phổ biến nhất và tư duy của React cũng khá tốt.

### Framework Expressjs

Expressjs hay còn được viết là Express js, Express.js. Đây là một framework mã nguồn mở miễn phí cho Node.js. Express.js được sử dụng trong thiết kế và xây dựng các ứng dụng web một cách đơn giản và nhanh chóng.

Vì Express js chỉ yêu cầu ngôn ngữ lập trình Javascript nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn với các lập trình viên và nhà phát triển.Expressjs cũng là một khuôn khổ của Node.js do đó hầu hết các mã code đã được viết sẵn cho các lập trình viên có thể làm việc.

Nhờ có Expressjs mà các nhà lập trình có thể dễ dàng tạo các ứng dụng 1 web, nhiều web hoặc kết hợp. Do có dung lượng khá nhẹ, Expressjs giúp cho việc tổ chức các ứng dụng web thành một kiến trúc MVC có tổ chức hơn.Để có thể sử dụng được mã nguồn này, chúng ta cần phải biết về Javascript và HTML.

Expressjs cũng là một phần của công nghệ giúp quản lý các ứng dụng web một cách dễ dàng hơn hay còn được gọi là ngăn xếp phần mềm MEAN.Nhờ có thư viện Javascript của Express js đã giúp cho các nhà lập trình xây dựng nên các ứng dụng web hiệu quả và nhanh chóng hơn. Expressjs cũng được sử dụng để nâng cao các chức năng của Node.js.

Trên thực tế, nếu không sử dụng Express.js, bạn sẽ phải thực hiện rất nhiều bước lập trình phức tạp để xây dựng nên một [API](https://itnavi.com.vn/blog/api-la-gi/) hiệu quả. Express js đã giúp cho việc lập trình trong Node.js trở nên dễ dàng hơn và có nhiều tính năng mới bổ sung.

* **Những tính năng của Expressjs là gì?**
* Phát triển máy chủ nhanh chóng: Expressjs cung cấp nhiều tính năng dưới dạng các hàm để dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này đã loại bỏ nhu cầu viết mã từ đó tiết kiệm được thời gian.
* Phần mềm trung gian Middleware: Đây là phần mềm trung gian có quyền truy cập vào cơ sở dữ liệu, yêu cầu của khách hàng và những phần mềm trung gian khác. Phần mềm Middleware này chịu trách nhiệm chính cho việc tổ chức có hệ thống các chức năng của Express.js.
* Định tuyến - Routing: Express js cung cấp cơ chế định tuyến giúp duy trì trạng thái của website với sự trợ giúp của URL.
* Tạo mẫu - Templating: Các công cụ tạo khuôn mẫu được Express.js cung cấp cho phép các nhà xây dựng nội dung động trên các website bằng cách tạo dựng các mẫu HTML ở phía máy chủ.
* Gỡ lỗi - Debugging: Để phát triển thành công các ứng dụng web không thể thiết đi việc gỡ lỗi. Giờ đây với Expressjs việc gỡ lỗi đã trở nên dễ dàng hơn nhờ khả năng xác định chính xác các phần ứng dụng web có lỗi.

### Dịch vụ lưu trữ dữ liệu Mongodb

**MongoDB** là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. [**MongoDB**](https://topdev.vn/viec-lam-it/mongodb-kt90) sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

* **Ưu điểm của MongoDB**
* Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
* Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.
* **Nhược điểm của MongoDB**
* Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.
* Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.
* Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.
* Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

### Thư viện thiết kế Ant Design

Trong quá trình phát triển các ứng dụng trên máy tính, có nhiều thông số thiết kế và hệ thống xử lý khác nhau rất phức tạp, điều này gây khó khăn gấp đôi cho các nhà thiết kế và lập trình viên, đồng thời sẽ làm giảm hiệu quả cho việc phát triển này. Sau khi có cho mình các dự án lớn và sơ lược lại, đội ngũ Ant UED đã cho ra mắt ngôn ngữ thiết kế Ant Design dành cho giao diện của ứng dụng, nhằm mục đích nhất quán các thông số giao diện người dùng trên nền tảng dự án nội bộ, giảm chi phí không cần thiết của việc triển khai ý tưởng và cung cấp nhiều tài nguyên cho việc thiết kế và lập trình front-end.

Ant Design đặc biệt được tạo ra cho các ứng dụng máy tính nội bộ, cam kết trải nghiệm của người dùng và các nhà thiết kế sản phẩm. Những người thiết kế giao diện và thiết kế trải nghiệm cho người dùng được gọi chung là thiết kế sản phẩm, Ant Design sẽ xóa bỏ ranh giới giữa việc quản lý sản phẩm, thiết kế tương tác, thiết kế hình ảnh, phát triển giao diện và quản trị dữ liệu người dùng. Nhờ việc tận dụng sự đồng nhất về đặc điểm kỹ thuật, Ant Design đơn giản hóa việc thiết kế và lập trình cho những người đang thực hiện Project, điều này sẽ giúp họ có thêm nhiều kinh nghiệm và tăng hiệu quả phát triển giao diện của sản phẩm và ứng dụng.

* Giá trị về thiết kế của Ant Design:
  + Sự tự nhiên
  + Sự chắc chắn

## Các kiến thức cần nắm được trong quá trình thực hiện đồ án

* Cần hiểu được cơ bản về ngôn ngữ lập trình Javascript
* Hiểu và sử dụng các kiến thức về Reactjs, Nodejs, Express.js
* Biết thao thác với Mongodb
* Ngoài ra còn cầm nắm được cấu trúc của một mô hình bán hàng online

# THIẾT KẾ HỆ THỐNG

## Khảo sát hệ thống

### Đối tượng khảo sát

Khảo sát một số website bán sách trực tuyến và thanh toán trực tuyến:

* <https://cachep.vn/>
* <https://ebook.waka.vn/>
* <https://bookbuy.vn/>
* <https://www.fahasa.com/>

### Quy trình tác nghiệp

* Mỗi hệ thống bán hàng trực tuyến thường gồm 2 phần chính:
  + Phần thứ nhất: phần giao dịch với khách hàng trên Internet
  + Phần thứ hai: phần danh cho quản trị viên cửa hàng quản trị hệ thống
* Mỗi phần sẽ có các chức năng con và phần mở rộng khác nhau phục vụ cho mục đích kinh doanh cụ thể của doanh nghiệp

### Quy trình thực hiện mua hàng của khách hàng

* Khách hàng duyệt ứng dụng trên trình duyệt Web.
* Trình duyệt sẽ hiển thị các sản phẩm và dịch vụ mà doanh nghiệp cung ứng. Khách hàng có thể duyệt sản phẩm theo chủng loại được liệt kê sẵn hoặc có thể sử dụng chức năng tìm kiếm để tìm kiếm theo tên sản phẩm.
* Sau khi tìm kiếm, tham khảo các thông tin về sản phẩm, khách hàng có thể đi đến quyết định là mua sản phẩm nào đó. Để làm điều này khách hành cần đưa sản phẩm đó vào trong giỏ hàng. Giỏ hàng là nơi chứa thông tin về các sản phẩm mà khách hàng định đặt mua. Khách hàng có thể thay đổi số lượng, thêm hoặc xoá sản phẩm trong giỏ hàng. Sau đó khách hàng có thể tiếp tục xem các sản phẩm khác hoặc tiến hành đặt hàng.
* Để đặt hàng thì khách hàng phải đăng nhập hệ thống. Nếu khách hàng chưa có tài khoản thì hệ thống sẽ chỉ đến trang đăng kí. Sau đó khách hàng sẽ cung cấp các thông tin về việc giao hàng cũng như thanh toán với doanh nghiệp. Nếu doanh nghiệp mà có liên kết với các Ngân hàng để xây dựng một hệ thống thanh toán trực tuyến thì hệ thống đang hoạt động sẽ chỉ người dùng đến trang thanh toán để người dùng cung cấp các thông tin về tài khoản (hoặc thẻ) của họ ở Ngân hàng. Ngân hàng sẽ dựa vào thông tin đó để thực hiện các nghiệp vụ thanh toán. Sau đó Ngân hàng sẽ gửi thông điệp thông báo tới doanh nghiệp.
* Sau khi đã cung cấp đầy đủ thông tin, khách hàng có thể xem lại, chỉnh sửa lại đơn hàng rồi gửi đơn hàng. Hệ thống sẽ phản hồi thông báo cho Khách hàng biết là đơn hàng đã được gửi và cửa hàng sẽ liên hệ lại với khách hàng về việc thanh toán và giao hàng. Nếu khách hàng chấp nhận thì thanh toán với cửa hàng và cửa hàng giao hàng đến khách hàng. Quá trình mua hàng kết thúc.
* Ngoài ra khách hàng còn có thể thực hiện các công việc khác như: Gửi yêu cầu, góp ý, Download tài liệu điện tử miễn phí, tham khảo cước phí vận chuyển, thay đổi thông tin tài khoản
* Các thông tin cần hiển thị trên trang chủ của ứng dụng: Danh mục loại sách mà cửa hàng cung cấp, sách mới phát hành, sách bán chạy, một vài chức năng như: tìm kiếm, đăng ký, đăng nhập, xem giỏ hàng Hoặc có thể cho cho phép đăng các quảng cáo để tăng thu nhập
* Các thông tin về tài khoản của khách hàng bao gồm: Điạ chỉ email, mật khẩu truy nhập, tên khách hàng, địa chỉ nơi ở, tỉnh/thành, quốc gia, điện thoại liên hệ.
* Mỗi cuốn sách có các thông tin sau: Tên sách, sách thuộc loại sách nào, giá bìa, giá bán, thông tin khuyến mại, hình ảnh minh hoạ, số lượng trong kho, 3 tên tác giả, năm xuất bản, nhà xuất bản, nội dung tóm tắt, kích thước, trọng lượng sách (phục vụ cho việc tính giá thành vận chuyển).
* Giỏ hàng lưu các thông tin về sản phẩm khách hàng chọn mua: Tên sản phẩm, số lượng đặt mua, đơn giá và tổng trị giá các sản phẩm trong giỏ hàng. Ngoài ra gồm các chức thêm, xoá, cập nhật sản phẩm trong giỏ hàng.
* Đơn đặt hàng gồm các thông tin về khách hàng: Email của khách hàng, địa chỉ giao hàng, số điện thoại và phương thức thanh toán. Các thông tin về sản phẩm đặt mua: tên sản phẩm, số lượng, đơn giá và tổng trị giá đơn hàng.
* Về phương thức thanh toán thì hệ thống hỗ trợ các phương thức thanh toán: Trực tiếp, chuyển khoản qua ngân hàng, chuyển qua đường bưu điện. Thông tin về số tài khoản của doanh nghiệp được cung cấp cho khách hàng để khách hàng thanh toán trước khi giao hàng.

### Quy trình thực hiện quản trị hệ thống của nhân viên quản trị

Để truy cập vào chức năng quản trị hệ thống, Quản trị viên cần có một tài khoản để truy nhập. Sau khi truy nhập quản trị viên có thể thực hiện các chức năng sau:

* Quản trị danh mục loại sách: Bao gồm các công việc thêm, xoá, cập nhật thông tin về các loại sách.
* Quản trị sách: Thêm, xóa và cập nhật thông tin về mỗi cuốn sách.
* Quản trị người dùng: Thêm, xoá, cập nhật thông tin về nhân viên.
* Quản trị đơn hàng: Xử lý các đơn đặt hàng của khách hàng, thanh toán và giao hàng với khách hàng.
* Quản trị các phương thức thanh toán: Thêm, xoá, cập nhật thông tin về các phương thức thanh toán.
* Quản trị danh mục Tỉnh/Thành phố: Thêm, xóa, cập nhật Tỉnh/Thành.
* Quản trị khách hàng: Theo dõi thông tin về khách hàng, xóa khách hàng khỏi danh sách thành viên khi cần.
* Quản trị yêu cầu của khách hàng: Xử lý và phản hồi các yêu cầu mà khách hàng đã gửi.
* Quản lý tài chính: Sử dụng các thông tin về các hóa đơn đã thanh toán với khách hàng để có thể thống kê tài chính từng tháng, từng quý.

### Các thông tin vào ra của hệ thống

#### Các thông tin đầu vào

* Người quản trị nhập thông tin về sách, loại sách mà doanh nghiệp đang bán, các tin tức cập nhật thường xuyên. Còn khách hàng cung cấp thông tin cá nhân, gửi bài viết góp ý cho doanh nghiệp giúp doanh nghiệp ngày càng hoàn thiện hơn. Cụ thể gồm các thông tin sau:
  + Các thông tin giới thiệu về hệ thống.
  + Các thông tin về loại sách và từng cuốn sách.
  + Các thông tin về khách hàng
  + Các thông tin về đơn hàng
  + Các bài viết, tin tức, thông tin khuyến mại
  + Các bài góp ý, ý kiến, hỏi đáp của khách hàng; bài trả lời,…

#### Các thông tin đầu ra

* Đưa ra các thông tin cần thiết cho khách hàng ở mọi khía cạnh mà khách hàng quan tâm đến sản phẩm của công ty, đồng thời kiểm soát được hoạt động của công ty.
* Đưa ra sản phẩm, tin tức khách hàng yêu cầu tìm kiếm, đưa ra sản phẩm mới nhất.
* Cụ thể bao gồm:
  + Cho phép tra cứu, xem thông tin về từng cuốn sách.
  + Cho phép khách hàng lựa chọn hàng, lựa chọn hình thức thanh toán và thanh toán qua đơn hàng.
  + Đưa ra các hóa đơn thanh toán, khi giao hàng cho khách hàng.
  + Cho phép tra cứu, tìm kiếm các thông tin về thành viên.
  + Cho phép khách hàng đóng góp ý kiến, bình luận, đánh giá chất lượng của sản phẩm
  + Cho phép khách hàng đăng ký thành viên của website
  + Cho phép ban quản trị cập nhật các tin tức, bài viết, thông tin khuyến mại, trả lời các ý kiến hỏi đáp của khách hàng
  + Cho phép ban quản trị cập nhật thông tin loại sách và mỗi cuốn sách.
  + Cho phép ban quản trị theo dõi thành viên

### Giới thiệu chung về website bán sách Comic Store

Website bán sách Comic Store **chuyên cung cấp truyện tranh gồm nhiều thể loại như manga của Nhật Bản hay Comic của Mỹ hay một số truyện tranh truyện thống của Việt Nam. Đây là một địa điểm rất tốt cho các độc giả và người yêu sách tìm kiếm và mua sắm sách.**

### Yêu cầu hệ thống

#### Yêu cầu chức năng

- Khách hàng:

+ Đăng nhập

+ Đăng ký

+ Tìm kiếm sản phẩm

+ Xem sản phẩm theo loại sách

+ Xem thông tin chi tiết sản phẩm

+ Thêm sản phẩm vào giỏ hàng

+ Đặt hàng

+ Kiểm tra đơn hàng

+ Cập nhật thông tin cá nhân

+ Đặt lại mật khẩu

+ Xem theo thể loại

+ Xem trang chủ

+ Cập nhật các sản phẩm có trong giỏ hàng

- Người quản trị

+ Đăng nhập

+ Quản lý loại sách

+ Quản lý sản phẩm

+ Quản lý đơn hàng

+ Quản lý tài khoản

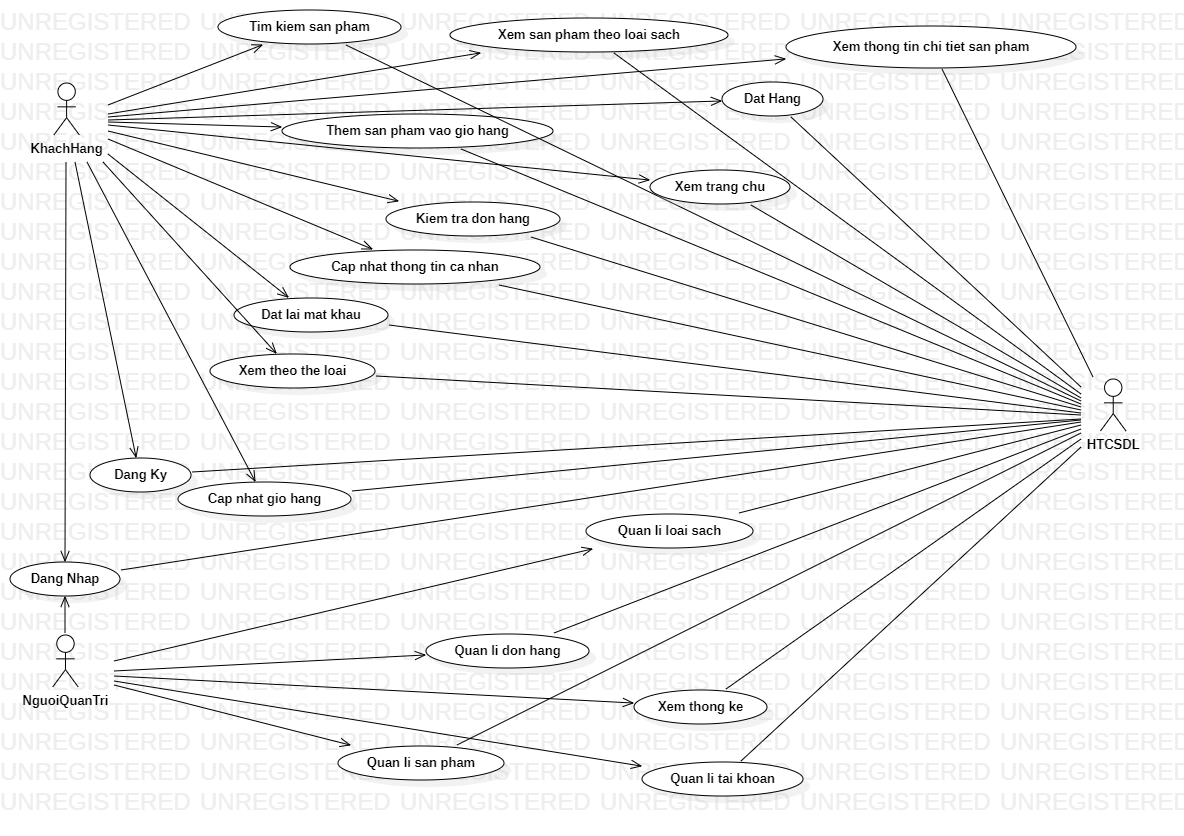
+ Báo cáo, thống kê

#### Yêu cầu phi chức năng

|  |  |  |
| --- | --- | --- |
| **Mục** | **Tên yêu cầu** | **Mô tả yêu cầu** |
| 1 | Giao diện | Giao diện hệ thống dễ sử dụng, trực quan, thân thiện |
| 2 | Tốc độ xử lý | Hệ thống 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 trình duyệt |

## Phân tích thiết kế hệ thống

Biểu đồ tổng quát Use case:

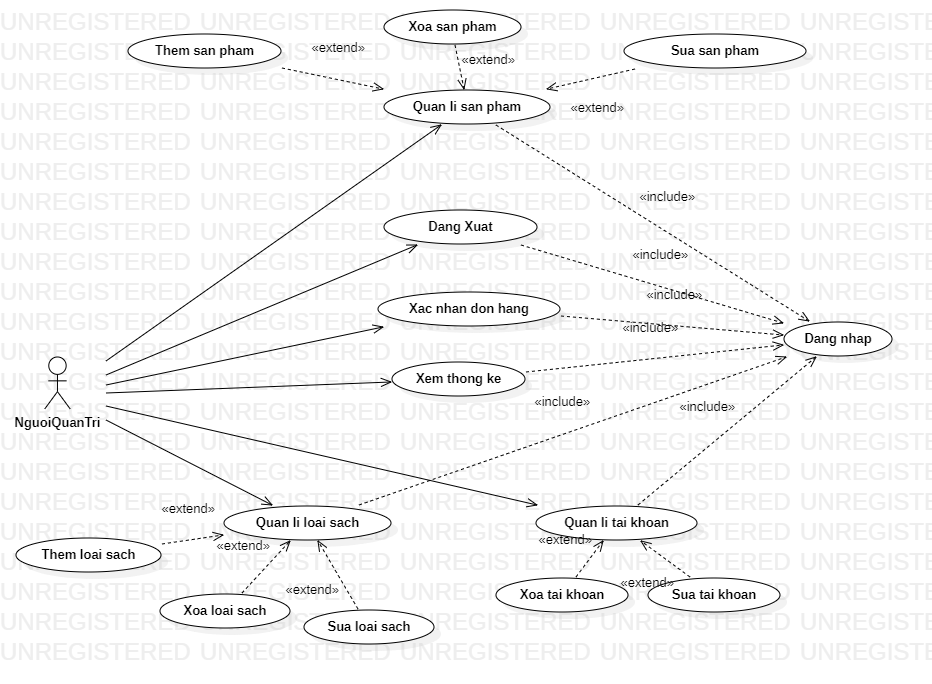


*Hình 1: Biểu đồ use case tổng quát*

### Người quản trị

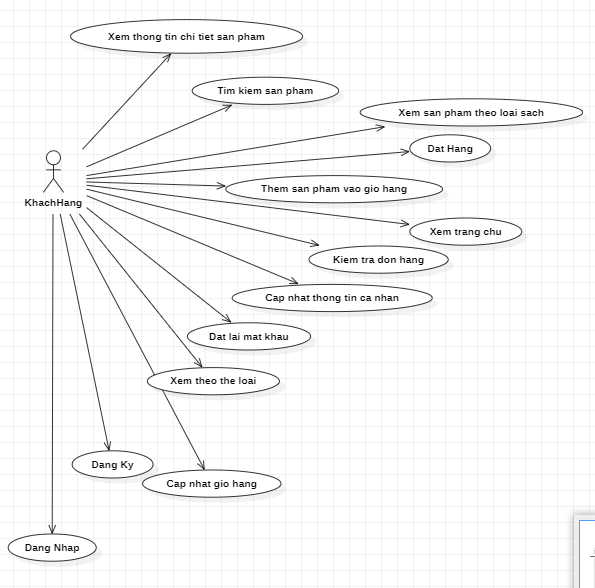


*Hình2 : Biểu đồ use case tổng quát quản trị*

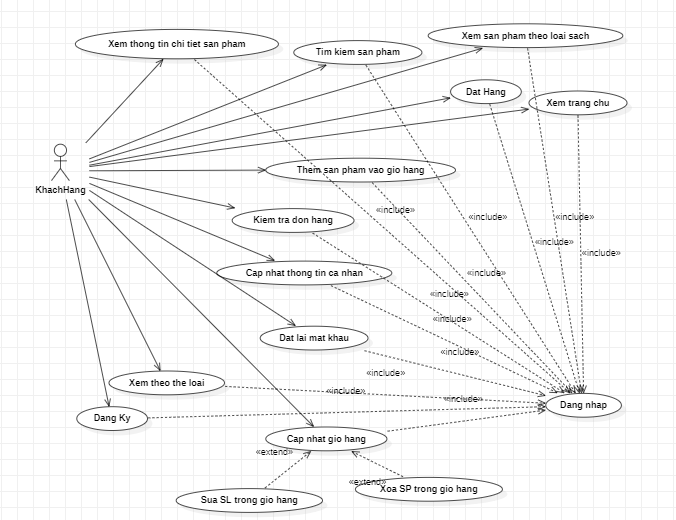
**

*Hình 3: Biểu đồ use case phân rã quản trị*

### Khách hàng



Hình 4: *Biểu đồ use case tổng quát khách hàng*

 Hình 5: *Biểu đồ use case phân rã khách hàng*

### Mô tả use case

#### Mô tả use case Đăng nhập (người quản trị)

* Use case này cho phép quản trị viên đăng nhập vào hệ thống với quyền quản trị
* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi người dùng kích vào nút đăng nhập. Người dùng sẽ được chuyển đến trang đăng nhập

- Người dùng nhập đầy đủ thông tin yêu cầu và nhấp chọn “Đăng nhập”

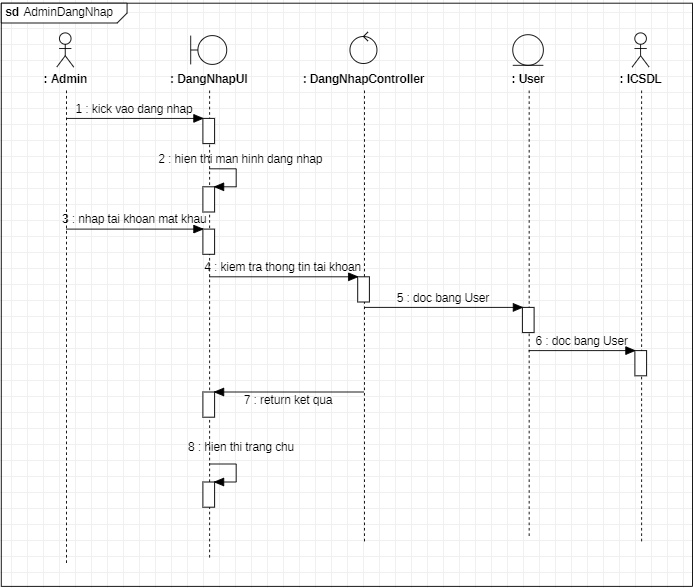
- Hệ thống kiểm tra thông tin người dùng trong bảng USER và chuyển vào giao diện trang web chính, use case kết thúc

* Luồng rẽ nhánh:

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

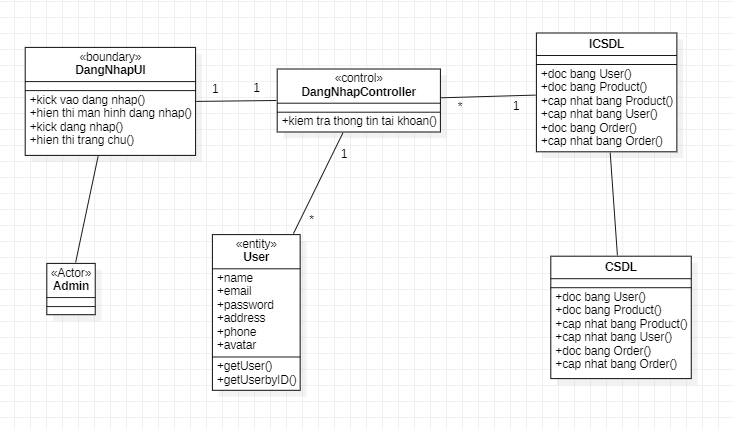
- Tại bước 2 luồng cơ bản, nếu người dùng nhập thông tin không hợp lệ. Hệ thống hiển thị thông báo yêu cầu người dùng nhập lại.

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Không có
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có
* Biểu đồ tuần tự:



*Hình 6: Biểu đồ tuần tự use case Đăng nhập(admin)*

* Biểu đồ lớp



*Hình 7: Biểu đồ lớp use case Đăng nhập(admin)*

#### Mô tả use case quản lý sản phẩm

Use case này cho phép người quản trị xem, thêm, sửa, xóa các sản phẩm trong bảng PRODUCT.

* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi admin bấm vào phần sản phẩm trong phần “Quản lí hệ thống”

- Hệ thống sẽ lấy tất cả thông tin từ bảng PRODUCT và hiển thị lên màn hình.

- Thêm sản phẩm: Người quản trị click vào biểu tượng “+” trên màn hình, hệ thống sẽ chuyển sang màn hình yêu cầu nhập đầy đủ thông tin sản phẩm. Người quản trị click nút “Xác nhận”, hệ thống sẽ tạo một sản phẩm mới trong bảng PRODUCT và hiển thị lại danh sách sản phẩm và xuất hiện một thống báo thêm thành công là “Success”

- Sửa sản phẩm: Người quản trị click vào nút màu vàng ở cột Action của sản phẩm muốn sửa, hệ thống sẽ hiển thị màn hình chi tiết sản phẩm, người quản trị nhập thông tin cần sửa. Người quản trị click nút “Cập nhật”, hệ thống sẽ lưu lại vào bảng PRODUCT và hiển thị lại danh sách sản phẩm lên màn hình. và xuất hiện một thống báo cập nhật thành công là “Success”

- Xóa sản phẩm: Người quản trị nhấn chọn vào nút màu đỏ trong cột Action của sản phẩm muốn xóa, hệ thống sẽ hiển thị một popup hỏi lại xem có thực sự muốn xóa không. Nhấn chọn “OK”, hệ thống sẽ xóa sản phẩm ra khỏi bảng PRODUCT và xuất hiện một thống báo xóa thành công là “Success”

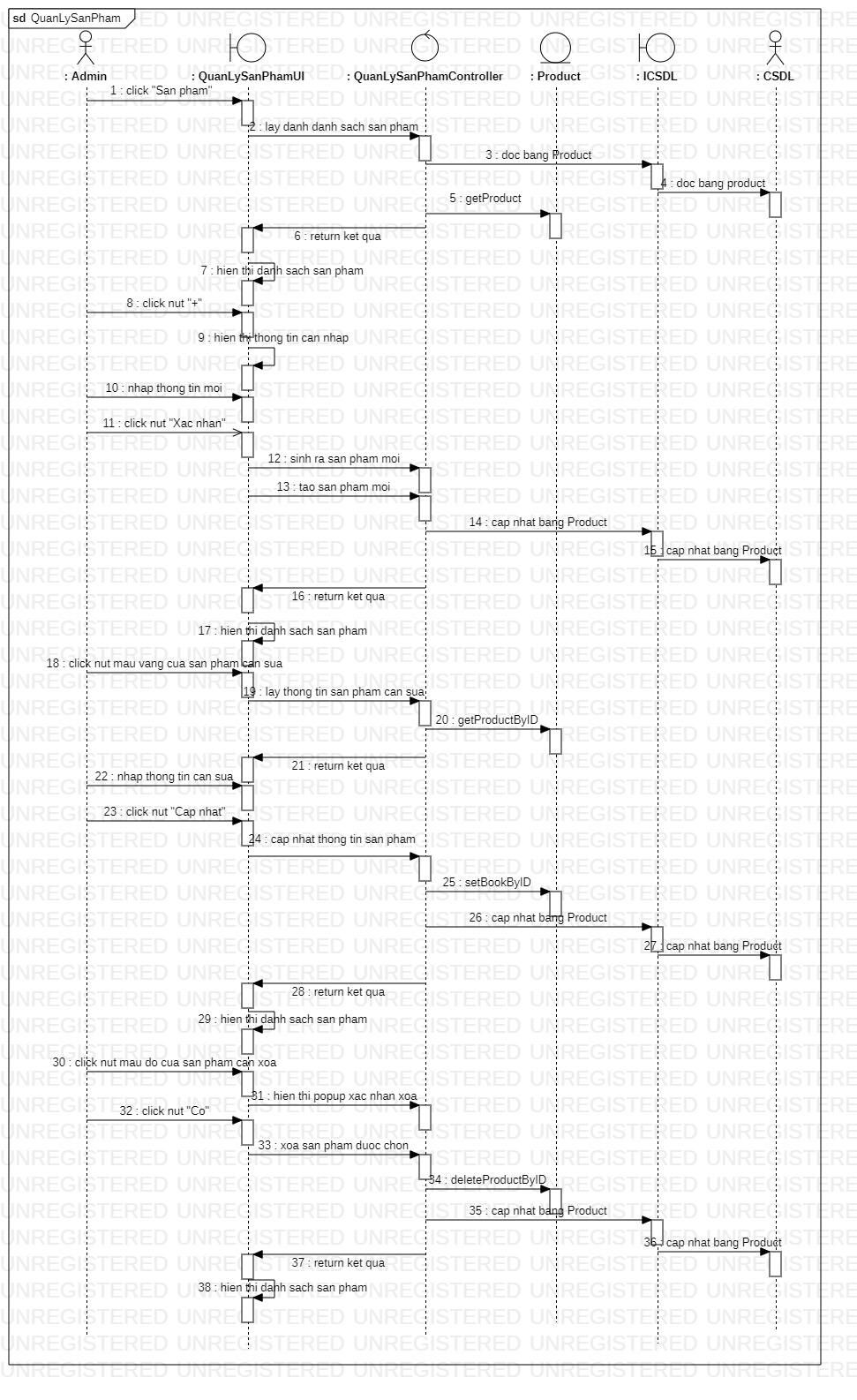
- Use case kết thúc.

* Luồng rẽ nhánh:

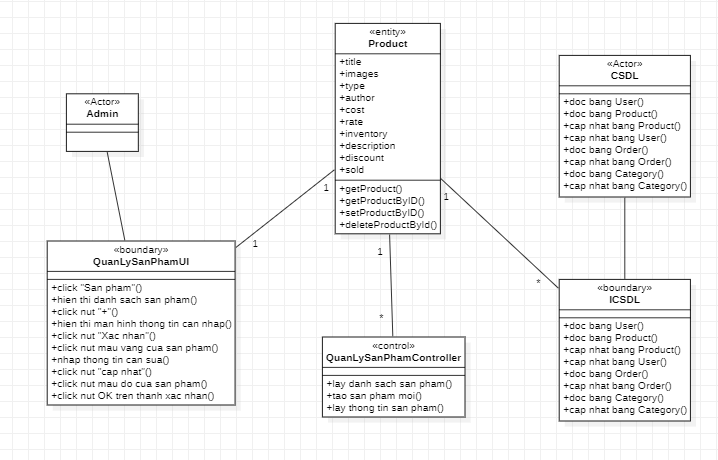
- Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.

- Người quản trị nhấn thoát khỏi trang, use case kết thúc

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Đăng nhập thành công tài khoản người quản trị vào website
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có
* Biểu đồ tuần tự

 *Hình 8: Biểu đồ tuần tự use case* quản lý sản phẩm *(admin)*

* Biểu đồ lớp



*Hình 9: Biểu đồ lớp use case* quản lý sản phẩm *(admin)*

#### Use case quản lý thể loại sách

Use case này cho phép người quản trị xem thêm các thể loại sách

* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi admin bấm vào “Thể loại” trên màn hình

- Hệ thống sẽ lấy tất cả thông tin từ bảng CATEGORY và hiển thị lên màn hình.

- Người quản trị click vào biểu tượng “+” trong mục sản phẩm của phần quản trị, hệ thống sẽ chuyển sang phần nhật thông tin sản phẩm.

- Xem thể loại: Tại ô thể loại người quản trị chọn vào phân input, hệ thống sẽ lây danh sách các thể loại đã có trong bảng CATEGORY.

- Thêm thể loại: Sau khi ấn vào phần input lựa chọn cuối cùng sẽ là “thêm thể loại” cho người quản trị có thể thêm thể loại mới. Click “thêm thể loại” và nhập thể loại mới vào ô xuất hiện ra rồi bổ sung tiếp các phần của sản phẩm. Khi nhấn xác nhận để tạo sản phẩm mới thể loại vừa thêm sẽ được thêm vào trong bảng CATEGORY

- Use case kết thúc.

* Luồng rẽ nhánh:

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

- Người quản trị nhấn thoát khỏi trang, use case kết thúc

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Đăng nhập thành công tài khoản người quản trị vào website
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có
* Dữ liệu liên quan:

#### Use quản lý đơn hàng

Use case này cho phép người quản trị có thể xem và cập nhật trạng thái đơn hàng.

* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi admin bấm vào “Đơn hàng” trên màn hình

- Hệ thống sẽ lấy tất cả thông tin từ bảng ORDER và hiển thị lên màn hình.

- Cập nhật đơn hàng: Người quản trị chọn đơn hàng muốn cập nhật, hệ thống sẽ hiển thị màn hình chi tiết đơn hàng, người quản trị xác nhận trạng thái đơn hàng. Người quản trị kick nút “Cập nhật”, hệ thống sẽ lưu lại vào bảng ORDER và hiển thị lại lên màn hình.

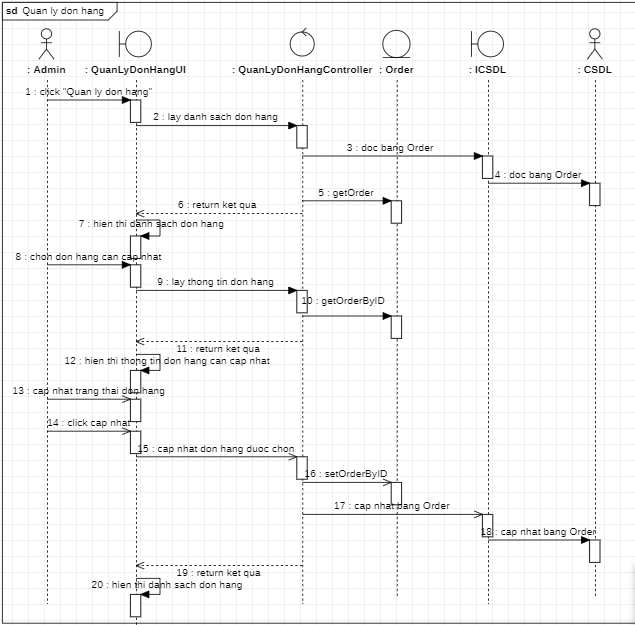
- Use case kết thúc.

* Luồng rẽ nhánh:

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

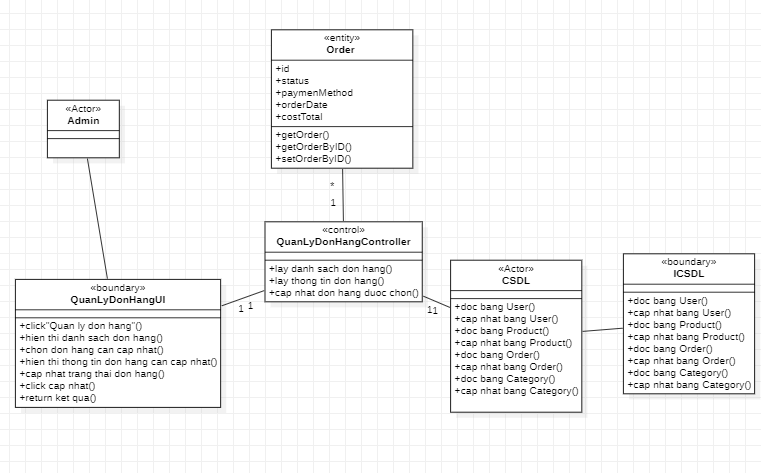
- Người quản trị nhấn thoát khỏi trang, use case kết thúc

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Đăng nhập thành công tài khoản người quản trị vào website
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có
* Biểu đồ tuần tự:



*Hình 10: Biểu đồ tuần tự use case* quản lý đơn hàng *(admin)*

* Biểu đồ lớp:



*Hình 11: Biểu đồ lớp use case đơn hàng (admin)*

#### Use case quản lý tài khoản

Use case này cho phép người quản trị xem, sửa, xóa tài khoản trong bảng USER

* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi admin bấm vào “Người dùng” trong phần quản lý hệ thống

- Hệ thống sẽ lấy tất cả thông tin từ bảng USER và hiển thị lên màn hình.

- Sửa thông tin tài khoản: Người quản trị chọn vào nút mài vàng của tài khoản muốn sửa, hệ thống sẽ hiển thị màn hình chi tiết thông tin tài khoản, người quản trị nhập thông tin cần sửa. Người quản trị kick nút “Cập nhật”, hệ thống sẽ lưu lại vào bảng USER và hiển thị lại lên màn hình.

- Xóa tài khoản: Người quản trị nhấn chọn vào nút màu đỏ của tài khoản muốn xóa, hệ thống sẽ hiển thị một dialog với các thao tác người dùng có thể tương tác. Nhấn chọn “Có”, hệ thống sẽ xóa tài khoản ra khỏi bảng USER.

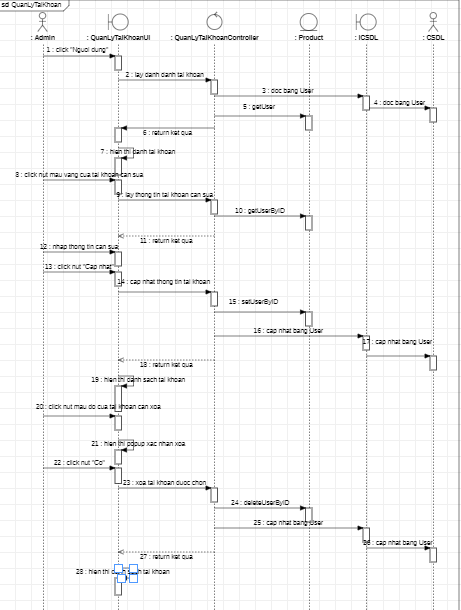
- Use case kết thúc.

* Luồng rẽ nhánh:

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

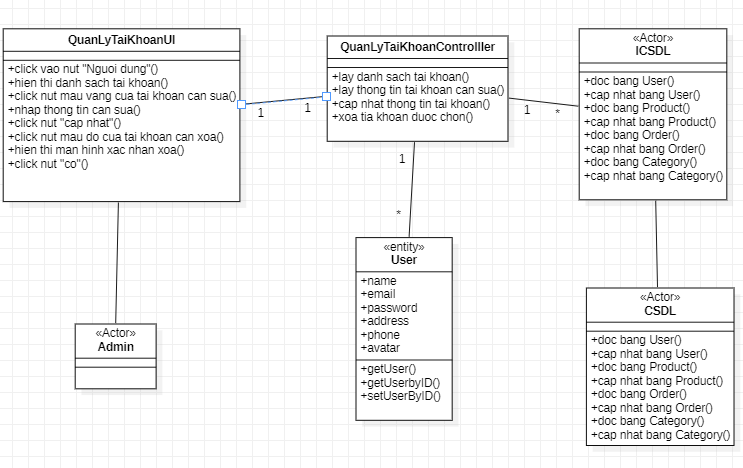
- Người quản trị nhấn thoát khỏi trang, use case kết thúc

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Đăng nhập thành công tài khoản người quản trị vào website
* Hậu điều kiện: không có
* Điểm mở rộng: không có
* Biểu đồ tuần tự



*Hình 12: Biểu đồ tuần tự use case quản lí tài khoản(admin)*

* Biểu đồ lớp

 *Hình 12: Biểu đồ lớp use case quản lí tài khoản(admin)*

#### Use case thống kê

Use case này cho phép quản trị xem thống kê doanh thu theo tháng

* Luồng sự kiện:
* Luồng cơ bản:

- Use case này bắt đầu khi người quản trị kích vào “ Thống kê ” trên menu quản trị. Hệ thống sẽ lấy thông tin từ bảng ORDER và thống kê doanh thu để hiển thị lên màn hình

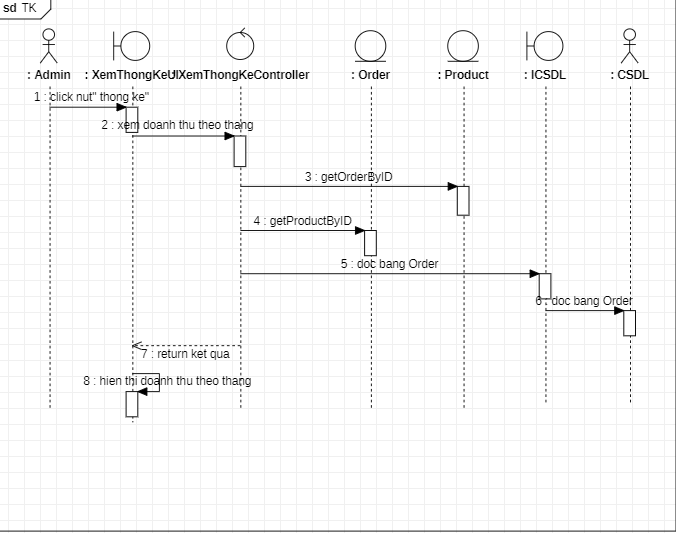
- Use case kết thúc.

* Luồng rẽ nhánh:

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

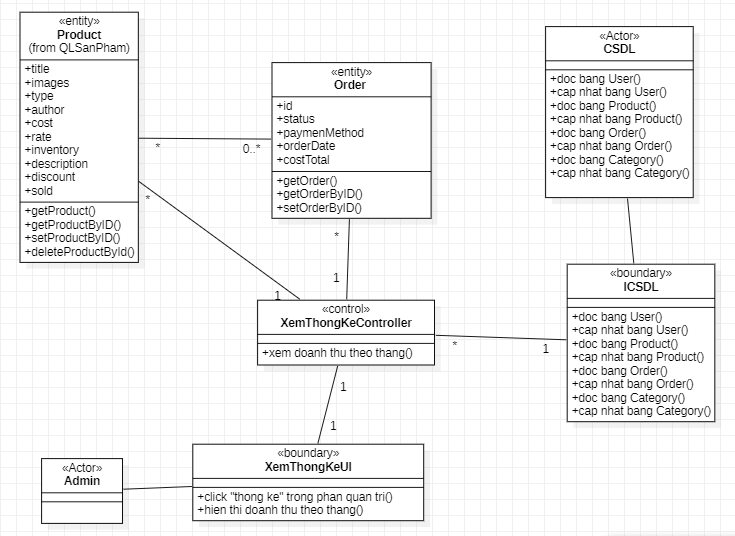
- Người quản trị nhấn thoát khỏi trang, use case kết thúc

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Đăng nhập thành công tài khoản người quản trị vào website
* Hậu điều kiện: không có
* Điểm mở rộng: không có
* Biểu đồ tuần tự



*Hình 14: Biểu đồ tuần tự use case thống kê(admin)*

* Biểu đồ lớp



*Hình 15: Biểu đồ lớp use case thống kê(admin)*

#### Use case đăng ký

Use case cho phép người dùng đăng ký tài khoản trên hệ thống cửa hàng

* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi người dùng nhấn chọn “Đăng ký”

- Hệ thống yêu cầu người dùng nhập vào email, họ tên, số điện thoại, mật khẩu và xác nhận mật khẩu

- Người dùng nhập đầy đủ thông tin yêu cầu và nhấp chọn “Đăng ký”

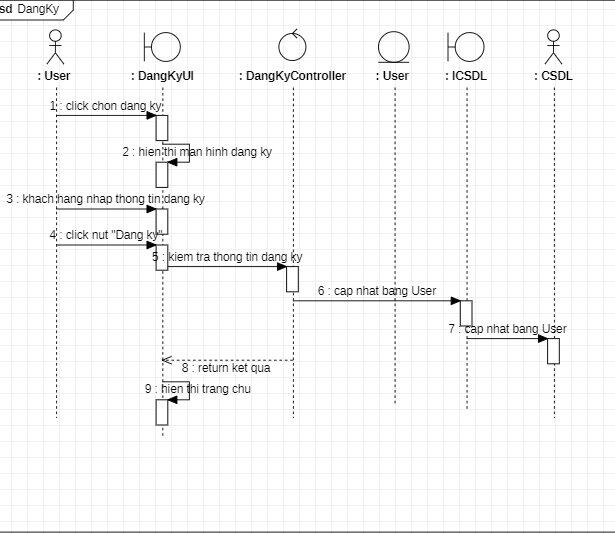
- Hệ thống kiểm tra thông tin đã nhập và hiển thị thông báo đăng ký thành công và ghi thông tin vào bảng USER, use case kết thúc.

* Luồng rẽ nhánh:

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

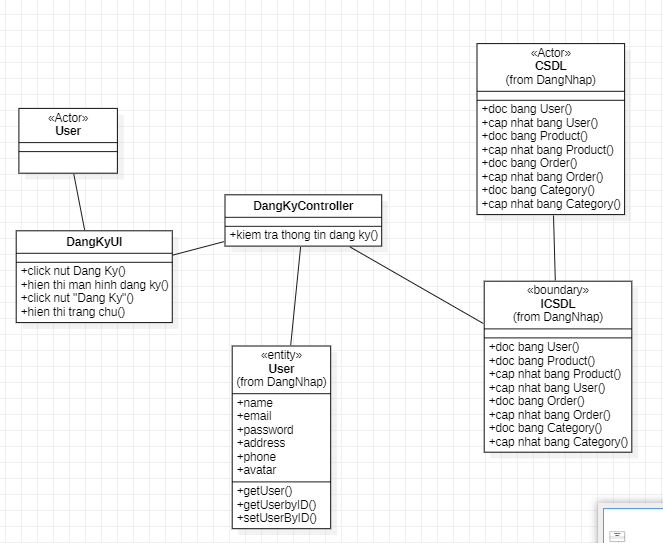
- Mật khẩu và nhập lại mật khẩu không giống nhau: Tại bước 3 của luồng cơ bản, nếu người dùng nhập thông tin mật khẩu và nhập lại mật khẩu không giống nhau hệ thống sẽ yêu cầu người dùng nhập lại

- Người dùng thoát khỏi trang đăng ký, use case sẽ kết thúc.

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Không có
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có
* Dữ liệu liên quan:
* Biểu đồ tuần tự: 

*Hình 16: Biểu đồ tuần tự use case đăng ký(user)*

* Biểu đồ lớp

 *Hình 17: Biểu đồ lớp use case thống kê(user)*

#### Mô tả use case Đăng nhập (khách hàng)

* Use case này cho phép khách hàng đăng nhập vào hệ thống
* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi người dùng kích vào nút đăng nhập. Người dùng sẽ được chuyển đến trang đăng nhập

- Người dùng nhập đầy đủ thông tin yêu cầu và nhấp chọn “Đăng nhập”

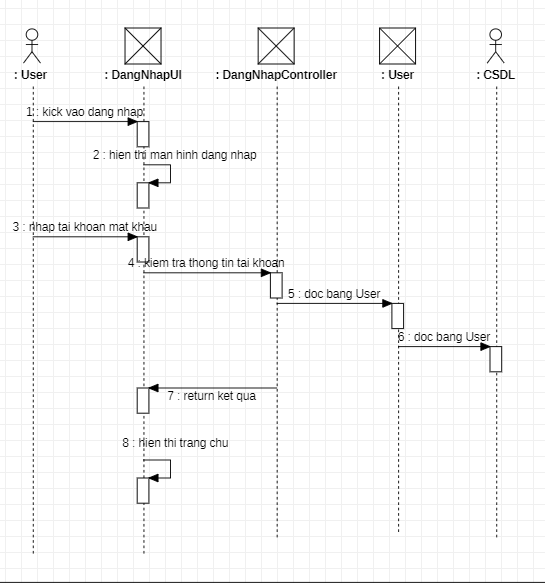
- Hệ thống kiểm tra thông tin người dùng trong bảng USER và chuyển vào giao diện trang web chính, use case kết thúc

* Luồng rẽ nhánh:

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

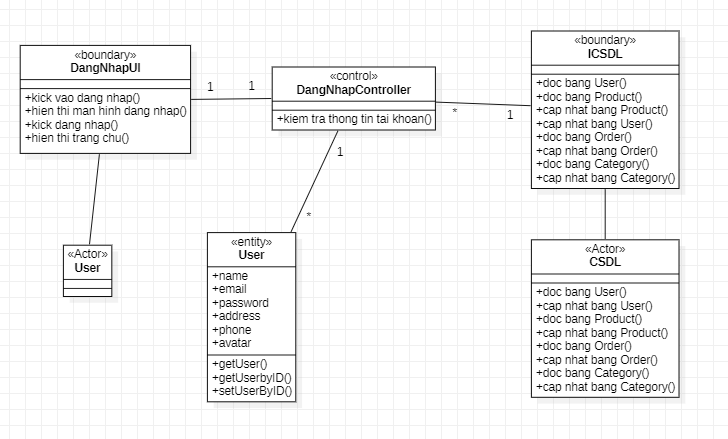
- Tại bước 2 luồng cơ bản, nếu người dùng nhập thông tin không hợp lệ. Hệ thống hiển thị thông báo yêu cầu người dùng nhập lại.

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Không có
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có
* Biểu đồ tuần tự:



*Hình 18: Biểu đồ tuần tự use case đăng nhập(user)*

* Biểu đồ lớp

 *Hình 19: Biểu đồ lớp use case đăng nhập(user)*

#### Use case xem trang chủ

* Use case này cho phép người dùng xem trang chủ của website.
* **Luồng sự kiện:**
* Luồng cơ bản:
  + - Use case bắt đầu khi người kích vào logo của website của ứng dụng, người dùng sẽ được chuyển tới màn hình trang chủ, hệ thống sẽ lấy thông tin từ bảng PRODUCT hiển thị các loại sách, sản phẩm lên màn hình.
    - Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.
* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

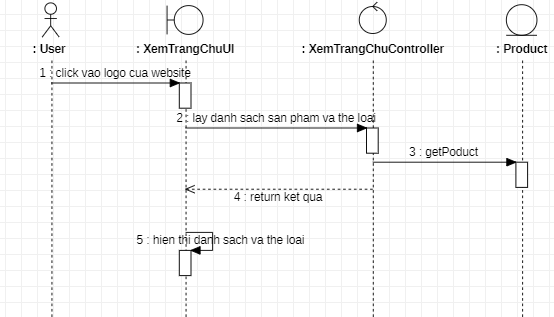
Đã đăng nhập tài khoản vào app.

* **Hậu điều kiện:**

Không có.

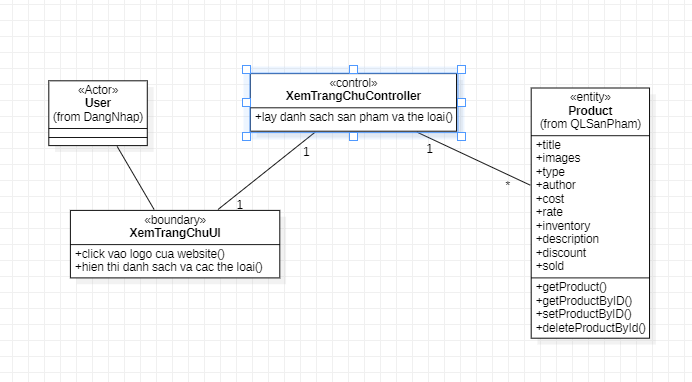
* **Điểm mở rộng:**

Không có.

* Biểu đồ tuần tự

*Hình 20: Biểu đồ tuần tự use case xem trang chủ(user)*

* Biểu đồ lớp



*Hình 20: Biểu đồ lớp use case xem trang chủ(user)*

#### Xem sản phẩm theo loại

Use case này cho phép người dùng xem sản phẩm theo loại

* Luồng sự kiện:
* Luồng cơ bản:

- Use case này bắt đầu khi người dùng nhấn chọn một tên loại sách

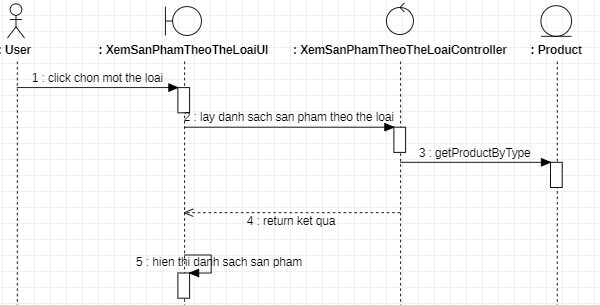
- Hệ thống lấy thông tin từ bảng PRODUCT và hiển thị danh sách lên màn hình theo loại đã chọn

- Use case kết thúc.

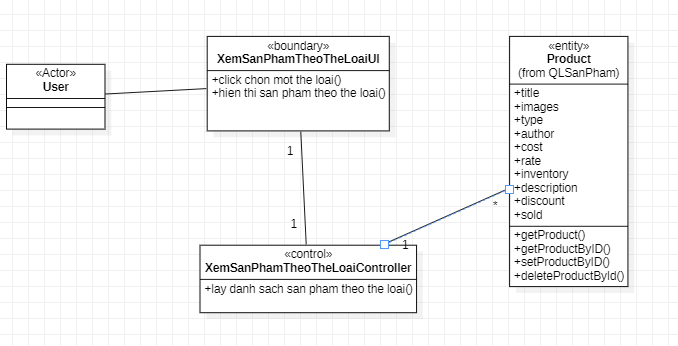
* Luồng rẽ nhánh:

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

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Không có
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có
* Biểu đồ tuần tự:

 *Hình 22: Biểu đồ tuần tự use case xem sản phẩm theo thể loại(user)*

* Biểu đồ lớp:

*Hình 20: Biểu đồ lớp use case xem sản phẩm theo thể loại(user)*

#### Xem thông tin chi tiết sản phẩm

Use case này cho phép người dùng xem thông tin chi tiết về sách

* Luồng sự kiện:
* Luồng cơ bản:

- Use case này bắt đầu khi người dùng chọn vào một quyển sách bất kỳ.

- Hệ thống sẽ lấy dữ liệu từ bảng PRODUCT và hiển thị chi tiết lên màn hình.

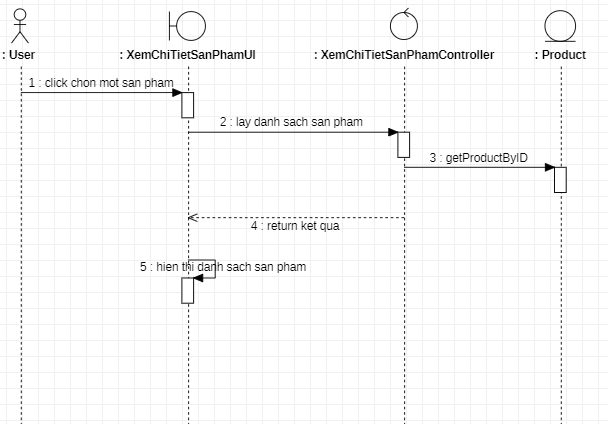
- Use case kết thúc.

* Luồng rẽ nhánh:

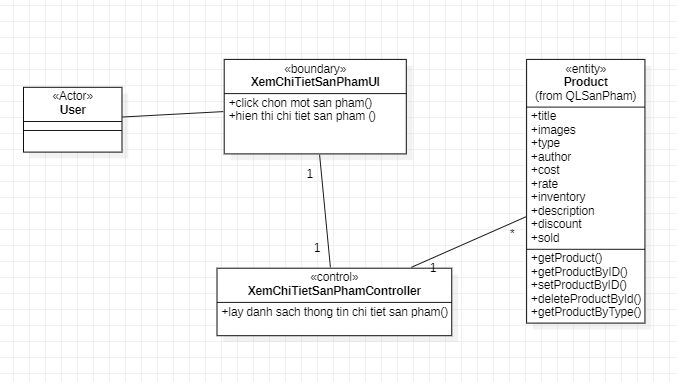
- Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiện thị thông báo lỗi, use case kết thúc.

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

- Biểu đồ tuần tự

 *Hình 22: Biểu đồ tuần tự use case xem chi tiết sản phẩm (user)*

- Biểu đồ lớp

*Hình 23: Biểu đồ lớp use case xem chi tiết sản phẩm (user)*

#### Tìm kiếm sản phẩm

Use case này cho phép người dùng tìm kiếm sản phẩm theo tên

* Luồng sự kiện:
* Luồng cơ bản:

- Use case này bắt đầu khi người dùng nhấn chọn vào thanh tìm kiếm.

- Người dùng nhấn vào ô tìm kiếm

- Người dùng nhập tên muốn tìm và bấm “Tìm kiếm”

- Hệ thống sẽ lấy thông tin từ bảng PRODUCT và hiển thị danh sách lên màn hình theo tên người dùng nhập.

- Use case kết thúc.

* Luồng rẽ nhánh:

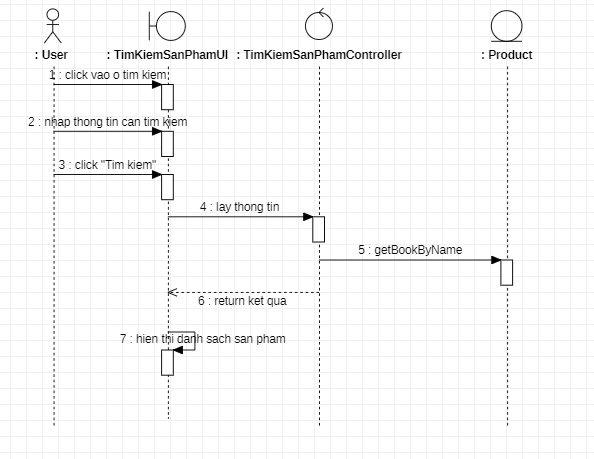
- Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiện thị thông báo lỗi, use case kết thúc.

- Tại bước 3 của luồng cơ bản, nếu người dùng không nhập gì, hệ thống sẽ không tìm kiếm.

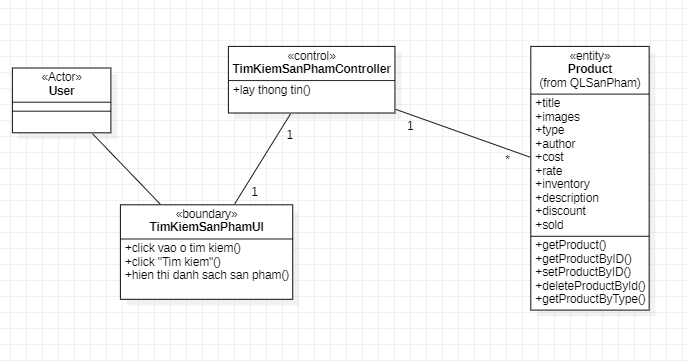
- Người dùng nhấn ra ngoài thanh tìm kiếm, use case kết thúc.

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

-Biểu đồ tuần tự

*Hình 24: Biểu đồ tuần tự use case tìm kiếm sản phẩm (user)*

-Biểu đồ lớp

 *Hình 25: Biểu đồ lớp use case tìm kiếm sản phẩm (user)*

#### Xem, Cập nhật thông tin cá nhân

Use case này cho phép người dùng thay đổi thông tin cá nhân.

* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi người dùng bấm vào “Thông tin người dùng”

- Hệ thống sẽ lấy thông tin từ bảng USER và hiển thị dữ liệu lên màn hình.

- Người dùng chỉnh sửa thống tin

- Người dùng chọn “Cập nhật”, hệ thống sẽ kiểm tra dữ liệu mà người dùng nhập và lưu lại thông tin mới của người dùng vào bảng USER.

- Use case kết thúc.

* Luồng rẽ nhánh:

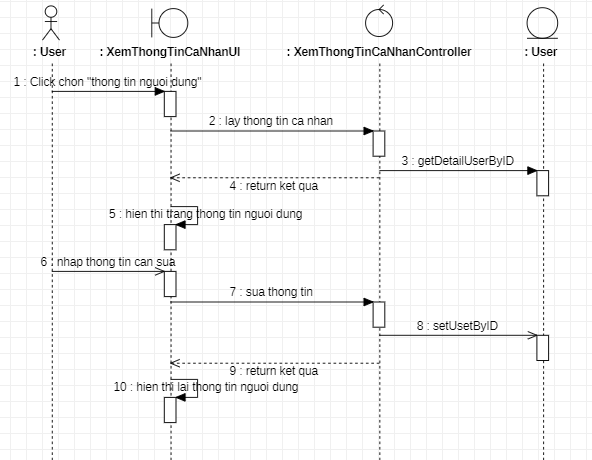
- Tại bất kỳ thời điểm nào trong quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.

- Tại bước 4 của luồng cơ bản, nếu người dùng nhập thông tin sai định dạng, hệ thống sẽ thông báo lỗi, use case kết thúc.

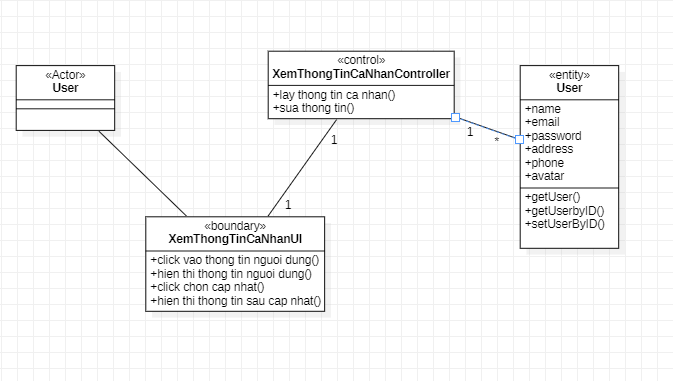
- Người dùng thoát khỏi trang tài khoản, use case kết thúc.

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Đã đăng nhập vào website
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có

- Biểu đồ tuần tự

 *Hình 26: Biểu đồ tuần tự use case cập nhật thông tin cá nhân (user)*

- Biểu đồ lớp

*Hình 27: Biểu đồ lớp use case cập nhật thông tin cá nhân (user)*

#### Đặt hàng

Use case này cho phép người dùng đặt hàng

Luồng sự kiện:

* Luồng cơ bản:

- Use case này bắt đầu khi người dùng click vào “Mua ngay”

- Hệ thống sẽ lấy thông tin sản phẩm và thêm vào bảng ORDER

- Người dùng nhập đầy đủ thông tin như: địa chỉ và số điện thoại và chọn phương thức thanh toán

- Người dùng bấm nút đặt hàng.

- Hệ thống sẽ hiển thị thông báo đặt hàng thành công.

* Luồng rẽ nhánh:

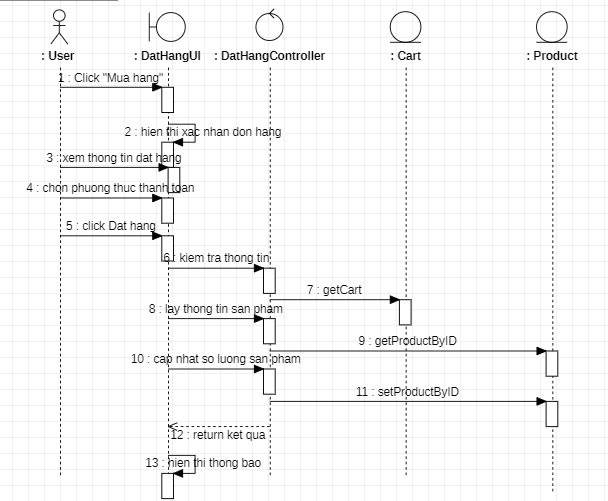
- Tại bất kỳ thời điểm nào trong quá trình thực hện, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.

- Tại bước 3 của luồng cơ bản, nếu người dùng không nhập đầy đủ dữ liệu, hệ thống sẽ hiển thị thông báo lỗi , use case kết thúc.

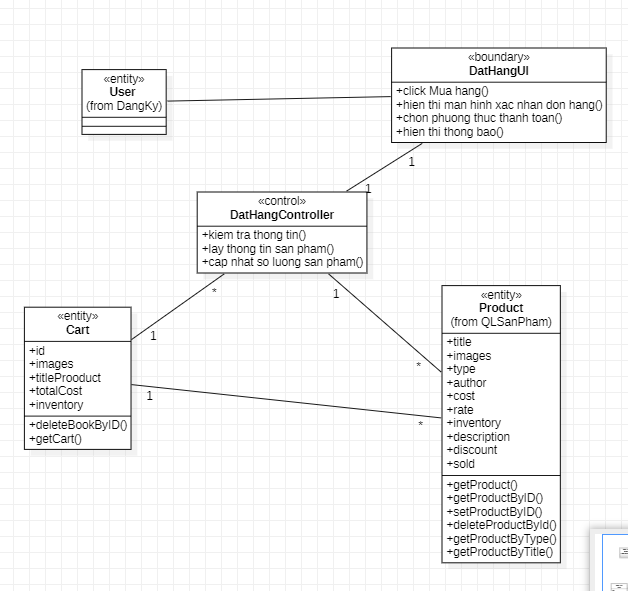
- Người dùng thoát khỏi trang đặt hàng, use case sẽ kết thúc.

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

-Biểu đồ tuần tự

*Hình 28: Biểu đồ tuần tự use case đặt hàng (user)*

-Biểu đồ lớp:

*Hình 29: Biểu đồ tuần tự use case đặt hàng (user)*

#### Kiểm tra đơn hàng

Use case này cho phép người dùng có thể kiểm tra tình trạng đơn hàng

* Luồng sự kiện:
* Luồng cơ bản:

- Use case này bắt đầu khi người dùng bấm vào “Đơn hàng” trong phần cá nhân

- Hệ thống sẽ lấy thông tin từ bảng ORDER và hiển thị dữ liệu lên màn hình.

- Use case kết thúc.

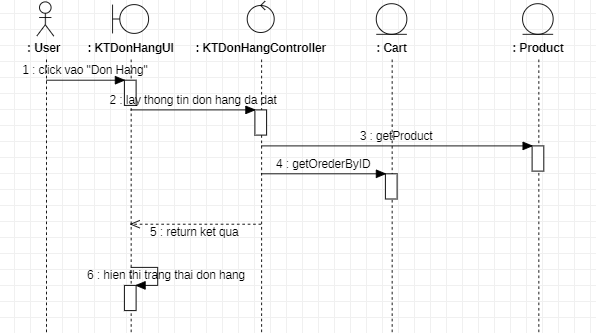
* Luồng rẽ nhánh:

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

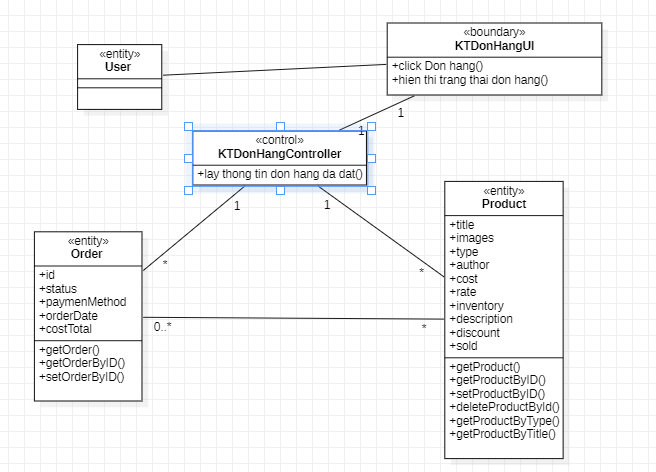
- Người dùng nhấn thoát khỏi trang đơn hàng, use case kết thúc.

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Đã đăng nhập vao website
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có

-Biểu đồ trình tự

 *Hình 30: Biểu đồ tuần tự use case kiểm tra đơn hàng(user)*

-Biểu đồ lớp

*Hình 31: Biểu đồ lớp use case kiểm tra đơn hàng(user)*

#### Xem, Cập nhật giỏ hàng

Use case này cho phép người dùng xem các sản phẩm đã thêm vào giỏ và chỉnh sửa số lượng sản phẩm có trong giỏ hàng.

* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi người dùng nhấn vào biểu tượng giỏ hàng trên màn hình.

- Hệ thống sẽ lấy thông tin từ bảng PRODUCT và hiển thị lên màn hình

- Cập nhật số lượng sản phẩm:

+ Tăng số lượng sản phẩm: người dùng nhấp biểu tượng dấu “+” ở sản phẩm tương ứng, hệ thống sẽ tự động tăng số lượng lên 1.

+ Giảm số lượng mua sản phẩm: người dùng nhấp vào biểu tượng dấu “-“ trên sản phẩm tương ứng, hệ thống sẽ tự động giảm số lượng đi 1.

+ Xóa sản phẩm: người dùng nhấp chọn biểu tượng “x” trên màn hình. Hệ thống sẽ hỏi lại có chắc chắn xóa không. Nếu người dùng ấn “Có” hệ thống sẽ xóa sản phẩm đó ra khỏi bảng CART.

- Use case kết thúc.

* Luồng rẽ nhánh:

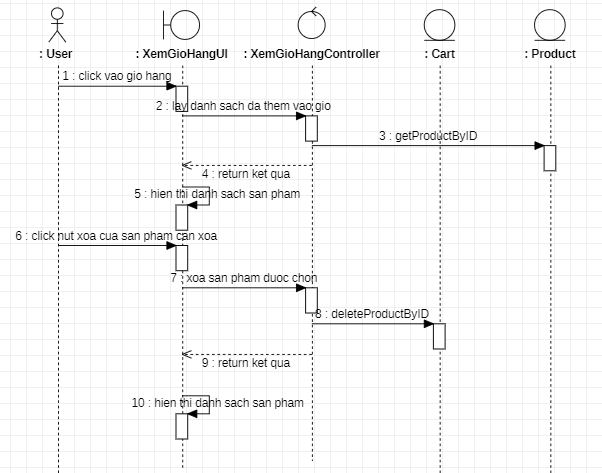
- Tại bước xóa của luồng cơ bản, khi thông báo xác nhận xóa hiển thị trên màn hình, người dùng chọn không thì thao tác xóa được hủy

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

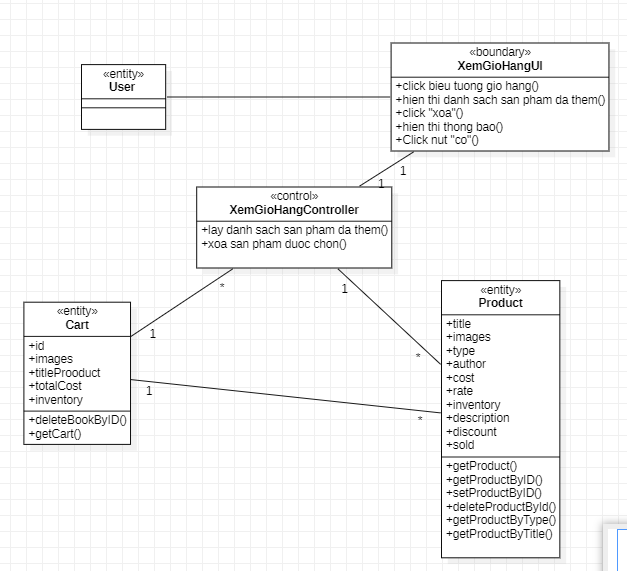
- Người dùng nhấn thoát khỏi trang giỏ hàng, use case kết thúc.

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Đã đăng nhập
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có

-Biểu đồ trình tự

*Hình 34: Biểu đồ tuần tự use case xem cập nhật giỏ hàng(user)*

-Biểu đồ lớp

 *Hình 35: Biểu đồ lớp use case xem cập nhật giỏ hàng(user)*

#### Quên mật khẩu

Use case này cho phép người dùng lấy lại mật khẩu

* Luồng sự kiện:
* Luồng cơ bản:

- Use case bắt đầu khi người dùng nhấn vào “Quên mật khẩu” trên màn hình đăng nhập.

- Người dùng nhập email và bấm vào nút “Tiếp tục”

- Người dùng nhập số điện thoại, mật khẩu mới, xác nhận mật khẩu mới và bấm vào nút xác nhận.

- Hệ thống sẽ hiển thị thay đổi mật khẩu thành công và cập nhật dữ liệu vào bảng USER

- Use case kết thúc.

* Luồng rẽ nhánh:

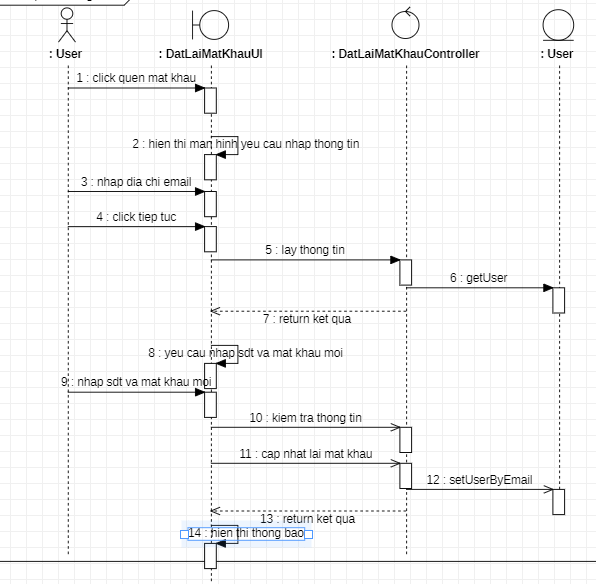
- Tại bước 2 và 3 của luồng cơ bản, nếu người dùng nhập thông tin sai, hệ thống sẽ hiển thị thông báo yêu cầu người dùng nhập lại

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

- Người dùng nhấn thoát khỏi trang, use case kết thúc.

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

-Biểu đồ trình tự

 *Hình 34: Biểu đồ tuần tự use case đặt lại mật khẩu(user)*

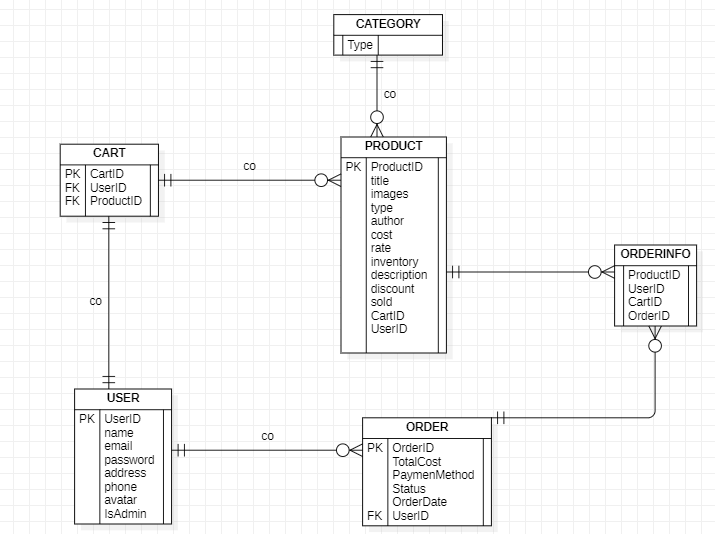
-Biểu đồ lớp

 *Hình 34: Biểu đồ lớp use case đặt lại mật khẩu(user)*

## Thiết kế giao diện và cơ sở dữ liệu

### Thiết kế CSDL

- Biểu đồ thực thể liên kết

 *Hình 35: Biểu đồ thực thể liên kết*

#### Bảng User

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| UserID | String | Khóa chính (PK) | ID khách hàng |
| Email | String |  | Email khách hàng |
| Password | String |  | Mật khẩu đăng nhập |
| Name | String |  | Họ tên khách hàng |
| IsAdmin | String |  | Quyền truy cập |
| Address | String |  | Địa chỉ |
| Phone | String |  | Số điện thoại |
| Avatar | String |  | Hình ảnh đại diện |

Bảng 2. Bảng User

#### Bảng Category

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| CategoryName | String |  | Tên thể loại |

Bảng 2. Bảng Category

#### Bảng Product

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| ProductID | String | Khóa chính (PK) | Mã sách |
| Title | String |  | Tên sách |
| Author | String |  | Tác giả |
| Type | String |  | Thể loại |
| Cost | Int |  | Giá bán |
| Description | String |  | Mô tả sản phẩm |
| Images | String |  | Hình ảnh sản phẩm |
| Rate | Double |  | Đánh giá sản phẩm |
| Page | Int |  | Số lượng trang |
| Sold | Int |  | SL sản phẩm đã bán |
| Inventory | Int |  | Số lượng còn lại |
| Discount | String |  | Ưu đãi |
| CartID | String | Khóa ngoài (PK) | Mã giỏ hàng |
| UserID | String | Khóa ngoài (PK) | Mã người dùng |

Bảng 2. Bảng Book

#### Bảng Order

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| OrderID | String | Khóa chính (PK) | Mã đơn hàng |
| TotalCost | Int |  | Tổng tiền |
| PaymentMethod | String |  | Phương thức thanh toán |
| Status | String |  | Trạng thái đơn hàng |
| OrderDate | String |  | Ngày mua hàng |
| UserID | String | Khóa ngoài (FK) | Mã khách hàng |

Bảng 2. Bảng Order

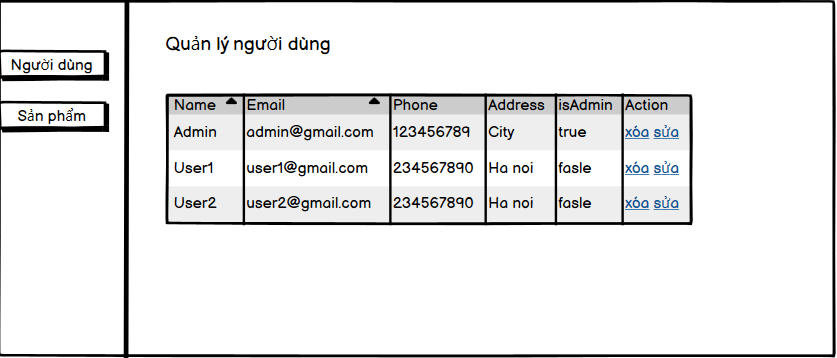
#### Bảng OrderInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| ProductID | String | Khóa ngoài (FK) | Mã sách |
| OrderID | String | Khóa ngoài (FK) | Mã đơn hàng |
| CartID | String | Khóa ngoài (PK) | Mã giỏ hàng |
| UserID | String | Khóa ngoài (FK) | Mã khách hàng |

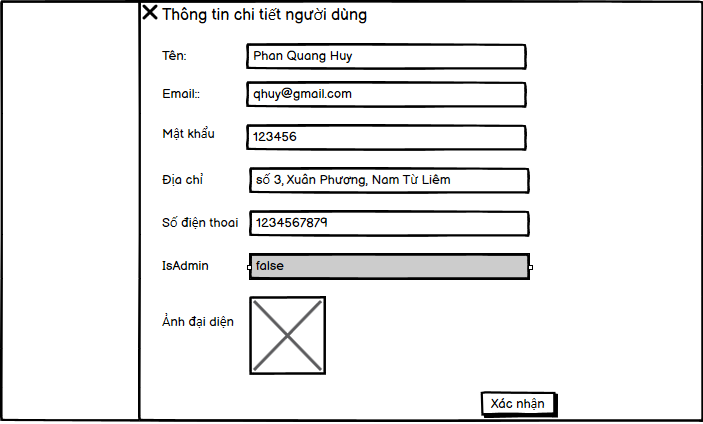
Bảng 2. Bang OrderInfo

### Xây dựng giao diện

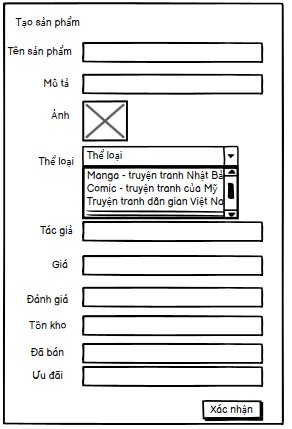
#### Quản trị viên

- Quản lí tài khoản

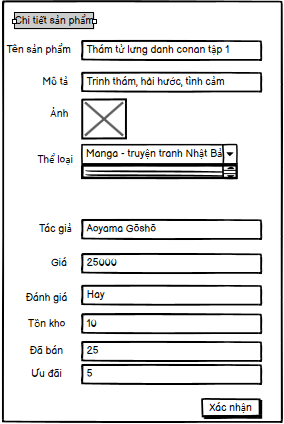
*Hình 36: Giao diện quản lí tài khoản của quản trị viên*

 - Quản lí sản phẩm

*Hình 37: Giao diện quản lí sản phẩm của quản trị viên*



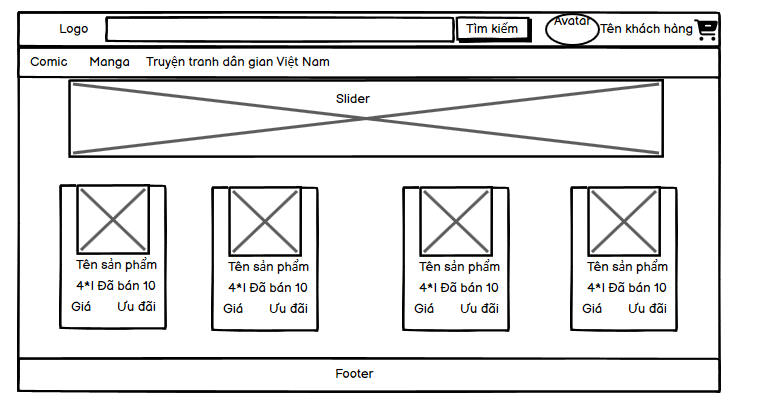
*Hình 38: Giao diện thêm sản phẩm của quản trị viên*



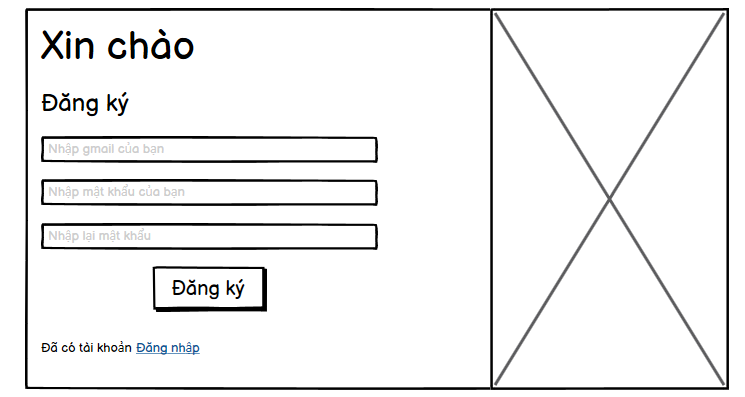
*Hình 39: Giao diện thêm sứa sản phẩm của quản trị viên*

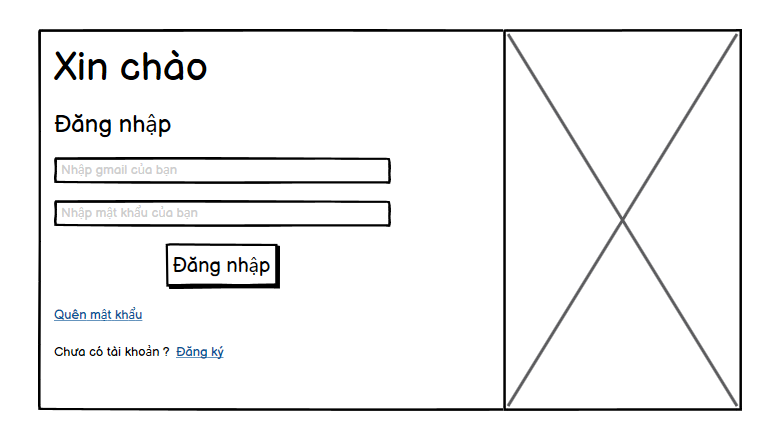
#### Khách hàng

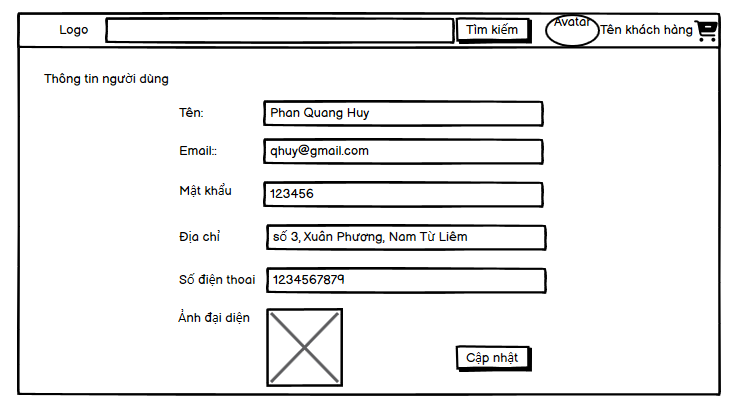
- Trang chủ



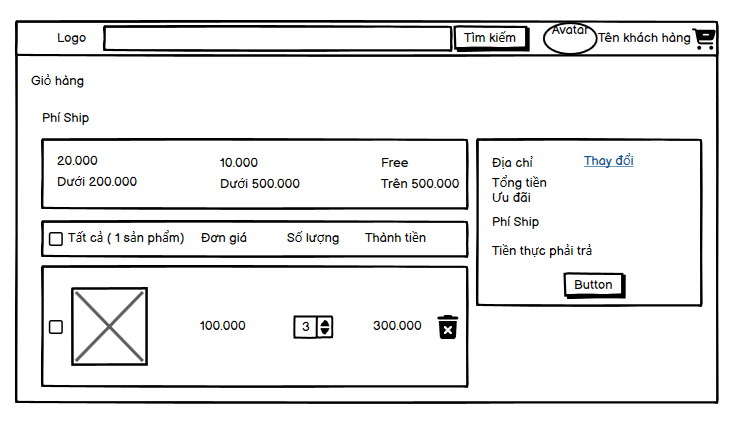
*Hình 40: Giao diện trang chủ*

 *Hình 41: Giao diện đăng ký*

 *Hình 42: Giao diện đăng nhập*

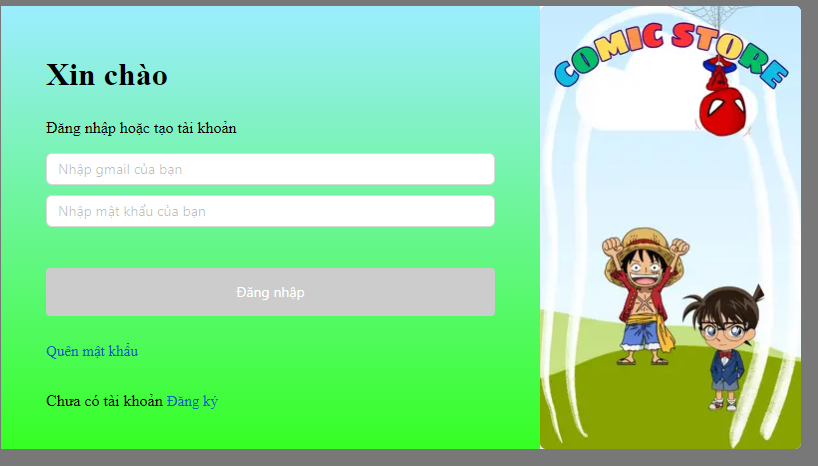


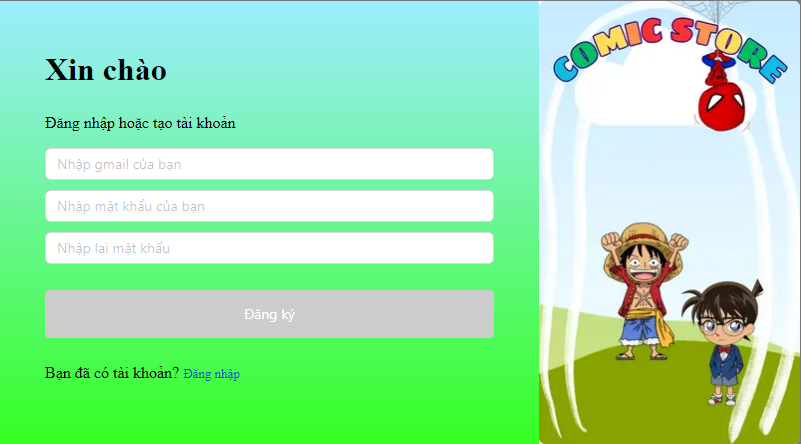
*Hình 43: Giao diện thông tin người dùng*

*Hình 42: Giao diện giỏ hàng*

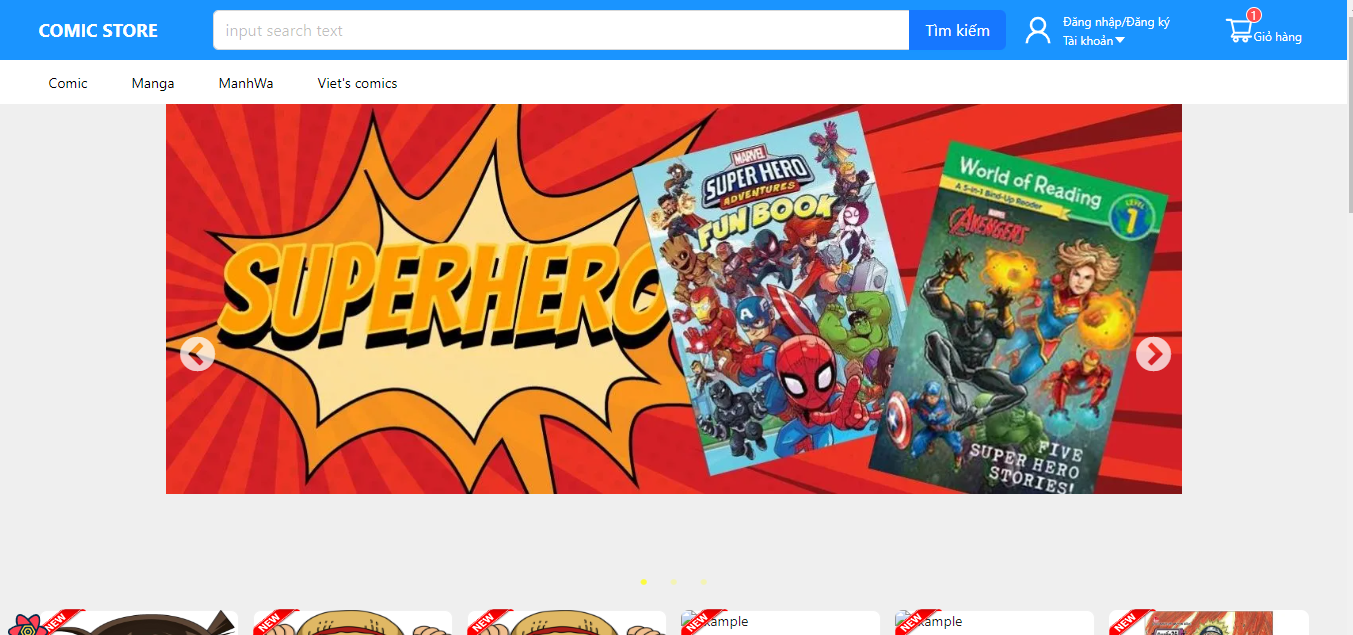
# WEBSITE BÁN SÁCH COMIC STORE

## Khách hàng

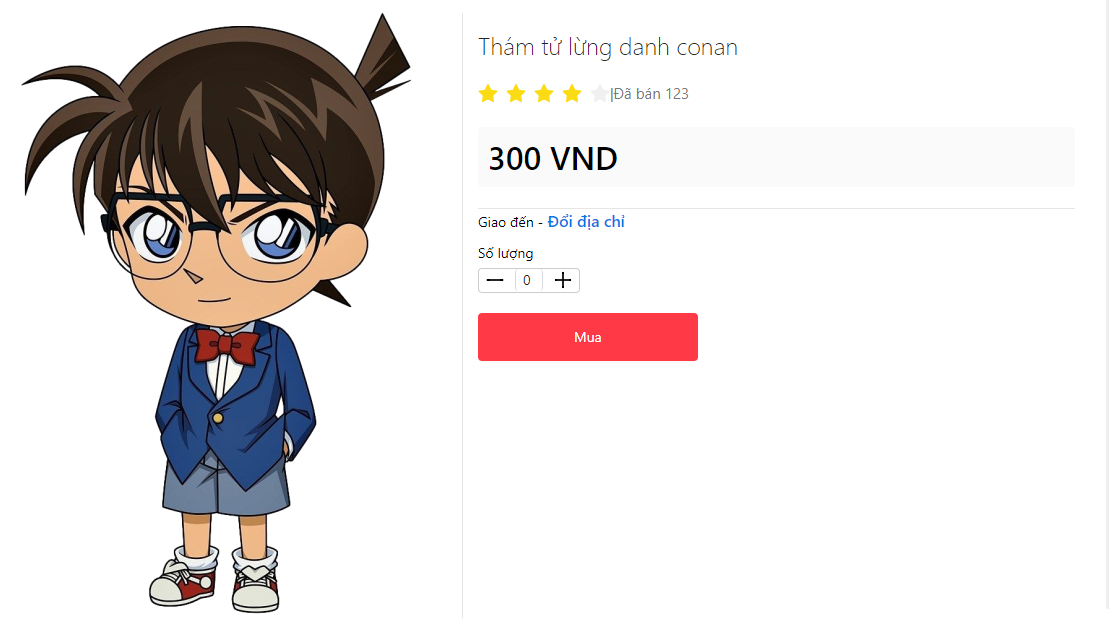
 *Hình 43: Giao diện đăng nhập*



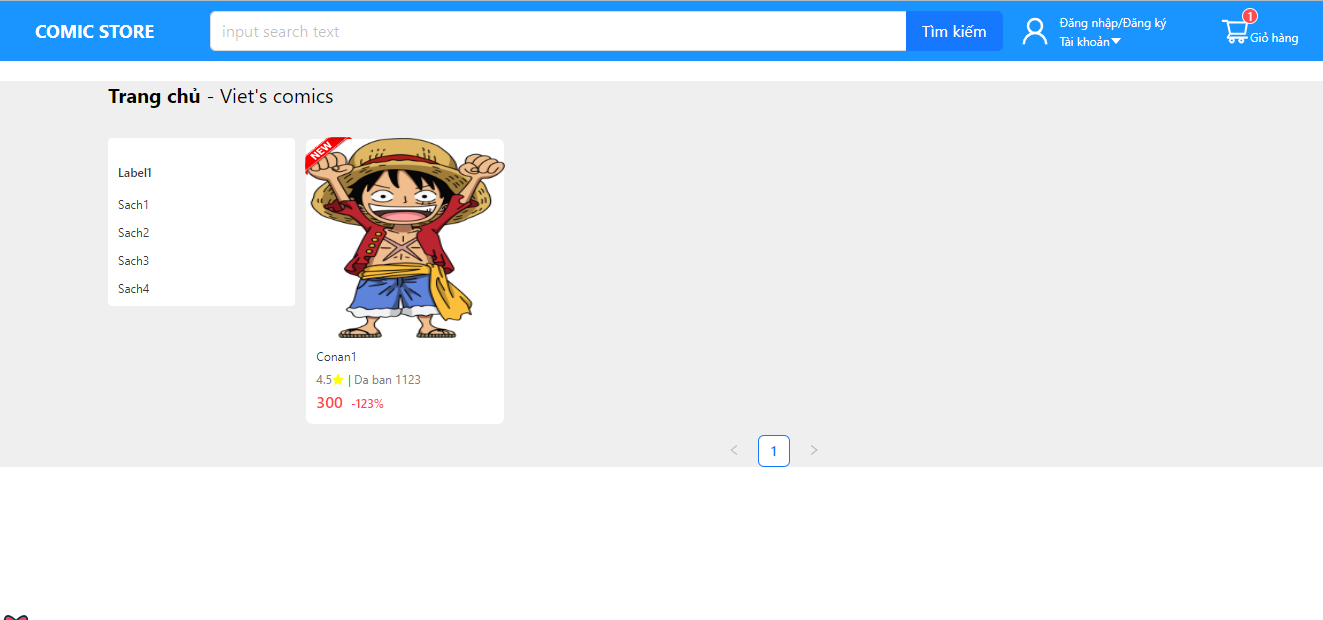
*Hình 44: Giao diện đăng kí*

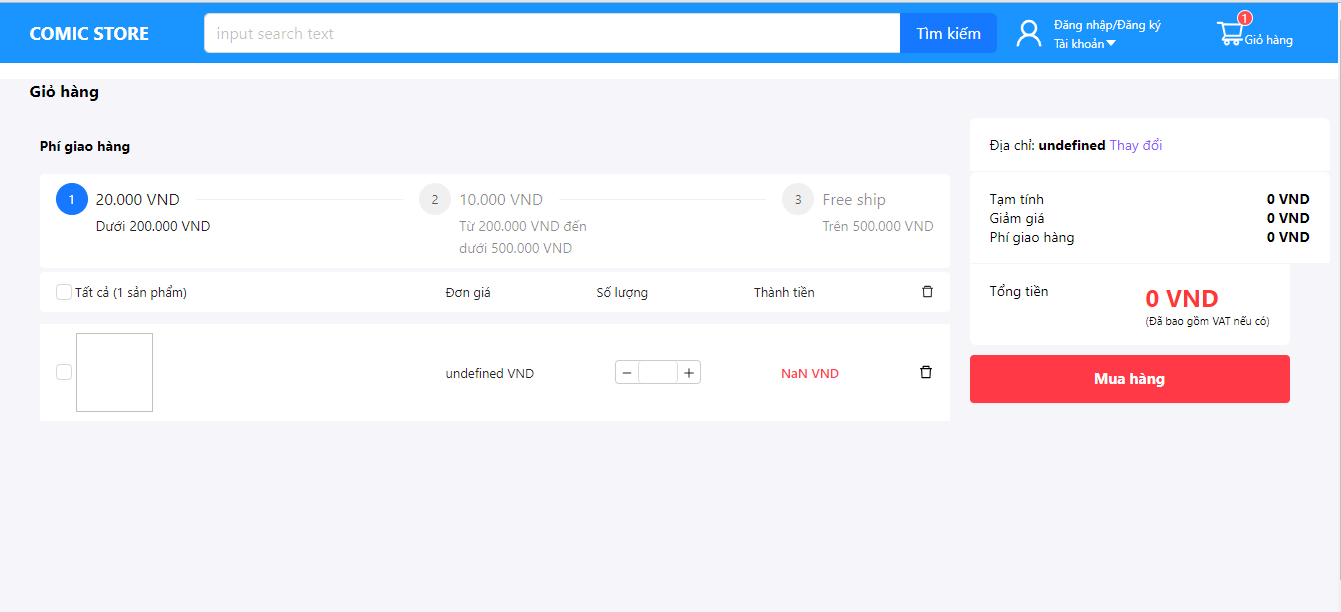


*Hình 45: Giao diện trang chủ*

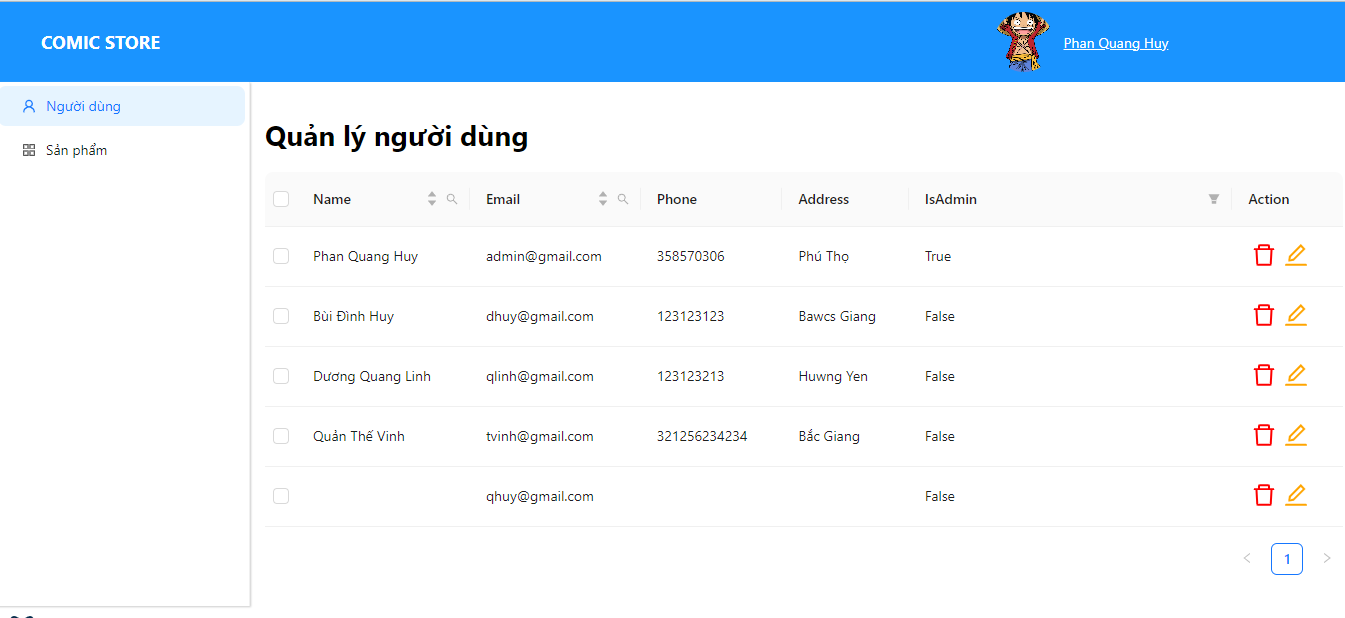


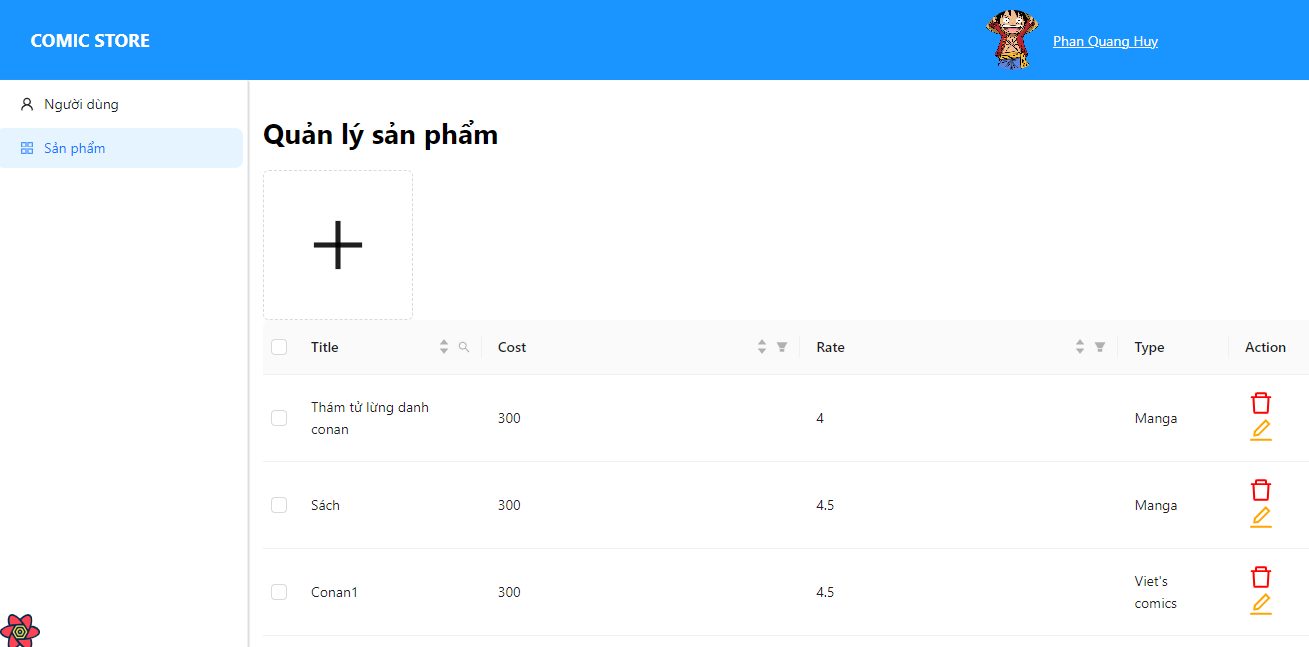
*Hình 46: Giao diện chi tiết sản phẩm*

 *Hình 47: Giao diện xem sản phẩm theo loại*

 *Hình 48: Giao diện giỏ hàng*

## Quản trị viên

 *Hình 49: Giao diện quản lí người dùng*



*Hình 50: Giao diện quản lí sản phẩm*

## Kiểm thử 1 số chức năng

### Đăng nhập khách hàng

|  |  |  |  |
| --- | --- | --- | --- |
| Stt | Phương pháp thực hiện | Kết quả | Trạng thái |
| 1 | Để trắng thông tin tài khoản mật khẩu | Nút đăng ký và đăng nhập sẽ disabled | Đạt |
| 2 | Nhập một email hoặc mật khẩu không chính xác | Hiển thị thông báo nhập sai | Đạt |
| 3 | Nhập email và mật khẩu tồn tại trong CSDL | Hiển thị thông báo đăng nhập thành công, đẩy về trang chủ | Đạt |
| 4 | Truy cập vào trang quản trị | Đẩy lại trang người dùng do không có quyền truy cập | Đạt |

Bảng 3. 1 Kiểm thử đăng nhập khách hàng

### Đặt hàng

|  |  |  |  |
| --- | --- | --- | --- |
| Stt | Phương pháp thực hiện | Kết quả | Trạng thái |
| 1 | Khi chưa đăng nhập | Đẩy về màn hình yêu cầu đăng nhập | Đạt |
| 2 | Sau khi đăng nhập, không nhập đầy đủ thông tin địa chỉ | Đặt hàng không thành công, hiển thị thông báo cho từng dòng | Đạt |
| 3 | Sau khi đăng nhập và nhập đầy đủ thông tin | Đẩy đến trang đặt hàng thành công, lưu thông tin đơn hàng | Đạt |
| 4 | Khi đang xem sản phẩm và bấm mua khi chưa đăng nhập | Chuyển đến phần đăng nhập. Sau khi đăng nhập người dùng vẫn ở lại trang đang xem | Đạt |

Bảng 3. 2 Kiểm thử Đặt hàng

### Các trang yêu cầu đăng nhập mới có thể truy cập

Thông tin cá nhân, đơn hàng của tài khoản, đổi mật khẩu đều cần đăng nhập thì mới có thể truy cập được => đạt yêu cầu

### Đăng nhập quản trị

|  |  |  |  |
| --- | --- | --- | --- |
| Stt | Phương pháp thực hiện | Kết quả | Trạng thái |
| 1 | Để trắng thông tin tài khoản mật khẩu | Nút đăng ký và đăng nhập sẽ disabled | Đạt |
| 2 | Nhập một email hoặc mật khẩu không chính xác | Hiển thị thông báo nhập sai, yêu cầu nhập lại | Đạt |
| 3 | Nhập email và mật khẩu tồn tại trong CSDL | Đẩy về trang theo đúng quyền đã chỉ định sẵn | Đạt |
| 4 | Đăng nhập tài khoản Admin | Admin sẽ được đẩy về Home, và có toàn quyền truy cập trong quản trị | Đạt |

Bảng 3. 3 Kiểm thử đăng nhập quản trị

# KẾT LUẬN

* 1. Kết quả

- Xây dựng được một website bán sách, với các tính năng cơ bản của một website bán hàng

- Hiểu được cách thức hoạt động của một website

- Trong quá trình thực hiện website, học hỏi và tiếp thu được thêm các kiến thức mà trước đó chưa được biết đến

2. Nhược điểm

* Vẫn còn tồn tại một số hạn chế, cũng như một số tính năng mới chỉ dừng lại ở khâu ý tưởng, do kiến thức còn hạn hẹp chưa đủ để thực hiện
* Một số các tính năng của website vẫn chưa được tối ưu như mong muốn

# TÀI LIỆU THAM KHẢO

[1] Jon Duckett, *HTML and CSS design and build Websites*.Năm xuât bản 2011.

[2] <https://200lab.io/blog/>

[3] <https://codelearn.io/>

[5] <https://topdev.vn/blog/>

[6] https://aptech.fpt.edu.vn/

[4] https://www.youtube.com/