**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GTVT**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GTVT**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Trần Thị Thu Huyền**

QUẢN LÝ THƯ VIỆN SỐ TRƯỜNG UTT

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành:** **Mạng máy tính và truyền thông dữ liệu**

**HÀ NỘI – 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GTVT**

**KHOA CÔNG NGHỆ THÔNG TIN**

**Trần Thị Thu Huyền**

QUẢN LÝ THƯ VIỆN SỐ TRƯỜNG UTT

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành:** **Mạng máy tính và truyền thông dữ liệu**

**Cán bộ hướng dẫn: Trần Việt Vương**

**TÓM TẮT**

Ngày nay, với sự phát triển mạnh mẽ của công nghệ thông tin và những ứng dụng của nó trong đời sống. Máy tính điện tử không còn là một thứ phương tiện lạ lẫm đối với mọi người mà nó dần trở thành một công cụ làm việc và giải trí thông dụng và hữu ích của chúng ta, không chỉ ở công sở mà còn ngay cả trong gia đình.

Trong nền kinh tế hiện nay, với xu thế toàn cầu hoá nền kinh tế thế giới, mọi mặt của đời sống xã hội ngày càng được nâng cao, đặc biệt là nhu cầu trao đổi hàng hoá của con người ngày càng tăng cả về số lượng và chất lượng. Thông qua các sản phẩm và công nghệ này, chúng ta dễ dàng nhận ra tầm quan trọng và tính tất yếu của thương mại điện tử. Với những thao tác đơn giản trên máy có nối mạng Internet bạn sẽ có tận tay những gì mình cần mà không phải mất nhiều thời gian. Bạn chỉ cần vào các trang dịch vụ thương mại điện tử, làm theo hướng dẫn và click vào những gì bạn cần. Các nhà dịch vụ sẽ mang đến tận nhà cho bạn.

Hiện nay, để xây dựng một website quản lý thì có rất nhiều ngôn ngữ lập trình có thể đáp ứng, nhưng trong đó, ngôn ngữ lập trình PHP là ngôn ngữ phổ biến nhất với một cộng đồng sử dụng đông đảo. Hơn thế nữa, dựa trên PHP, các framework cũng được xây dựng để hỗ trợ cho việc lập trình website một cách dễ dàng hơn; trong đó Laravel Framework được các lập trình viên PHP sử dụng phổ biến nhất với những tính năng vượt trội do framework này cung cấp.

Do đó, trong đề tài “**QUẢN LÝ THƯ VIỆN SỐ TRƯỜNG UTT**” này, em sẽ nghiên cứu, giới thiệu về Laravel Framework, áp dụng những kiến thức đã tìm hiểu và nghiên cứu được về Laravel Framework để xây dựng một website quản lý thư viện.

**Từ khóa:** PHP, Laravel Framework

**LỜI CAM ĐOAN**

Em xin cam đoan đồ án tốt nghiệp “**QUẢN LÝ THƯ VIỆN SỐ TRƯỜNG UTT**” là công trình nghiên cứu của bản thân. Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, nếu sai em xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

*Hà Nội, ngày.........tháng..........năm 2024*

Sinh viên thực hiện

**Trần Thị Thu Huyền**

**LỜI CẢM ƠN**

Lời đầu tiên, em xin chân thành cảm ơn khoa Công nghệ thông tin, trường Đại học Công nghệ Giao thông vận tải đã tạo điều kiện thuận lợi cho em thực hiện đồ án tốt nghiệp.

Đặc biệt em xin chân thành cảm ơn giảng viên **Trần Việt Vương** – người trực tiếp tận tình hướng dẫn, chỉ bảo em trong suốt thời gian nghiên cứu và phát triển đồ án.

Em cũng xin chân thành cảm ơn tất cả các thầy, các cô Khoa Công nghệ thông tin trường Đại học Công nghệ Giao thông vận tải đã hướng dẫn, trang bị cho em những kiến thức chuyên môn trong suốt quá trình học tập trên ghế nhà trường để em có thể áp dụng những kiến thức đã học vào thực tiễn và tự hoàn thiện mình hơn về cả tri thức lẫn kỹ năng sống.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các thầy giáo, cô giáo và các bạn về những vấn đề đã triển khai để có thể hoàn thiện đồ án hơn nữa.

Em xin trân trọng cảm ơn!

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

*Hà Nội, ngày.........tháng..........năm 2024*

**Giảng viên hướng dẫn**

**Trần Việt Vương**

**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN 1](#_Toc153751948)

1.1. [Lý do chọn đề tài 1](#_Toc153751949)

[1.2. Mục tiêu của đề tài 2](#_Toc153751950)

[1.3. Giới hạn và phạm vi của đề tài 2](#_Toc153751951)

1.4. [Kết quả đạt dược 2](#_Toc153751952)

[1.4.1. Các chức năng của website 2](#_Toc153751953)

[1.4.2. Về vận hành 2](#_Toc153751954)

[1.4.3. Về an toàn bảo mật 3](#_Toc153751955)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG 4](#_Toc153751956)

[2.1. Cơ sở lý thuyết 4](#_Toc153751957)

[2.1.1. Phân tích và thiết kế hướng dối tượng OOAD và UML 4](#_Toc153751958)

[2.1.2. Tổng quan về HTML 6](#_Toc153751959)

[2.1.3. Tổng quan về CSS 7](#_Toc153751960)

[2.1.4. Tổng quan về Javascript và Jquery 8](#_Toc153751961)

[2.1.5. Tổng quan về Bootstrap 9](#_Toc153751962)

[2.1.6. Tổng quan về PHP và Framework Laravel 9](#_Toc153751963)

[2.1.7. Hệ quản trị cơ sở dữ liệu MySQL 12](#_Toc153751964)

2.2. [Công cụ sử dụng 16](#_Toc153751965)

[2.2.1. XAMPP 16](#_Toc153751966)

[2.2.2. Sublime Text 17](#_Toc153751967)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 19](#_Toc153751968)

[3.1. Khảo sát và đặc tả nghiệp vụ 19](#_Toc153751969)

[3.1.1. Giới thiệu đơn vị khảo sát 19](#_Toc153751970)

[3.1.2. Nội dung khảo sát 19](#_Toc153751971)

[3.1.3. Hiện trạng khảo sát 20](#_Toc153751972)

[3.2. Yêu cầu xây dựng hệ thống 21](#_Toc153751973)

[3.3. Mô tả bài toán 21](#_Toc153751974)

[3.4. Phân tích thiết kế 22](#_Toc153751975)

[3.4.1. Xác định các tác nhân (Actor) và chức năng (UseCase) 22](#_Toc153751976)

[3.4.2. Biểu đồ UseCase 23](#_Toc153751977)

[3.4.3. Biểu đồ tuần tự 48](#_Toc153751978)

[3.4.4. Biểu đồ hoạt động 54](#_Toc153751979)

[3.4.5. Biểu đồ Class 59](#_Toc153751980)

[CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH 60](#_Toc153751981)

4.1. [Xây dựng cơ sở dữ liệu 60](#_Toc153751982)

[4.2. Xây dựng giao diện và chức năng hệ thống 63](#_Toc153751983)

[4.2.1. Giao diện người dùng 63](#_Toc153751984)

[4.2.2. Giao diện và chức năng quản trị 69](#_Toc153751985)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 80](#_Toc153751986)

[TÀI LIỆU THAM KHẢO 82](#_Toc153751987)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên viết tắt** | **Tên đầy đủ** |
| 1 | PHP | Personal Home Page |
| 2 | HTML | HyperText Markup Language |
| 3 | CSS | Cascading Style Sheets |
| 4 | JS | JavaScript |
| 5 | OSI | Open Systems Interconnection |
| 6 | MVC | Model - View - Controller |
| 9 | OOAD | Object Oriented Analysis and Design |
| 10 | UML | Unified Modeling Language |
| 11 | SVG | Scalable Vector Graphics |
| 12 | XML | Extensible Markup Language |
| 13 | IE | Internet Explorer |
| 14 | ODBC | Open Database Connectivity |
| 15 | DBMS | Database Management System |

**CHƯƠNG 1. TỔNG QUAN**

## **Lý do chọn đề tài**

Ngày nay, với sự phát triển không ngừng của công nghệ thông tin đã được ứng dụng trong nhiều lĩnh vực của đời sống. Nhờ có công nghệ thông tin mà kinh tế có thể phát triển và đời sống con người được nâng cao, xã hội ngày càng tân tiến. Việc tận dụng ngàng công nghệ thông tin vào mọi ngành nghề đã không là một việc quá xa lạ, đặc biệt là giúp cho việc quản lý trở lên dễ dàng. Vì vậy việc ứng dụng công nghệ thông tin vào việc quản lý, dường như trở thành việc rất quen thuộc đối với các doanh nghiệp, nhà trường.

Do đó ứng dụng của công nghệ thông tin vào việc quản lý dường như không còn xa lạ với các doanh nghiệp. Lợi ích mà các phần mềm quản lý mang lại khiến ta không thể thừa nhận tín hiệu quả của nó mà phần mềm quản lý là một trong số đó. Khi quản lý đòi hỏi sự chính xác tuyệt đói trong tính toán, cập nhật thông tin một cách hiệu quả cao trong công việc.

Với mong muốn tiếp cận và đóng góp sự phổ biến của thương mại điện tử ở Việt Nam, em đã tìm hiểu và thiết kế website quản lý thư viện với nhiều ưu điểm vượt trội so với việc quản lý thông thường việc áp dụng diễn ra một cách nhanh chóng, tiết kiệm đã trở nên vô cùng cấp thiết.

Là sinh viên được trang bị những kiến thức của ngành công nghệ thông tin với những kiến thức đã tiếp thu và vận dụng lý thuyết đó vào công việc thực tế nên em đã chọn đề tài “**QUẢN LÝ THƯ VIỆN SỐ TRƯỜNG UTT**” để thực hiện đồ án tốt nghiệp của mình với mục đích nghiên cứu và xây dựng một website bán hàng chuyên nghiệp giúp đẩy mạnh công tác bán hàng online, mang lại sự tiện lợi cho cả người bán và người mua.

## **Mục tiêu của đề tài**

* Khảo sát, tìm hiểu được thêm kiến thức, các vấn đề liên quan, nắm bắt được quy trình kinh doanh, vận hành và hoạt động của các website quản lý thư viện. .
* Khảo sát thực trạng và nhu cầu mượn/trả của người dùng hiện nay.
* Trình bày được các vấn đề liên quan đến ngôn ngữ PHP, cơ sở dữ liệu MySQL – XAMPP.
* Phân tích hệ thống chương trình bao gồm các giao diện, các chức năng nhằm đáp ứng nhu cầu người dùng.

## **Giới hạn và phạm vi của đề tài**

* Xây dựng website dựa trên các ngôn ngữ lập trình PHP (HTML/CSS/JS) với framework Laravel mô hình MVC và xây dựng dữ liệu qua hệ quản trị cơ sở dữ liệu MySQL.
* Chương trình tập chung xây dựng các chức năng cơ bản của một website quản lý thư viện.
* Xây dựng chương trình dựa trên các thiết kế.

## **Kết quả đạt dược**

### **Các chức năng của website**

Xây dựng website gồm các phần chính:

* **Người dùng:** Giao diện người dùng, xem thông tin sản phẩm (chi tiết v.v.), xem các bài viết tin tức giới thiệu, tìm kiếm, đăng ký, đăng nhập thành viên...…
* **Nhân viên:** Giao diện trang Quản trị viên, quản lý sản phẩm, bài viết tin tức, giới thiệu với các chức năng cơ bản (thêm, sửa, xóa), quản lý tài khoản khách hàng, quản lý các danh mục sản phẩm, bài viết, thống kê sản phẩm, …

### **Về vận hành**

* Website luôn được đảm bảo vận hành một cách chính xác, 24/24.
* Các thao tác, giao diện dễ sử dụng. Có cơ chế cho việc quản lý chỉnh sửa thông tin dễ dàng, mọi lúc mọi nơi.
* Việc nâng cấp có thể thực hiện nhưng không gây mất mát thông tin cũng như đảm bảo website vẫn hoạt động bình thường.

### **Về an toàn bảo mật**

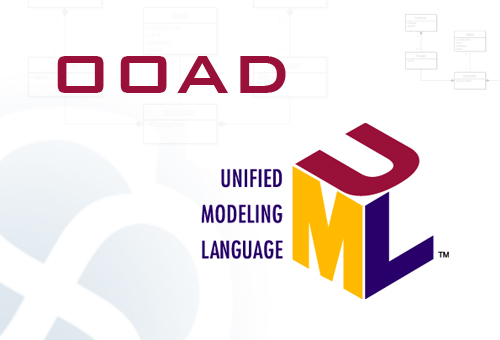
* Việc cập nhật thông tin nhân viên chỉ người quản trị, quản lý được phép truy cập.

# **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG**

## **Cơ sở lý thuyết**

### **Phân tích và thiết kế hướng dối tượng OOAD và UML**

OOAD cần các bản vẽ để mô tả hệ thống được thiết kế, còn UML là ngôn ngữ mô tả các bản vẽ nên cần nội dung thể hiện.  Do vậy, chúng ta phân tích và thiết kế theo hướng đối tượng và sử dụng UML để biểu diễn các thiết kế đó nên chúng thường đi đôi với nhau.



Hình 2.1: OOAD và UML

**Phân tích và thiết kế hướng đối tượng (Object-Oriented Analysis and Design - OOAD)** là một kỹ thuật tiếp cận phổ biến dùng để phân tích, thiết kế một ứng dụng, hệ thống. Nó dựa trên bộ các nguyên tắc chung, đó là một tập các hướng dẫn để giúp người dùng tránh khỏi một thiết kế xấu. Nó giúp phân tích xem hệ thống gồm những đối tượng nào và chúng tương tác, liên kết với nhau như thế nào, từ việc mô tả được tất cả các đối tượng và sự tương tác của chúng sẽ giúp người dùng hiểu rõ về hệ thống và cài đặt được.

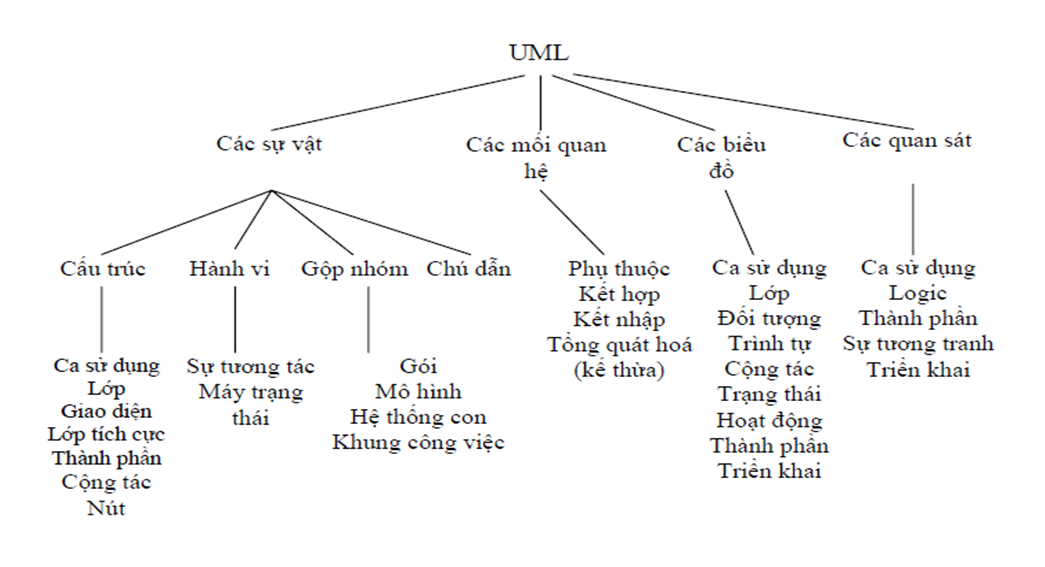
* Ưu điểm: gần gũi với thế giới thực, dễ tái sử dụng, có thể thừa kế từ đó làm giảm chi phí và tính mở cao, hệ thống đáng tin cậy hơn.
* Nhược điểm: phức tạp, khó theo dõi luồng dữ liệu hơn phân tích thiết kế hệ thống hướng chức năng.
* Trong kỹ nghệ phần mềm để sản xuất được một sản phẩm phần mềm người ta chia quá trình phát triển sản phẩm ra nhiều giai đoạn như thu thập và phân tích yêu cầu, phân tích và thiết kế hệ thống, phát triển (coding), kiểm thử, triển khai và bảo trì. Trong đó, giai đoạn phân tích, thiết kế bao giờ cũng là giai đoạn khó khăn và phức tạp nhất. Giai đoạn này giúp chúng ta hiểu rõ yêu cầu đặt ra, xác định giải pháp, mô tả chi tiết giải pháp.

**Ngôn ngữ mô hình hóa thống nhất** (tiếng Anh: Unified Modeling Language, viết tắt thành **UML**) là một ngôn ngữ mô hình gồm các ký hiệu đồ họa mà các phương pháp hướng đối tượng sử dụng để thiết kế các hệ thống thông tin một cách nhanh chóng.

Mục đích UML cung cấp cho người dùng một ngôn ngữ mô hình hoá trực quan sẵn sàng để dùng và có ý nghĩa:

* Cho phép phát triển và trao đổi những mô hình mang nhiều ý nghĩa.
* Cung cấp khả năng mở rộng và chuyên môn hoá để mở rộng những khái niệm cốt lõi.
* Độc lập với ngôn ngữ lập trình chuyên biệt và các tiến trình phát triển.
* Cung cấp nền tảng về sự hiểu biết ngôn ngữ mô hình hoá.
* Khuyến khích và hỗ trợ sự phát triển của các công cụ hướng đối tượng.
* Hỗ trợ những khái niệm phát triển cấp độ cao như collaboration, framework, pattern và component.
* Tích hợp một cách tốt nhất với thực tiễn.

UML sử dụng để vẽ cho nhiều lĩnh vực khác nhau như phần mềm, cơ khí, xây dựng v.v… trong phạm vi các bài viết này chúng ta chỉ nghiên cứu cách sử dụng UML cho phân tích và thiết kế hướng đối tượng trong ngành phần mềm. OOAD sử dụng UML bao gồm các thành phần sau:



Hình 2.2: Các phần tử của UML

### **Tổng quan về HTML**

HTML (HyperText Markup Language – ngôn ngữ siêu văn bản) là một trong các loại ngôn ngữ được sử dụng trong lập trình web. HTML giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes, …



Hình 2.3: Ngôn ngữ HTML

HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được và chỉ có thể được sử dụng để bố cục và định dạng trang web. Ưu điểm nổi trội nhât và cũng là thế mạnh của HTML là khả năng xây dựng cấu trúc và khiến trang web đi vào quy củ một hệ thống hoàn chỉnh.

**Ưu điểm của HTML:**

* Có nhiều tài nguyên hỗ trợ với cộng đồng người dùng vô cùng lớn
* Có thể hoạt động mượt mà trên hầu hết mọi trình duyệt hiện nay
* Học HTML khá đơn giản
* Các markup sử dụng trong HTML thường ngắn gọn, có độ đồng nhất cao
* Sử dụng mã nguồn mở, hoàn toàn miễn phí
* HTML là chuẩn web được vận hành bởi W3C
* Dễ dàng để tích hợp với các loại ngôn ngữ backend (ví dụ như: PHP, Node.js, …)

**Nhược điểm của HTML:**

* Chỉ được áp dụng chủ yếu cho web tĩnh. Nếu muốn tạo các tính năng động, lập trình viên phải dùng thêm JavaScript hoặc ngôn ngữ backend của bên thứ 3 (ví dụ như: PHP).
* Mỗi trang **HTML** cần được tạo riêng biệt, ngay có khi có nhiều yếu tố trùng lặp như header, footer.
* Khó để kiểm soát cách đọc và hiển thị file HTML của trình duyệt (ví dụ, một số trình duyệt cũ không render được tag mới. Do đó, dù trong HTML document có sử dụng các tag này thì trình duyệt cũng không đọc được).
* Một vài trình duyệt còn chậm cập nhật để hỗ trợ tính năng mới của HTML.

### **Tổng quan về CSS**

CSS là viết tắt của Cascading Style Sheets. Đây là một ngôn ngữ style sheet được sử dụng để mô tả giao diện và định dạng của một tài liệu viết bằng ngôn ngữ đánh dấu (markup), cung cấp một tính năng bổ sung cho HTML. CSS thường được sử dụng với HTML để thay đổi phong cách của trang web và giao diện người dùng. Nó cũng có thể được sử dụng với bất kỳ loại tài liệu XML nào bao gồm cả XML đơn giản, SVG và XUL.



Hình 2.4: Ngôn ngữ lập trình CSS

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó. Nghĩa là nó sẽ được khai báo bằng vùng chọn, sau đó các thuộc tính và giá trị sẽ nằm bên trong cặp dấu ngoặc nhọn {}. Mỗi thuộc tính sẽ luôn có một giá trị riêng, giá trị có thể là dạng số, hoặc các tên giá trị trong danh sách có sẵn của CSS. Phần giá trị và thuộc tính phải được cách nhau bằng dấu hai chấm, và mỗi một dòng khai báo thuộc tính sẽ luôn có dấu chấm phẩy ở cuối. Một vùng chọn có thể sử dụng không giới hạn thuộc tính.

Trước đây khi chưa có CSS, những người thiết kế web phải trộn lẫn giữa các thành phần trình bày và nội dung với nhau. Nhưng với sự xuất hiện của CSS, người ta có thể tách rời hoàn toàn phần trình bày và nội dung. Giúp cho phần code của trang web cũng gọn hơn và quan trọng hơn cả là dễ chỉnh sửa hơn.

### **Tổng quan về Javascript và Jquery**

JavaScript là một ngôn ngữ lập trình được sử dụng để tạo ra những trang web tương tác, được tích hợp và nhúng trong HTML. JavaScript cho phép kiểm soát các hành vi của trang web tốt hơn so với khi chỉ có một mình HTML. JavaScript kết hợp vào HTML, chạy trên Windows, Macintosh và các hệ thống hỗ trợ Netscape khác.

JQuery là một thư viện JavaScript nhanh, nhỏ, đa nền tảng và giàu tính năng. JQuery được thiết kế để đơn giản hóa client-side scripting của HTML. Nó thực hiện những nhiệm vụ như HTML document traversal và manipulation, animation, event handling, AJAX một cách rất đơn giản bằng một API dễ sử dụng và có khả năng hoạt động trên nhiều loại trình duyệt khác nhau. Mục đích chính của JQuery là làm việc sử dụng JavaScript trên trang web trở nên dễ dàng hơn, giúp website có tính tương tác và hấp dẫn hơn.

### **Tổng quan về Bootstrap**

Bootstrap là một framework HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo một chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet, ...

Bootstrap bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm nhiều Component, Javascript hỗ trợ cho việc thiết kế reponsive của bạn dễ dàng, thuận tiện và nhanh chóng hơn.

Bootstrap là một trong những framework được sử dụng nhiều nhất trên thế giới để xây dựng một website. Bootstrap đã xây dựng lên một chuẩn riêng và rất được người dùng ưa chuộng. Chính vì thế, người dùng hay nghe tới một cụm từ rất thông dụng "Thiết kế theo chuẩn Bootstrap".

Từ cái "chuẩn mực" này, người dùng có thể thấy rõ được những điểm thuận lợi khi sử dụng Bootstrap:

* Rất dễ để sử dụng: Bất kỳ ai có kiến thức cơ bản về HTML và CSS đều có thể bắt đầu sử dụng Bootstrap.
* Responsive: Bootstrap xây dựng sẵn responsive css trên các thiết bị điện thoại, máy tính bảng và máy tính để bàn... Tính năng này khiến cho người dùng tiết kiệm được rất nhiều thời gian trong việc tạo ra một website thân thiện với các thiết bị điện tử, thiết bị cầm tay.
* Tương thích với trình duyệt: Bootstrap tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera). Tuy nhiên, với IE browser, Bootstrap chỉ hỗ trợ từ IE9 trở lên. Điều này vô cùng dễ hiểu vì IE8 không hỗ trợ HTML5 và CSS3.

### **Tổng quan về PHP** **và Framework Laravel**

PHP là viết tắt của chữ "Hypertext Preprocessor", đây là một ngôn ngữ lập trình được sử dụng để viết ở phía máy chủ (lập trình web). Hiện nay có rất nhiều ngôn ngữ có thể viết ở máy chủ như C#, Java, Python và PHP chính là một trong những ngôn ngữ của nhóm này.



Hình 2.5: Ngôn ngữ lập trình PHP

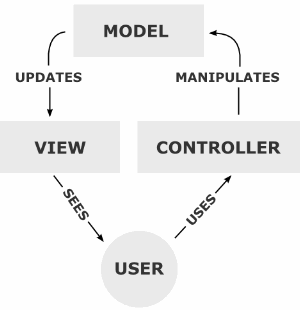
PHP có cú pháp đơn giản, tốc độ nhanh và nhỏ gọn, dễ học và thời gian để tiếp cận PHP ngắn hơn so với các ngôn ngữ lập trình khác nên hiện nay PHP đang chiếm số lương lập trình viên tương đối lớn, có thể gọi là lớn nhất. Hiện PHP có rất nhiều các CMS mã nguồn mở như WordPress, OpenCart nên nó rất đa dạng và đáp ứng hầu hết các website thông thường từ blog cá nhân cho đến website giới thiệu công ty và bán hàng.

Laravel là một nền tảng ứng dụng web (web application framework) nguồn mở dựa trên nền tảng MVC (Model - View - Controller) được dùng để xây dựng các ứng dụng web động tương tác với PHP. Nó cho phép các nhà phát triển xây dựng một ứng dụng web nhanh hơn - so với việc viết mã hỗn tạp - bằng cách cung cấp một bộ thư viện đầy đủ cho các tác vụ thông thường, cũng như cung cấp một mô hình tương tác đơn giản và dễ hiểu cho việc kết nối tới những bộ thư viện đó. Những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu - rõ ràng, một hệ thống đóng gói Modular và quản lý gói phụ thuộc, nhiều cách khác nhau để truy cập vào các cơ sở dữ liệu quan hệ, nhiều tiện ích khác nhau hỗ trợ việc triển khai vào bảo trì ứng dụng.



Hình 2.6: Framwork Laravel

**MVC:** Mô hình MVC là một kiến trúc phần mềm, được tạo ra với mục đích quản lý và xây dựng dự án phần mềm có hệ thống hơn. Mô hình MVC được dùng khá rộng rãi và đặc biệt là trong các ngôn ngữ lập trình Web. Mô hình MVC chia một ứng dụng ra ba thành phần khác nhau: Model, View và Controller. Với việc sử dụng mô hình MVC giúp cho mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần còn lại.

* ***Model:*** Đây là thành phần chứa tất cả các nghiệp vụ Logic, phương thức xử lý, truy xuất Database, đối tượng mô tả dữ liệu như các Class, hàm xử lý, v.v.
* ***View:*** Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như Textbox, Images, v.v. Hiểu một cách đơn giản, nó là tập hợp các Form hoặc các file HTML.
* ***Controller:*** Giữ nhiệm vụ tiếp nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng.

Hình 2.7: Mô hình MVC

### **Hệ quản trị cơ sở dữ liệu MySQL**

#### 2.1.7.1. Khái niệm

**Cơ sở dữ liệu** ( **Database** ) là một tập hợp các [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u_(m%C3%A1y_t%C3%ADnh)" \o "Dữ liệu (máy tính)) có tổ chức, thường được lưu trữ và truy cập điện tử từ hệ thống máy tính. Khi cơ sở dữ liệu phức tạp hơn, chúng thường được phát triển bằng cách sử dụng các kỹ thuật thiết kế và mô hình hóa chính thức.

**Hệ quản trị cơ sở dữ liệu** ( **Database Management System – DBMS** ) là phần mềm hay hệ thống được thiết kế để quản trị một cơ sở dữ liệu.Cụ thể, các chương trình thuộc loại này hỗ trợ khả năng lưu trữ, sửa chữa, xóa và tìm kiếm thông tin trong một cơ sở dữ liệu (CSDL). Có rất nhiều loại hệ quản trị CSDL khác nhau: từ phần mềm nhỏ chạy trên máy tính cá nhân cho đến những hệ quản trị phức tạp chạy trên một hoặc nhiều siêu máy tính. Các ngôn ngữ lập trình truy vấn như SQL thường đi kèm theo với hệ quản trị cơ sở dữ liệu để mọi lập trình viên có thể dễ dàng tương tác với dữ liệu hơn khi họ cần.



Hình 2.8: **Database Management System**

Các DBMS hiện tại cung cấp các chức năng khác nhau cho phép quản lý cơ sở dữ liệu và dữ liệu có thể được phân loại thành bốn nhóm chức năng chính:

* **Định nghĩa dữ liệu**: Tạo, sửa đổi và loại bỏ các định nghĩa xác định tổ chức dữ liệu
* **Cập nhật**: Chèn, sửa đổi và xóa dữ liệu thực tế
* **Truy xuất**: Cung cấp thông tin dưới dạng có thể sử dụng trực tiếp hoặc để xử lý thêm bởi các ứng dụng khác. Dữ liệu được truy xuất có thể được cung cấp ở dạng cơ bản giống như được lưu trữ trong cơ sở dữ liệu hoặc ở dạng mới thu được bằng cách thay đổi hoặc kết hợp dữ liệu hiện có từ cơ sở dữ liệu
* **Quản trị**: Đăng ký và giám sát người dùng, thực thi bảo mật dữ liệu, giám sát hiệu suất, duy trì tính toàn vẹn dữ liệu, xử lý kiểm soát đồng thời và khôi phục thông tin đã bị hỏng do một số sự kiện như lỗi hệ thống không mong muốn

**Ưu điểm của DBMS:**

* Quản lý được dữ liệu dư thừa.
* Đảm báo tính nhất quán cho dữ liệu.
* Tạo khả năng chia sẻ dữ liệu nhiều hơn.
* Cải tiến tính toàn vẹn cho dữ liệu.

**Nhược điểm của DBMS:**

* Tốt thì khá phức tạp.
* Tốt thường rất lớn chiếm nhiều dung lượng bộ nhớ.
* Giá cả khác nhau tùy theo môi trường và chức năng.
* Được viết tổng quát cho nhiều người dùng thì thường chậm.

#### 2.1.7.2. Hệ quản trị cơ sở MySQL

**MySQL** là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u" \o "Hệ quản trị cơ sở dữ liệu) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F" \o "Phần mềm nguồn mở) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ.



Hình 2.9: **MySQL**

**MySQL** là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL). **MySQL** được sử dụng cho việc bổ trợ NodeJs, [PHP](https://vi.wikipedia.org/wiki/PHP" \o "PHP), [Perl](https://vi.wikipedia.org/wiki/Perl" \o "Perl), và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl, ...

Các **ưu điểm** của MySQL:

* **Tốc độ:** MySQL rất nhanh. Những nhà phát triển cho rằng MySQL là cơ sở dữ liệu nhanh nhất để có thể sử dụng.
* **Dễ sử dụng:** MySQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn.
* **Giá thành:** MySQL là miễn phí cho hầu hết các việc sử dụng trong một tổ chức.
* **Hỗ trợ ngôn ngữ truy vấn:** MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Người dùng cũng có thể truy cập MySQL bằng cách sử dụng các ứng dụng mà hỗ trợ ODBC (Open Database Connectivity -một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft).
* **Năng lực:** Nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Người dùng có thể truy cập MySQL tương tác với sử dụng một vài giao diện để có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web…
* **Kết nối và bảo mật:** MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó có thể chia sẽ dữ liệu của bản thân với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người mà không nên nhìn thấy dữ liệu thì không thể nhìn thấy được.
* **Tính linh động:** MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với mọi phần cứng từ các máy PC ở nhà cho đến các máy server.
* **Sự phân phối rộng:** MySQL rất dễ dàng đạt được, chỉ cần sử dụng trình duyệt web đã có. Nếu như không hiểu làm thế nào mà nó làm việc hay tò mò về thuật toán, người dùng có thể lấy mã nguồn và tìm tòi nó. Nếu người dùng không thích một vài cái, thì có thể thay đổi nó.
* **Sự hỗ trợ:** Người dùng có thể tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ trả lời các câu hỏi trên mailing list thường chỉ trong vài phút. Khi lỗi được phát hiện, các nhà phát triển sẽ đưa ra cách khắc phục trong vài ngày, thậm chí có khi trong vài giờ và cách khắc phục đó sẽ ngay lập tức có sẵn trên Internet.

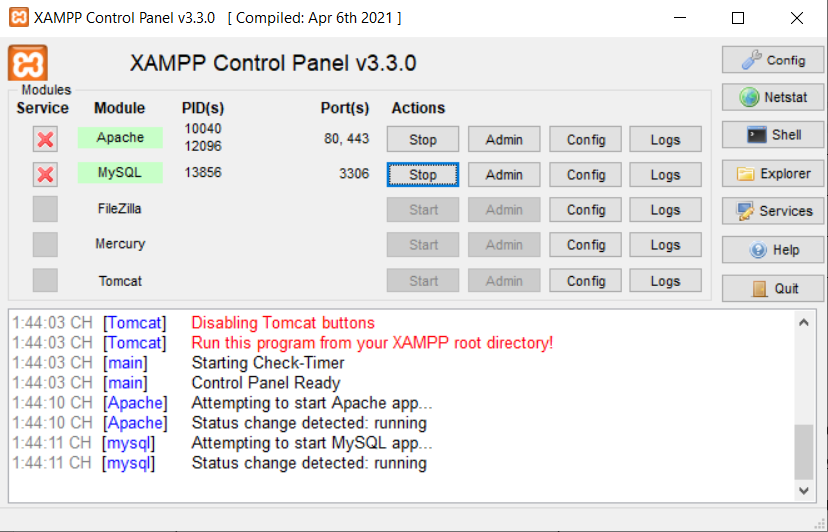
Bảng 2.1: Kiểu dữ liệu trong MySQL

|  |  |
| --- | --- |
| **Kiểu dữ liệu** | **Kích thước** |
| TINYINT | -128 - 127 |
| SMALLINT | -32768 - 32767 |
| MEDIUMINT | -8388608 - 8388607 |
| INT | -2147483648 - 2147483647 |
| BIGINT | - 2^64 − 1 - 26^3 − 1 |
| FLOAT | 4 bytes |
| DOUBLE | 8 bytes |
| DECIMAL, NUMERIC | 4 bytes |
| DATE | 3 bytes |
| DATETIME | 8 bytes |
| TIMESTAMP | 4 bytes |
| TIME | 3 bytes |
| YEAR | 1 byte |
| CHAR | M × w bytes |
| VARCHAR | L + 2 bytes |
| TINYBLOB, TINYTEXT | L + 1 bytes |
| BLOB, TEXT | L + 2 bytes |
| MEDIUMBLOB, MEDIUMTEXT | L + 3 bytes |
| LONGBLOB, LONGTEXT | L + 4 bytes |

## **Công cụ sử dụng**

### **XAMPP**

XAMPP là chương trình tạo máy chủ Web (Web Server) được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server và các công cụ như phpMyAdmin. Không như Appserv, Xampp có chương trình quản lý khá tiện lợi, cho phép chủ động bật tắt hoặc khởi động lại các dịch vụ máy chủ bất kỳ lúc nào.



Hình 2.10: **Giao diện Xampp**

Xampp là một mã nguồn mở máy chủ web đa nền được phát triển bởi Apache Friends, bao gồm chủ yếu là Apache HTTP Server, MariaDB database, và interpreters dành cho những đối tượng sử dụng ngôn ngữ PHP và Perl. Xampp là viết tắt của Cross-Platform (đa nền tảng-X), Apache (A), MariaDB (M), PHP (P) và Perl (P). Nó phân bố Apache nhẹ và đơn giản, khiến các lập trình viên có thể dễ dàng tạo ra máy chủ web local để kiểm tra và triển khai trang web của mình. Tất cả mọi thứ cần cho phát triển một trang web - Apache ( ứng dụng máy chủ), Cơ sở dữ liệu (MariaDB) và ngôn ngữ lập trình (PHP) được gói gọn trong 1 tệp. Xampp cũng là 1 đa nền tảng vì nó có thể chạy tốt trên cả Linux, Windows và Mac. Hầu hết việc triển khai máy chủ web thực tế đều sử dụng cùng thành phần như XAMPP nên rất dễ dàng để chuyển từ máy chủ local sang máy chủ online.

### **Sublime Text**

Sublime Text là trình soạn thảo và soạn thảo code miễn phí được rất nhiều **[lập trình viên](https://aptech.fpt.edu.vn/lap-trinh-vien.html" \t "_blank)** sử dụng hiện nay. Phần mềm lập trình Sublime Text cung cấp các plug-in miễn phí và hỗ trợ trên nhiều nền tảng khác nhau có thể kể đến như Windows, Mac và Linux để mang đến cho người dùng sự tiện lợi, thoải mái khi sử dụng. Nhờ các tính năng tiện lợi do Sublime Text cung cấp, các lập trình viên có thể sử dụng phần mềm này để dễ dàng chỉnh sửa các local file hoặc một codebase. Đặc biệt, công cụ này không tốn một khoản chi phí nào khi sử dụng nhưng cũng đầy đủ các tính năng hỗ trợ cần thiết.

Sublime Text được sử dụng [Integrated Development Editor](https://en.wikipedia.org/wiki/Integrated_development_environment" \t "_blank) (IDE) chương trình chỉnh sửa phát triển tích hợp giống như mã [Visual Studio](https://visualstudio.microsoft.com/" \t "_blank) và [NetBeans](https://netbeans.org/" \t "_blank). Phiên bản hiện tại của trình soạn thảo Sublime Text là 4.0 và tương thích với các hệ điều hành khác nhau như Windows, [Linux](https://dizibrand.com/linux/" \t "_blank" \o "Linux) và MacOS.

Các tính năng vượt trội của Sublime Text:

* Tính năng Goto Anything – tính năng điều hướng nhanh đến các biểu tượng, từ hoặc dòng, đến các file.
* Tính năng Goto Definition - Sublime Text tự động tạo ra một mục toàn Project của phương thức, hàm và lớp. Đây chính là việc sử dụng thông tin từ các định nghĩa cú pháp đó nhé!
* Tính năng Multiple Selections – đây là một tính năng cho phép bạn thay đổi nhiều dòng trong cùng một thời điểm, đổi tên biến dễ dàng và thực hiện các thao tác trong file nhanh chóng một cách bất ngờ.
* Tính năng Command Palette – nó chứa các chức ít được sử dụng thường xuyên như thay đổi cú pháp, sắp xếp, cài đặt thụt lề.
* Hệ sinh thái gói và API mạnh mẽ - trình soạn thảo mã nguồn Sublime Text có API Python mạnh mẽ, điều này cho phép  các Plugin tăng cường các chức năng tích hợp.
* Tính năng tùy chỉnh mọi thứ - Sublime Text cung cấp các phím tắt, Macro, Snippet, Menu và hơn thế nữa tất cả chúng đều tùy chỉnh với các tệp Json đơn giản trong Sublime Text. Điều này mang đến sự linh hoạt cho nhiều người dùng khác nhau và tùy chỉnh với từng dự án và từng loại tệp để đạt được hiệu quả tốt nhất.
* Split Editing – tính năng này cho phép bạn tận dụng tối đa màn hình rộng cửa mình cùng với Split Editing để chỉnh sửa các phải liên kề nhau hoặc chỉnh sửa 2 vị trí của file, đặc biệt bạn còn có thể chỉnh sửa nhiều cột, nhiều hàng như ý muốn của bản thân.
* Tính năng chuyển đối Project tức thì – trong Sublime Text, các Project nắm toàn bộ nội dung của không gian làm việc, cả các bản chưa được lưu và các bản sửa đổi, điều này giúp bạn dễ dàng chuyển đổi các Project khi cần thiết và các thay đổi sẽ được khôi phục trong lần mở tiếp theo của Project.
* Hiệu suất – vì nó được xây dựng từ các thành phần tùy chỉnh, điều này tạo khả năng thích ứng tốt nhất và vượt trội, từ đó mang đến hiệu suất làm việc tốt nhất.

# **CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## **3.1. Khảo sát và đặc tả nghiệp vụ**

### **3.1.1. Giới thiệu đơn vị khảo sát**

* Trường Đại Học Công Nghệ Giao Thông Vận Tải.
* Địa chỉ: 54 Triều Khúc, Phường Thanh Xuân Nam, Thanh Xuân, TP Hà Nội.
* Quy mô: Thư viện sử dụng trong trường..
* Cách thức hoạt động: hotline, địa điểm cố định.

### **3.1.2. Nội dung khảo sát**

* Đối tượng khảo sát: Quản lý
* Phương thức khảo sát: Trao đổi online kết hợp tự tìm hiểu.
* Mục đích: Thu thập thông tin, nắm bắt được cách thức vận hành của thư viện và nghiệp vụ chuyên môn.

Bảng 3.1: Bảng khảo sát

|  |  |  |
| --- | --- | --- |
| **STT** | **Câu hỏi** | **Câu trả lời** |
| 1 | Chào anh, anh cho em hỏi yêu cầu thư viện của trường đang mong muốn xây dựng như thế nào ạ? | Bên trường đang cần 1 website để nhà trường có thể quản lý các đầu sách, giúp cho cán bộ và sinh viên trong trường dễ dàng tìm kiếm các đầu sách, giao diện thân thiện dễ sử dụng….. |
| 2 | Cho em hỏi là, cửa hàng còn có những ai hỗ trợ chị trong việc quản lý nữa ạ? Và họ có những nhiệm vụ cụ thể như thế nào ạ? | Thư viện sẽ có nhân viên hỗ trợ nắm bắt mỗi nghiệp vụ mà họ được giao để thư viện có thể hoạt động tốt. |
| 3 | Anh có cần một hệ thống website để giúp quản lý hiệu quả hơn không ạ? | Anh có cần một hệ thống website để giúp quản lý hiệu quả hơn không ạ? |

### **3.1.3. Hiện trạng khảo sát**

*Đối tượng tham gia:*

* Người quản lý: Quản lý hoạt động trên Website, quản lý sản phẩm, bài viết, tin tức, liên hệ, sách mượn, ....
* Người sử dụng: khách hàng

*Về hệ thống cũ:*

Ưu điểm:

* Đã cung cấp đầy đủ cho người dùng về thông tin cần thiết về sản phẩm

Nhược điểm:

* Khó tiếp cận đến khách hàng
* Giao diện chưa thân thiện với người dùng

## **3.2. Yêu cầu xây dựng hệ thống**

**Trang chủ**

* Trình bày thân thiện.
* Hiển thị các chức năng nổi bật của website.
* Hiển thị thông tin của thư viện (Số điện thoại, Địa chỉ liên hệ).
* Các chức năng của Website (Tìm kiếm, Xem thông tin loại sản phẩm, Các sản phẩm và bài viết nổi bật).
  + Đơn giản, định hướng, giúp người dùng dễ sử dụng.

**Trang danh sách sản phẩm**

* + Trình bày thân thiện.
  + Các sản phẩm mới nhất và bài viết mới nhất.
  + Sắp xếp, phân loại sản phẩm theo các loại sản phẩm khác nhau.

**Trang chi tiết sản phẩm**

* Trình bày thân thiện, dễ sử dụng.
* Thông tin đầy đủ, tất cả các thông tin chi tiết về sản phẩm.
* Cho phép người dùng thêm các sản phẩm muốn mua vào giỏ hàng

**Trang quản trị**

* Quản lý tài khoản.
* Quản lý danh mục sản phẩm.
* Quản lý sản phẩm.
* Quản lý bài viết.
* ….

## **3.3. Mô tả bài toán**

Website bán quản lý thư viện online với mục tiêu hỗ trợ khách trong việc lựa chọn sản phẩm một cách nhanh chóng và thuận tiện nhất không phải đến trực tiếp thư viện mà vẫn biết đầy đủ các thông tin của các sản phẩm trong thư viện đang có tiết kiệm được thời gian đối với khách hàng cũng như thư viện. Hệ thống website mô tả như sau:

Thư viện số là một hệ thống lưu trữ, tổ chức và phân phối thông tin dưới dạng số, cho phép người dùng truy cập và sử dụng tài nguyên một cách dễ dàng và tiện lợi. Dưới đây là mô tả chi tiết về cách hoạt động của thư viện số:

### 1. \*\*Thu thập Tài nguyên Số\*\*

- \*\*Số hóa Tài liệu\*\*: Các tài liệu giấy, sách, tạp chí hay các tài sản trí tuệ khác sẽ được chuyển đổi sang định dạng số thông qua máy quét, máy ảnh hoặc các phần mềm số hóa.

- \*\*Tích hợp Tài nguyên Số\*\*: Thư viện số cũng có thể thu thập tài nguyên từ các nguồn đã có sẵn trực tuyến, bao gồm cơ sở dữ liệu, trang web và các kho lưu trữ số khác.

### 2. \*\*Tổ chức và Quản lý Tài nguyên\*\*

- \*\*Phân loại\*\*: Các tài nguyên số được phân loại theo chủ đề, thể loại, ngày phát hành, tác giả, v.v. Điều này giúp người dùng dễ dàng tìm kiếm và duyệt qua các tài liệu.

- \*\*Thư mục và Metadata\*\*: Mỗi tài liệu số sẽ kèm theo metadata (dữ liệu mô tả) để cung cấp thông tin về tài liệu đó như tên tác giả, tiêu đề, năm xuất bản, từ khóa, và mô tả nội dung.

### 3. \*\*Lưu trữ\*\*

- \*\*Hệ thống Lưu trữ\*\*: Tài liệu số được lưu trữ trên các máy chủ hoặc trong các kho lưu trữ đám mây, đảm bảo an toàn, bền vững và dễ dàng truy cập.

- \*\*Sao lưu\*\*: Để bảo vệ tài nguyên khỏi mất mát, các thư viện số thường thực hiện việc sao lưu định kỳ dữ liệu.

### 4. \*\*Truy cập và Tìm kiếm\*\*

- \*\*Giao diện Người dùng\*\*: Thư viện số thường cung cấp một giao diện dễ sử dụng, cho phép người dùng tìm kiếm tài liệu thông qua các thanh tìm kiếm, bộ lọc và danh mục.

- \*\*Công cụ Tìm kiếm\*\*: Các công cụ tìm kiếm mạnh mẽ giúp người dùng có thể tìm thấy tài liệu dựa trên từ khóa, tên tác giả, ngày phát hành, v.v.

### 5. \*\*Dịch vụ và Tương tác\*\*

- \*\*Dịch vụ Trực tuyến\*\*: Nhiều thư viện số cung cấp dịch vụ trực tuyến như đặt mượn tài liệu, tải về, và truy cập vào các khóa học trực tuyến hoặc sản phẩm số khác.

- \*\*Hỗ trợ Người dùng\*\*: Thư viện số thường có đội ngũ hỗ trợ người dùng để giải đáp thắc mắc và hướng dẫn cách sử dụng.

### 6. \*\*Bảo mật và Quyền truy cập\*\*

- \*\*Quyền truy cập\*\*: Thư viện số sẽ quản lý quyền truy cập cho các tài nguyên số, đảm bảo rằng chỉ những người có quyền lợi hợp pháp mới được truy cập tài liệu.

- \*\*Bảo mật Dữ liệu\*\*: Hệ thống bảo mật được áp dụng để bảo vệ tài liệu và thông tin cá nhân của người dùng.

### 7. \*\*Cập nhật và Bảo trì\*\*

- \*\*Cập nhật Tài nguyên\*\*: Thư viện số thường xuyên cập nhật tài nguyên, bổ sung tài liệu mới và loại bỏ những tài liệu không còn giá trị.

- \*\*Bảo trì Hệ thống\*\*: Đảm bảo hệ thống hoạt động trơn tru, thường xuyên bảo trì máy chủ và phần mềm để cải thiện hiệu suất và bảo mật.

Tóm lại, thư viện số không chỉ là một kho lưu trữ tài nguyên mà còn là một hệ thống toàn diện giúp người dùng dễ dàng truy cập, tìm kiếm và sử dụng thông tin một cách hiệu quả và tiện lợi.

Quản trị viên có quyền quản lý tất cả các chức năng trên trang web quản trị..

## **3.4. Phân tích thiết kế**

### **3.4.1. Xác định các tác nhân (Actor) và chức năng (UseCase)**

Bảng 3.2: Xác định Actor và UseCase

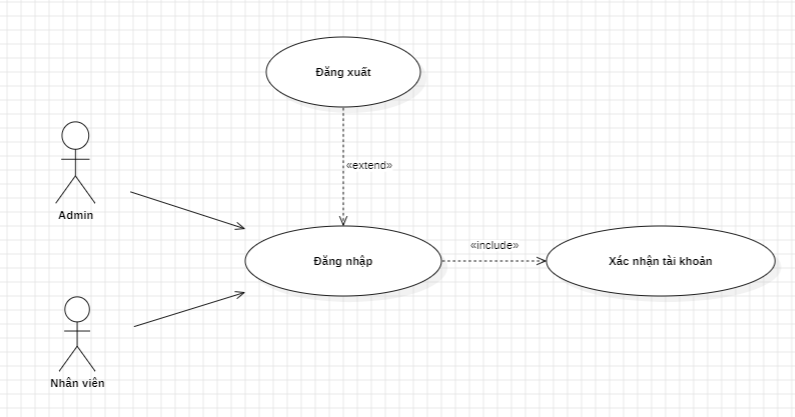
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tác nhân** | **Mô tả tác nhân** | **Chức năng** |
| 1 | Khách hàng | Khách hàng người muốn xem hoặc mua hàng trên trang web | * Đăng nhập/Đăng ký tài khoản * Xem thông tin sản phẩm * Xem các bài viết * Tìm kiếm sản phẩm   … |
| 2 | Nhân viên | Nhân viên là người hỗ trợ quản lý hoạt động cho các chức năng của trang web | * Đăng nhập * Quản lý các bài viết * Quản lý sản phẩm * Quản lý danh mục sản phẩm * Quản lý thương hiệu * … |
| 3 |

### **3.4.2. Biểu đồ UseCase**

#### 3.4.2.1. Biểu đồ UseCase tổng quát

Hình 3.1: Biểu đồ UseCase tổng quát

#### 3.4.2.2. Biểu đồ chi tiết UseCase đăng nhập/đăng xuất của Admin và Nhân viên

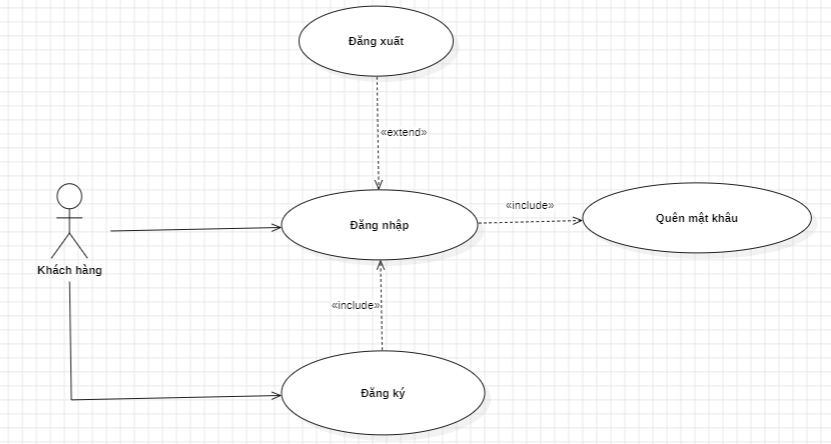


Hình 3.2: Biểu đồ chi tiết UseCase “Đăng nhập/Đăng xuất”

Bảng 3.3: Đặc tả UseCase “Đăng nhập/Đăng xuất”

|  |  |
| --- | --- |
| **Tên Usecase** | **Đăng nhập / Đăng xuất** |
| Tác nhân | * Nhân viên, Quản trị viên |
| Tiền điều kiện (đăng nhập) | * Người dùng đã có tài khoản trong hệ thống và đăng nhập thành công |
| Mục đích  (đăng nhập) | * Xác nhận và xác thực danh tính người dùng khi họ truy cập vào hệ thống * Bảo vệ thông tin cá nhân người dùng và đảm bảo chỉ có người được ủy quyền mới có thể đăng nhập được vào. * Người dùng đăng nhập vào hệ thống thông qua “Email” và “Mật khẩu” |
| Dòng sự kiện chính (đăng nhập) | * Người dùng muốn đăng nhập vào hệ thống * Hệ thống hiển thị giao diện đăng nhập * Người dùng nhập “Email” và “Mật khẩu” * Hệ thống kiểm tra và xác nhận thông tin đăng nhập * Hiển thị giao diện chính của hệ thống * Kết thúc Usecase. |
| Dòng sự kiện phụ (đăng nhập) | * Nếu người dùng nhập sai hoặc không nhập email/mật khẩu thì không thực hiện đăng nhập. * Nếu người dùng nhập sai mật khẩu sẽ thông báo cho người dùng biết. |
| Hậu điều kiện (đăng nhập) | * **Trường hợp thành công**: Hệ thống sẽ hiển thị giao diện quản lý chính. Người dùng có thể thực hiện các chức năng theo đúng quyền hạn của mình. * **Trường hợp thất bại**: Hệ thống sẽ đưa ra thông báo” Tài khoản nhập sai hoặc không tồn tại” và yêu cầu đăng nhập lại. |
| Tiền điều kiện (đăng xuất) | * Người dùng đã đăng nhâp trước khi Usecase xảy ra |
| Mục đích (đăng xuất) | Đăng xuất ra khỏi hệ thống |
| Dòng sự kiện chính (đăng xuất) | * Người dùng muốn đăng xuất ra khỏi hệ thống * Người dùng nhấn vào biểu tượng đăng xuất * Hiển thị giao diện trang chủ chưa đăng nhập * Kết thúc Usecase |
| Hậu điều kiện (đăng xuất) | * **Trường hợp thành công**: Hệ thống sẽ hiển thị giao diện màn hình trang chủ hệ thống ở trạng thái chưa đăng nhập. * **Trường hợp thất bại**: Hệ thống sẽ đưa ra thông báo, yêu cầu thức hiện lại. |

#### 3.4.2.3. Biểu đồ chi tiết UseCase “Đăng nhập – Đăng ký” của khách hàng



Hình 3.3: Biểu đồ chi tiết UseCase “Đăng nhập – Đăng ký”

Bảng 3.3: Đặc tả UseCase “Đăng nhập – Đăng ký”

|  |  |
| --- | --- |
| **Tên Usecase** | **Đăng nhập / Đăng ký** |
| Tác nhân | * Khách hàng |
| Tiền điều kiện (đăng ký) | * Khách hàng muốn đăng ký tài khoản để đăng nhập và vào trang web mua sắm sản phẩm |
| Mục đích (đăng ký) | * Xác nhận và xác thực danh tính khách hàng khi họ truy cập vào hệ thống * Tạo tài khoản cho khách hàng truy cập vào trang web để mua sắm sản phẩm * Cung cấp quyền lợi cho khách hàng khi mua sản phẩm * Người dùng đăng nhập vào hệ thống thông qua “Email” và “Mật khẩu” |
| Dòng sự kiện chính (đăng ký) | * Khách hàng muốn đăng ký để đăng nhập vào hệ thống * Hệ thống hiển thị giao diện đăng ký * Khách hàng nhập thông tin cá nhân * Hệ thống kiểm tra và xác nhận thông tin đăng ký * Hiển thị giao diện đăng nhập của hệ thống * Kết thúc Usecase. |
| Dòng sự kiện phụ (đăng ký) | * Nếu khách hàng nhập sai thông tin hệ thống sẽ không tạo được tài khoản cho khách hàng. |
| Tiền điều kiện (đăng nhập) | * Khách hàng đã có tài khoản trong hệ thống và đăng nhập thành công |
| Mục đích (đăng nhập) | * Xác nhận và xác thực danh tính khách hàng khi họ truy cập vào hệ thống * Bảo vệ thông tin cá nhân khách hàng và đảm bảo chỉ có người được ủy quyền mới có thể đăng nhập được vào. * Khách hàng đăng nhập vào hệ thống thông qua “Email” và “Mật khẩu” |
| Dòng sự kiện chính (đăng nhập) | * Khách hàng muốn đăng nhập vào hệ thống * Hệ thống hiển thị giao diện đăng nhập * Khách hàng nhập “Email” và “Mật khẩu” * Hệ thống kiểm tra và xác nhận thông tin đăng nhập * Hiển thị giao diện chính của hệ thống * Kết thúc Usecase. |
| Dòng sự kiện phụ (đăng nhập) | * Nếu khách hàng nhập sai hoặc không nhập email/mật khẩu thì không thực hiện đăng nhập. * Nếu khách hàng nhập sai mật khẩu sẽ thông báo cho người dùng biết. |
| Hậu điều kiện (đăng nhập) | * **Trường hợp thành công**: Hệ thống sẽ hiển thị giao diện quản lý chính. Khách hàng có thể thực hiện các chức năng theo đúng quyền hạn của mình. * **Trường hợp thất bại**: Hệ thống sẽ đưa ra thông báo” Tài khoản nhập sai hoặc không tồn tại” và yêu cầu đăng nhập lại. |
| Tiền điều kiện (đăng xuất) | * Khách hàng đã đăng nhâp trước khi Usecase xảy ra |
| Mục đích (đăng xuất) | * Đăng xuất ra khỏi hệ thống |
| Dòng sự kiện chính (đăng xuất) | * Khách hàng muốn đăng xuất ra khỏi hệ thống * Khách hàng nhấn vào biểu tượng đăng xuất * Hiển thị giao diện trang chủ chưa đăng nhập * Kết thúc Usecase |
| Hậu điều kiện (đăng xuất) | * **Trường hợp thành công**: Hệ thống sẽ hiển thị giao diện màn hình trang chủ hệ thống ở trạng thái chưa đăng nhập. * **Trường hợp thất bại**: Hệ thống sẽ đưa ra thông báo, yêu cầu thức hiện lại. |

#### 3.4.2.4. Biểu đồ chi tiết UseCase “Xem sản phẩm” của Khách hàng

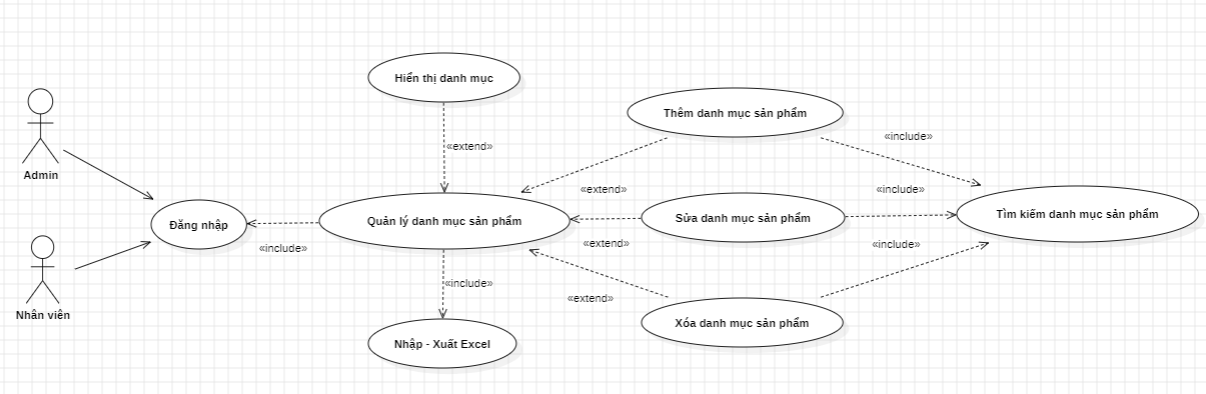


Hình 3.4: Biểu đồ chi tiết UseCase “Xem sản phẩm”

Bảng 3.4: Đặc tả UseCase “Xem sản phẩm”

|  |  |
| --- | --- |
| **Tên Usecase** | **Xem sản phẩm** |
| Tác nhân | * Khách hàng |
| Tiền điều kiện | * Khách hàng đã có tài khoản trong hệ thống và đăng nhập thành công hoặc không cần đăng nhập |
| Mục đích | * Tìm hiểu rõ hơn về sản phẩm mà khách hàng đang có nhu cầu tìm kiếm mua sắm * Khách hàng sẽ xem và so sánh các sản phẩm đã xem cùng phân khúc rồi đưa ra quyết định mua. * Lưu sản phẩm vào danh sách yêu thích |
| Dòng sự kiện chính | * Khách hàng có thể đăng nhập hoặc không đăng nhập vào hệ thống vẫn có thể xem được chi tiết sản phẩm * Sau khi vào trang chủ, hệ thống sẽ hiển thị ra danh sách sản phẩm mới, sản phẩm theo danh mục và thương hiệu * Khách hàng có thể tìm kiếm và xem chi tiết các sản phẩm cần xem, hệ thống sẽ chuyển trang hiển thị chi tiết sản phẩm * Kết thúc Usecase. |
| Hậu điều kiện | * **Trường hợp thành công**: Sau khi tìm kiếm có kết quả thống hiển thị ra sản phẩm mà khách hàng cần xem để xem và mua * **Trường hợp thất bại**: Nếu không có sản phẩm mà khách hàng cần xem sau khi tìm kiếm, hệ thống sẽ không hiển thị thông tin tìm kiếm sản phẩm |

#### 3.4.2.8. Biểu đồ chi tiết UseCase “Quản lý danh mục sản phẩm” của Admin và Nhân viên

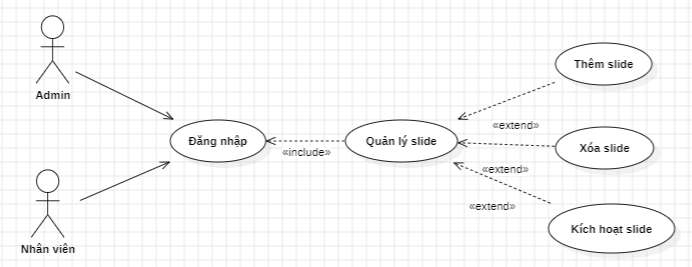


Hình 3.8: Biểu đồ chi tiết UseCase “Quản lý danh mục sản phẩm”

Bảng 3.8: Đặc tả UseCase “Quản lý danh mục sản phẩm”

|  |  |
| --- | --- |
| **Tên Usecase** | **Quản lý danh mục sản phẩm** |
| Tác nhân | * Quản trị viên và Nhân viên |
| Tiền điều kiện | * Quản trị viên và Nhân viên đăng nhập thành công vào hệ thống |
| Mục đích | * Quản trị viên và Nhân viên sử dụng chức năng để thêm mới, sửa, xóa danh mục sản phẩm và tìm kiếm danh mục sản phẩm |
| Dòng sự kiện chính | * Quản trị viên và Nhân viên chọn quản lý danh mục sản phẩm   **Thêm mới danh mục sản phẩm:**   * Chọn chức năng thêm danh mục sản phẩm * Nhập thông tin danh mục sản phẩm cần thêm vào form vừa hiển thị * Sau khi nhập thông tin xong, hệ thống sẽ kiểm tra dữ liệu hợp lệ, lưu vào cơ sở dữ liệu và gửi thông báo * Kết thúc Usecase   **Sửa danh mục sản phẩm:**   * Chọn chức năng liệt kê danh mục sản phẩm và tìm kiếm danh mục sản phẩm cần sửa * Danh mục sản phẩm được hiển thị * Nhấn vào biểu tượng sửa trên danh mục sản phẩm, hệ thống sẽ hiện ra form sửa * Sau khi nhập thông tin cần sửa, nhấn nút cập nhật, hệ thống sẽ kiểm tra dữ liệu hợp lệ, lưu vào cơ sở dữ liệu và gửi thông báo * Kết thúc Usecase   **Xóa danh mục sản phẩm:**   * Chọn chức năng liệt kê danh mục sản phẩm và tìm kiếm danh mục sản phẩm cần xóa * Danh mục sản phẩm được hiển thị * Nhấn vào biểu tượng xóa trên sản phẩm * Hiển thị cửa sổ xác nhận xóa, nếu hủy bỏ hệ thống sẽ không xóa, nếu xác nhận xóa danh mục sản phẩm sẽ bị xóa khỏi hệ thống * Hệ thống kiểm tra, thực hiện xóa và gửi thông báo * Kết thúc Usecase   **Tìm kiếm danh mục sản phẩm:**   * Quản trị viên và Nhân viên chọn tìm kiếm danh mục sản phẩm theo tên * Hệ thống hiển thị ngay lập tức danh mục sản phẩm cần tìm kiếm * Kết thúc Usecase |
| Dòng sự kiện phụ | **Thêm mới, sửa danh mục sản phẩm:**   * Hệ thống thông báo thông tin nhập chưa đầy đủ. * Thêm mới / Cập nhật không thành công. |
| Hậu điều kiện | * **Trường hợp thành công:** Thêm mới, sửa, xóa thông tin danh mục sản phẩm thành công, được lưu vào hệ thống * **Trường hợp thất bại:** Hệ thống thông báo lỗi, thông tin không được lưu |

#### 3.4.2.9. Biểu đồ chi tiết UseCase “Quản lý Slider” của Admin và Nhân viên

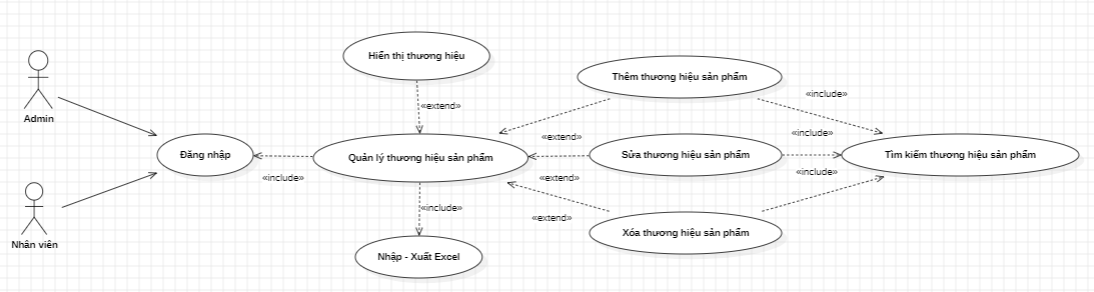


Hình 3.9: Biểu đồ chi tiết UseCase “Quản lý Slider”

Bảng 3.9: Đặc tả UseCase “Quản lý Slider”

|  |  |
| --- | --- |
| **Tên Usecase** | **Quản lý Slider** |
| Tác nhân | * Quản trị viên và Nhân viên |
| Tiền điều kiện | * Quản trị viên và Nhân viên đăng nhập thành công vào hệ thống |
| Mục đích | * Quản trị viên và Nhân viên sử dụng chức năng để thêm mới và xóa slider |
| Dòng sự kiện chính | * Quản trị viên và Nhân viên chọn quản lý sldier   **Thêm mới slider:**   * Chọn chức năng thêm slider * Thêm ảnh slider và nhập tên slider * Nhấn nút thêm slider * Kết thúc Usecase   **Xóa sldier:**   * Chọn chức năng liệt kê slider * Danh mục slider được hiển thị * Nhấn vào biểu tượng xóa trên slider * Hiển thị cửa sổ xác nhận xóa, nếu hủy bỏ hệ thống sẽ không xóa, nếu xác nhận xóa slider sẽ bị xóa khỏi hệ thống * Hệ thống kiểm tra, thực hiện xóa và gửi thông báo * Kết thúc Usecase |
| Hậu điều kiện | * Thêm mới slider thông tin khuyến mãi hiển thị ở đầu trang sẽ tiếp cận đến với khách hàng nhanh hơn và hiệu quả hơn trong việc thúc đẩy doanh số |

#### 3.4.2.10. Biểu đồ chi tiết UseCase “Quản lý thương hiệu sản phẩm” của Admin và Nhân viên

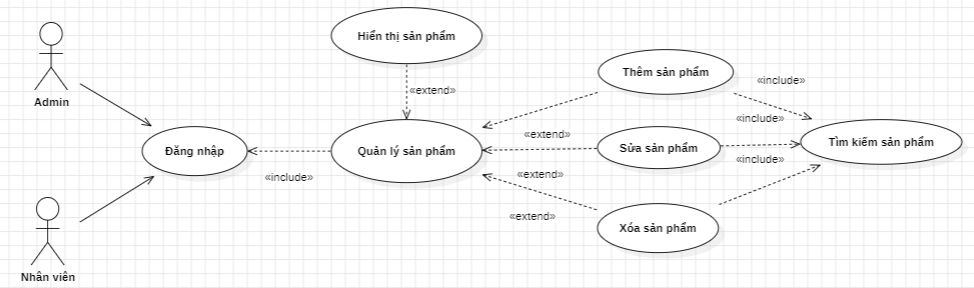


Hình 3.10: Biểu đồ chi tiết UseCase “Quản lý thương hiệu sản phẩm”

Bảng 3.10: Đặc tả UseCase “Quản lý thương hiệu sản phẩm”

|  |  |
| --- | --- |
| **Tên Usecase** | **Quản lý thương hiệu sản phẩm** |
| Tác nhân | * Quản trị viên và Nhân viên |
| Tiền điều kiện | * Quản trị viên và Nhân viên đăng nhập thành công vào hệ thống |
| Mục đích | * Quản trị viên và Nhân viên sử dụng chức năng để thêm mới, sửa, xóa thương hiệu sản phẩm và tìm kiếm thương hiệu sản phẩm |
| Dòng sự kiện chính | * Quản trị viên và Nhân viên chọn quản lý thương hiệu sản phẩm   **Thêm mới thương hiệu sản phẩm:**   * Chọn chức năng thêm thương hiệu sản phẩm * Nhập thông tin thương hiệu sản phẩm cần thêm vào form vừa hiển thị * Sau khi nhập thông tin xong, hệ thống sẽ kiểm tra dữ liệu hợp lệ, lưu vào cơ sở dữ liệu và gửi thông báo * Kết thúc Usecase   **Sửa thương hiệu sản phẩm:**   * Chọn chức năng liệt kê thương hiệu sản phẩm và tìm kiếm thương hiệu sản phẩm cần sửa * Thương hiệu sản phẩm được hiển thị * Nhấn vào biểu tượng sửa trên thương hiệu sản phẩm, hệ thống sẽ hiện ra form sửa * Sau khi nhập thông tin cần sửa, nhấn nút cập nhật, hệ thống sẽ kiểm tra dữ liệu hợp lệ, lưu vào cơ sở dữ liệu và gửi thông báo * Kết thúc Usecase   **Xóa thương hiệu sản phẩm:**   * Chọn chức năng liệt kê thương hiệu sản phẩm và tìm kiếm thương hiệu sản phẩm cần xóa * Thương hiệu sản phẩm được hiển thị * Nhấn vào biểu tượng xóa trên thương hiệu sản phẩm * Hiển thị cửa sổ xác nhận xóa, nếu hủy bỏ hệ thống sẽ không xóa, nếu xác nhận xóa thương hiệu sản phẩm sẽ bị xóa khỏi hệ thống * Hệ thống kiểm tra, thực hiện xóa và gửi thông báo * Kết thúc Usecase   **Tìm kiếm thương hiệu sản phẩm:**   * Quản trị viên và Nhân viên chọn tìm kiếm thương hiệu sản phẩm theo tên * Hệ thống hiển thị ngay lập tức thương hiệu sản phẩm cần tìm kiếm * Kết thúc Usecase |
| Dòng sự kiện phụ | **Thêm mới, sửa thương hiệu sản phẩm:**   * Hệ thống thông báo thông tin nhập chưa đầy đủ. * Thêm mới / Cập nhật không thành công. |
| Hậu điều kiện | * **Trường hợp thành công:** Thêm mới, sửa, xóa thông tin thương hiệu sản phẩm thành công, được lưu vào hệ thống * **Trường hợp thất bại:** Hệ thống thông báo lỗi, thông tin không được lưu |

#### 3.4.2.11. Biểu đồ chi tiết UseCase “Quản lý sản phẩm” của Admin và Nhân viên

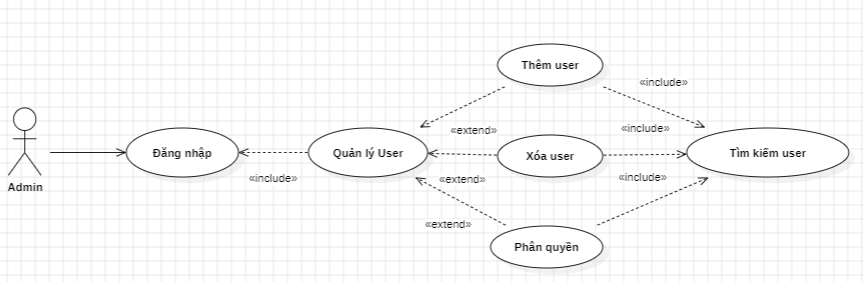


Hình 3.11: Biểu đồ chi tiết UseCase “Quản lý sản phẩm”

Bảng 3.11: Đặc tả UseCase “Quản lý thương hiệu sản phẩm”

|  |  |
| --- | --- |
| **Tên Usecase** | **Quản lý sản phẩm** |
| Tác nhân | * Quản trị viên và Nhân viên |
| Tiền điều kiện | * Quản trị viên và Nhân viên đăng nhập thành công vào hệ thống |
| Mục đích | * Quản trị viên và Nhân viên sử dụng chức năng để thêm sản phẩm mới, sửa sản phẩm, xóa sản phẩm và tìm kiếm sản phẩm |
| Dòng sự kiện chính | * Quản trị viên và Nhân viên chọn quản lý sản phẩm   **Thêm mới sản phẩm:**   * Chọn chức năng thêm sản phẩm * Nhập thông tin sản phẩm cần thêm vào form vừa hiển thị * Sau khi nhập thông tin xong, hệ thống sẽ kiểm tra dữ liệu hợp lệ, lưu vào cơ sở dữ liệu và gửi thông báo * Kết thúc Usecase   **Sửa sản phẩm:**   * Chọn chức năng liệt kê sản phẩm và tìm kiếm sản phẩm cần sửa * Sản phẩm được hiển thị * Nhấn vào biểu tượng sửa trên sản phẩm, hệ thống sẽ hiện ra form sửa * Sau khi nhập thông tin cần sửa, nhấn nút cập nhật, hệ thống sẽ kiểm tra dữ liệu hợp lệ, lưu vào cơ sở dữ liệu và gửi thông báo * Kết thúc Usecase   **Xóa sản phẩm:**   * Chọn chức năng liệt kê sản phẩm và tìm kiếm sản phẩm cần xóa * Sản phẩm được hiển thị * Nhấn vào biểu tượng xóa trên sản phẩm * Hiển thị cửa sổ xác nhận xóa, nếu hủy bỏ hệ thống sẽ không xóa, nếu xác nhận xóa sản phẩm sẽ bị xóa khỏi hệ thống * Hệ thống kiểm tra, thực hiện xóa và gửi thông báo * Kết thúc Usecase   **Tìm kiếm sản phẩm:**   * Quản trị viên và Nhân viên chọn tìm kiếm sản phẩm theo tên * Hệ thống hiển thị ngay lập tức sản phẩm cần tìm kiếm * Kết thúc Usecase |
| Dòng sự kiện phụ | **Sửa sản phẩm:**   * Hệ thống thông báo thông tin nhập chưa đầy đủ. * Cập nhật không thành công. |
| Hậu điều kiện | * **Trường hợp thành công:** Thêm mới, sửa, xóa thông tin sản phẩm thành công, được lưu vào hệ thống * **Trường hợp thất bại:** Hệ thống thông báo lỗi, thông tin không được lưu |

#### 3.4.2.13. Biểu đồ chi tiết UseCase “Quản lý User” của Admin

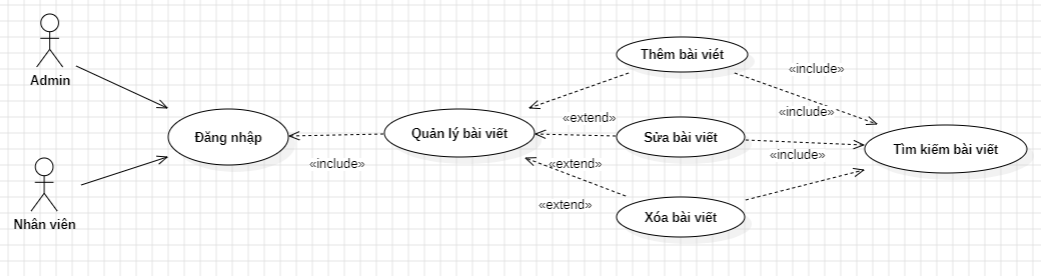


Hình 3.13: Biểu đồ chi tiết UseCase “Quản lý User”

Bảng 3.13: Đặc tả UseCase “Quản lý User”

|  |  |
| --- | --- |
| **Tên Usecase** | **Quản lý User** |
| Tác nhân | * Quản trị viên |
| Tiền điều kiện | * Quản trị viên đăng nhập thành công vào hệ thống |
| Mục đích | * Quản lý user nhằm mục đích phân quyền cho các user nhân viên trong hệ thống |
| Dòng sự kiện chính | * Quản trị viên và Nhân viên chọn quản lý tài khoản   **Thêm tài khoản:**   * Quản trị viên chọn chức năng thêm tài khoản để cấp tài khoản cho user nhân viên mới * Nhập thông tin user bao gồm họ tên, email, số điện thoại và password * Sau khi nhập thông tin xong ấn nút thêm user để tạo mới user, hệ thống sẽ kiểm tra dữ liệu hợp lệ, lưu vào cơ sở dữ liệu và gửi thông báo * Kết thúc Usecase   **Phân quyền user:**   * Quản trị viên chọn chức năng liệt kê tài khoản, tìm kiếm tài khoản cần phân quyền * Hiển thị tài khoản cần tìm kiếm và chọn roles và chọn phân quyền cho tài khoản * Hệ thống kiểm tra dữ liệu hợp lệ, lưu thông tin vào cơ sở dữ liệu và thông báo * Kết thúc Usecase   **Xóa tài khoản:**   * Quản trị viên chọn chức năng liệt kê tài khoản, tìm kiếm tài khoản cần xóa * Hiển thị tài khoản cần xóa, nhấn vào biểu tượng xóa trên tài khoản * Hiển thị cửa sổ xác nhận xóa, nếu hủy bỏ hệ thống sẽ không xóa, nếu xác nhận xóa đơn hàng sẽ bị xóa khỏi hệ thống * Hệ thống kiểm tra, thực hiện xóa và gửi thông báo * Kết thúc Usecase |
| Hậu điều kiện | * **Trường hợp thành công:** Thêm và xóa tài khoản thành công, được lưu vào hệ thống * **Trường hợp thất bại:** Hệ thống thông báo lỗi, thông tin không được lưu |

#### 3.4.2.15. Biểu đồ chi tiết UseCase “Quản lý bài viết” của Admin và Nhân viên



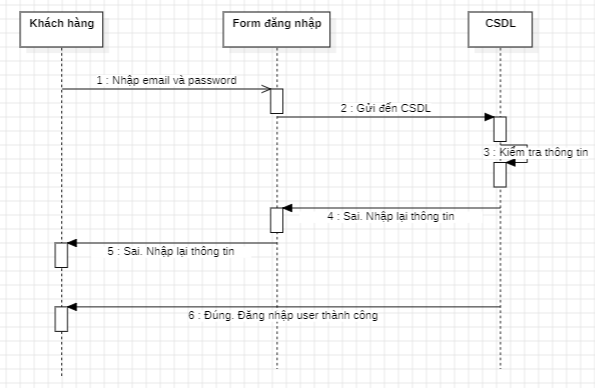
Hình 3.15: Biểu đồ chi tiết UseCase “Quản lý bài viết”

Bảng 3.15: Đặc tả UseCase “Quản lý bài viết”

|  |  |
| --- | --- |
| **Tên Usecase** | **Quản lý bài viết** |
| Tác nhân | * Quản trị viên và Nhân viên |
| Tiền điều kiện | * Quản trị viên và Nhân viên đăng nhập thành công vào hệ thống |
| Mục đích | * Quản trị viên và Nhân viên sử dụng chức năng để thêm mới, sửa, xóa bài viết |
| Dòng sự kiện chính | * Quản trị viên và Nhân viên chọn quản lý bài viết   **Thêm bài viết:**   * Chọn chức năng thêm bài viết * Nhập thông tin bài viết cần thêm vào form vừa hiển thị * Sau khi nhập thông tin xong, hệ thống sẽ kiểm tra dữ liệu hợp lệ, lưu vào cơ sở dữ liệu và gửi thông báo * Kết thúc Usecase   **Sửa bài viết:**   * Chọn chức năng liệt kê bài viết và tìm kiếm bài viết cần sửa * Bài viết được hiển thị * Nhấn vào biểu tượng sửa trên bài viết, hệ thống sẽ hiện ra form sửa * Sau khi nhập thông tin cần sửa, nhấn nút cập nhật, hệ thống sẽ kiểm tra dữ liệu hợp lệ, lưu vào cơ sở dữ liệu và gửi thông báo * Kết thúc Usecase   **Xóa bài viết:**   * Chọn chức năng liệt kê bài viết và tìm kiếm bài viết cần xóa * Bài viết cần tìm được hiển thị * Nhấn vào biểu tượng xóa trên bài viết * Hiển thị cửa sổ xác nhận xóa, nếu hủy bỏ hệ thống sẽ không xóa, nếu xác nhận xóa mã giảm giá sẽ bị xóa khỏi hệ thống * Hệ thống kiểm tra, thực hiện xóa và gửi thông báo * Kết thúc Usecase |
| Dòng sự kiện phụ | **Thêm mới, sửa bài viết:**   * Hệ thống thông báo thông tin nhập chưa đầy đủ. * Thêm mới không thành công. |
| Hậu điều kiện | * **Trường hợp thành công:** Thêm mới, sửa, xóa thông tin bài viết thành công, được lưu vào hệ thống * **Trường hợp thất bại:** Hệ thống thông báo lỗi, thông tin không được lưu |

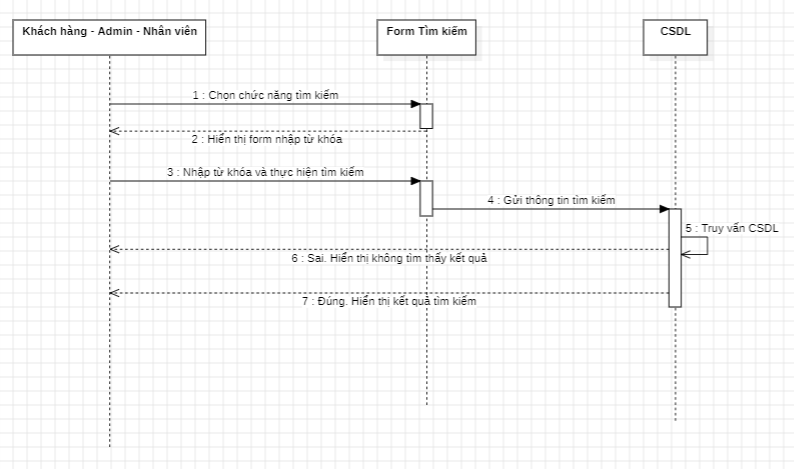
### **3.4.3. Biểu đồ tuần tự**

#### 3.4.3.1. Biểu đồ tuần tự đăng nhập của Khách hàng



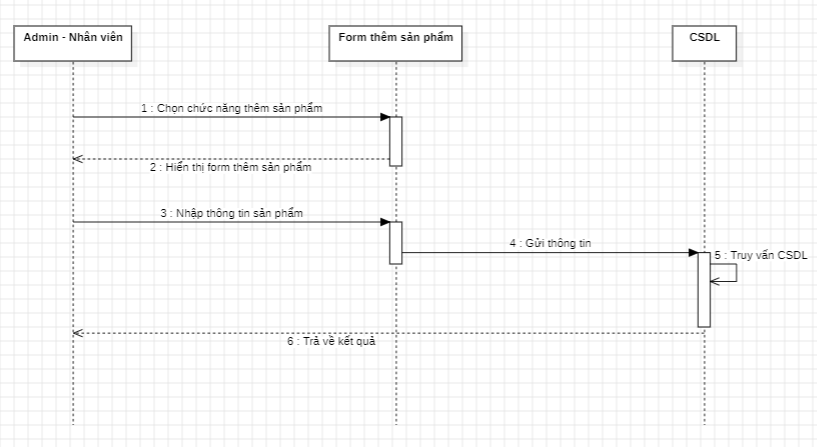
Hình 3.18: Biểu đồ tuần tự đăng nhập của Khách hàng

#### 3.4.3.2. Biểu đồ tuần tự tìm kiếm



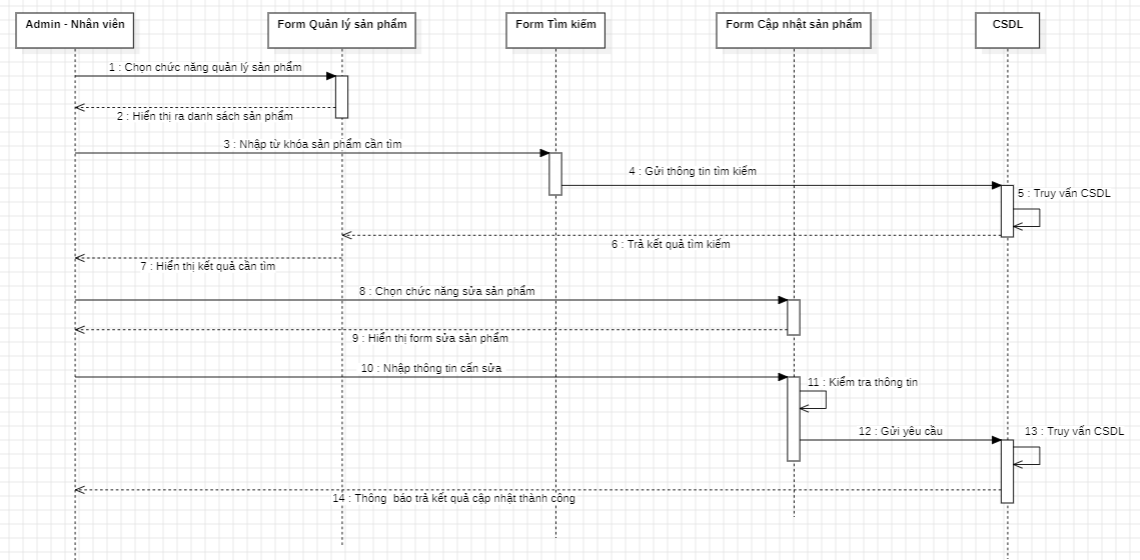
Hình 3.19: Biểu đồ tuần tự tìm kiếm

#### 3.4.3.3. Biểu đồ tuần tự thêm sản phẩm



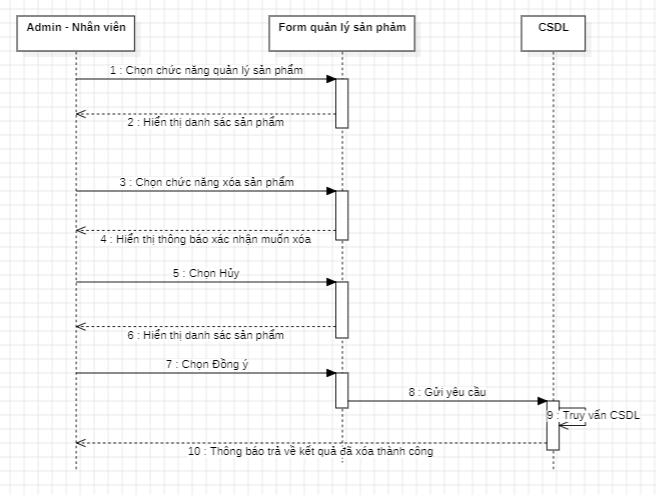
Hình 3.21: Biểu đồ tuần tự thêm sản phẩm

#### 3.4.3.4. Biểu đồ tuần tự sửa sản phẩm



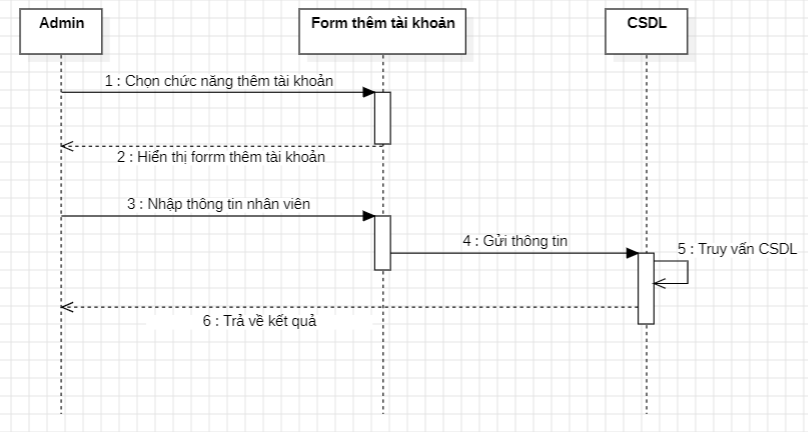
Hình 3.22: Biểu đồ tuần tự sửa sản phẩm

#### 3.4.3.5. Biểu đồ tuần tự xóa sản phẩm



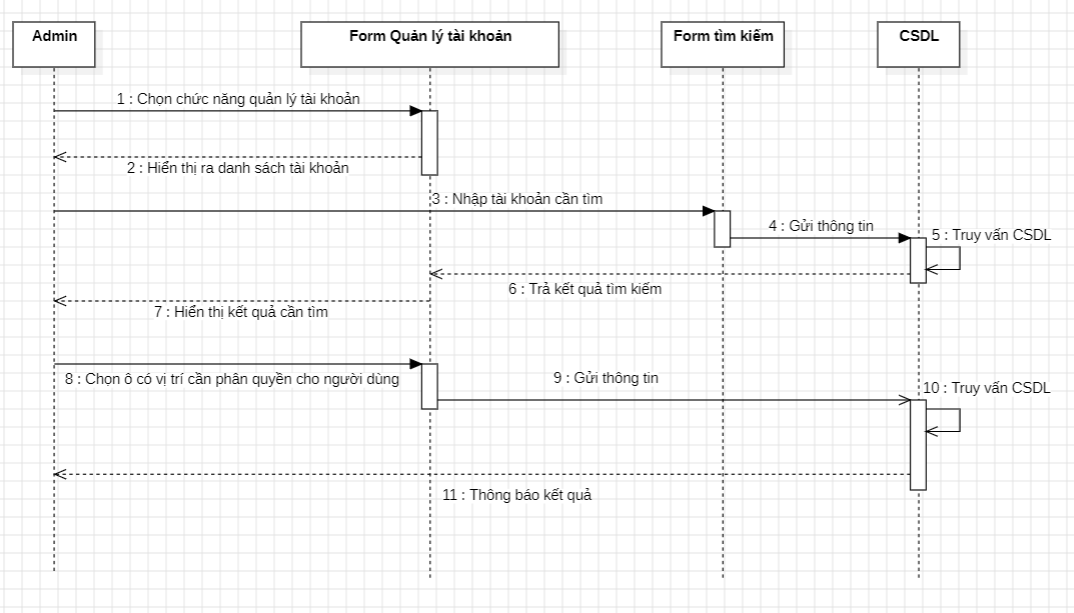
Hình 3.23: Biểu đồ tuần tự xóa sản phẩm

#### 3.4.3.6. Biểu đồ tuần tự thêm tài khoản



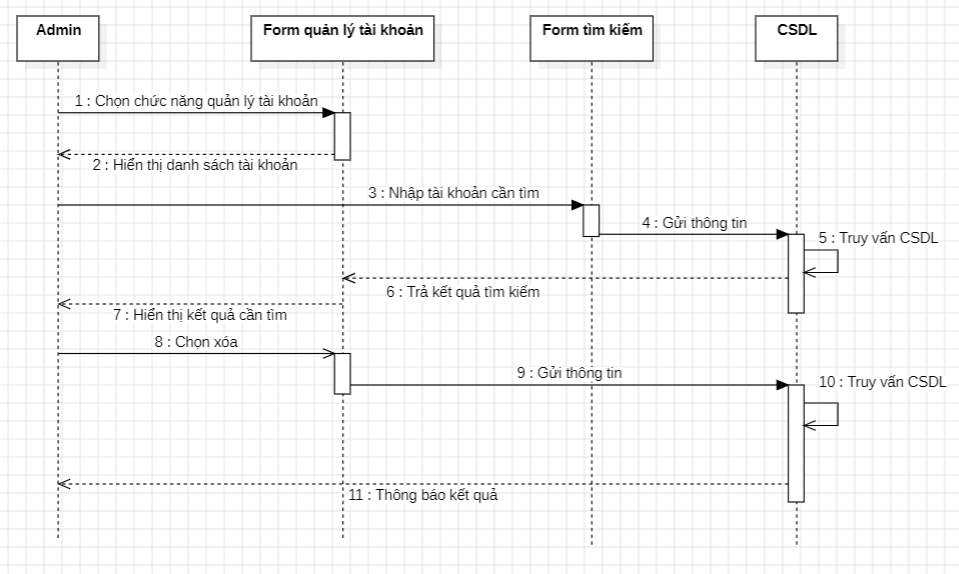
Hình 3.24: Biểu đồ tuần tự thêm tài khoản

#### 3.4.3.7. Biểu đồ tuần tự phân quyền



Hình 3.25: Biểu đồ tuần tự phân quyền

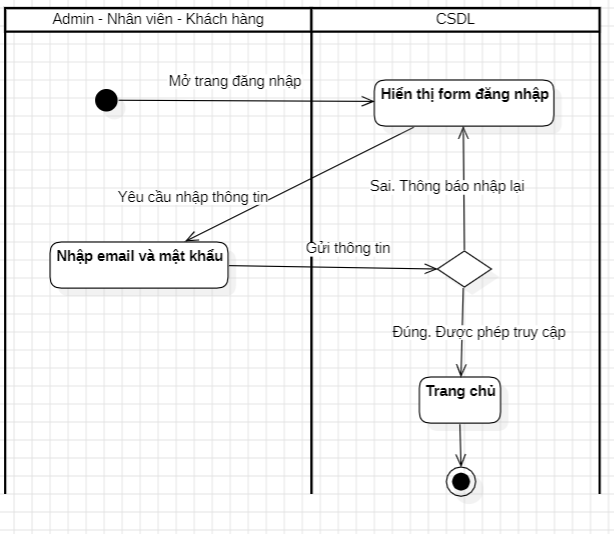
#### 3.4.3.8. Biểu đồ tuần tự xóa tài khoản



Hình 3.26: Biểu đồ tuần tự xóa tài khoản

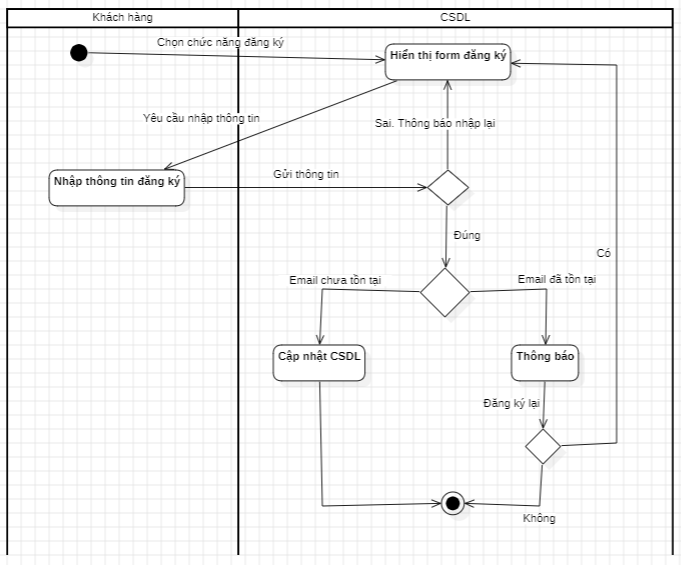
**3.4.4. Biểu đồ hoạt động**

#### 3.4.4.1. Biểu đồ hoạt động đăng nhập



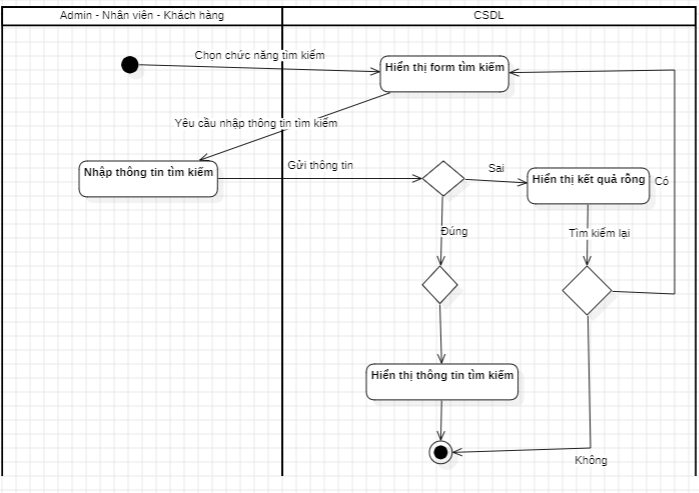
Hình 3.30: Biểu đồ hoạt động đăng nhập

#### 3.4.4.2. Biểu đồ hoạt động đăng ký



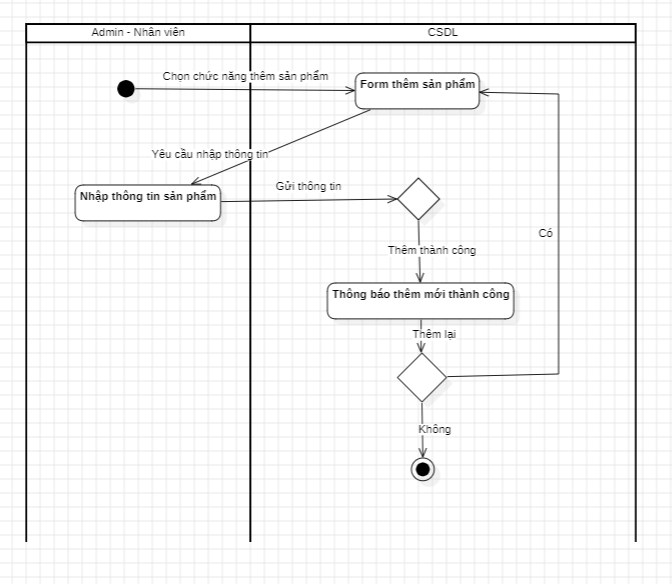
Hình 3.31: Biểu đồ hoạt động đăng ký

#### 3.4.4.3. Biểu đồ hoạt động tìm kiếm



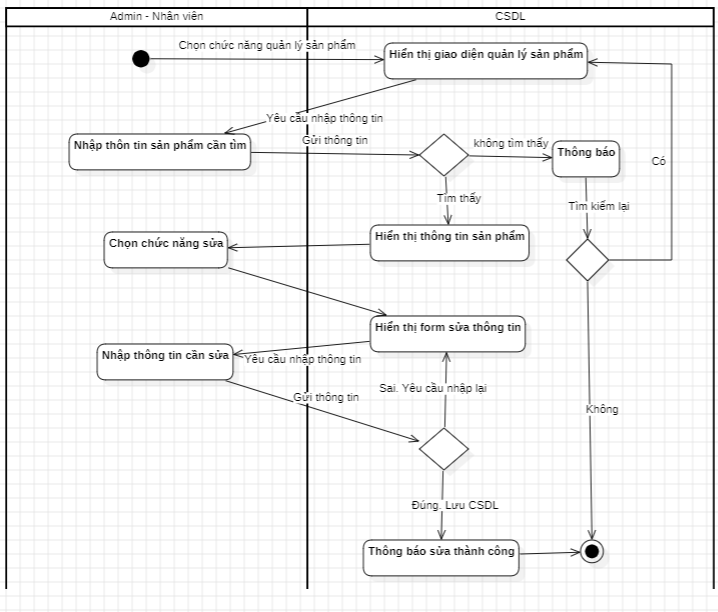
Hình 3.32: Biểu đồ hoạt động tìm kiếm

#### 3.4.4.4. Biểu đồ hoạt động thêm sản phẩm



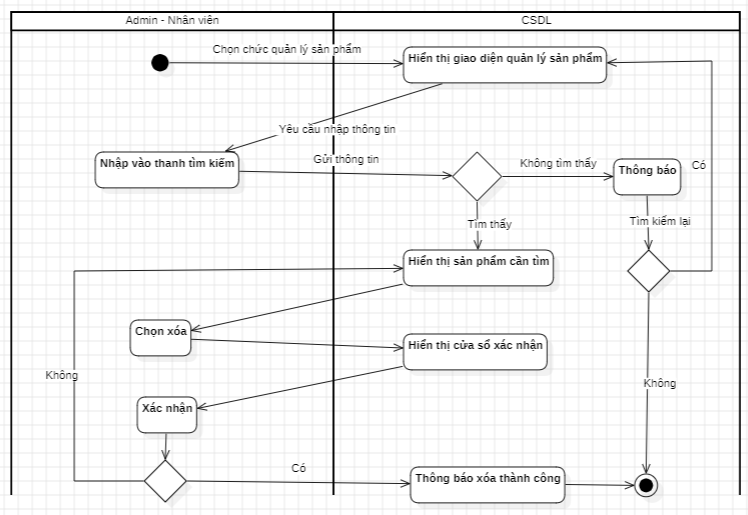
Hình 3.33: Biểu đồ hoạt động thêm sản phẩm

#### 3.4.4.5. Biểu đồ hoạt động sửa sản phẩm



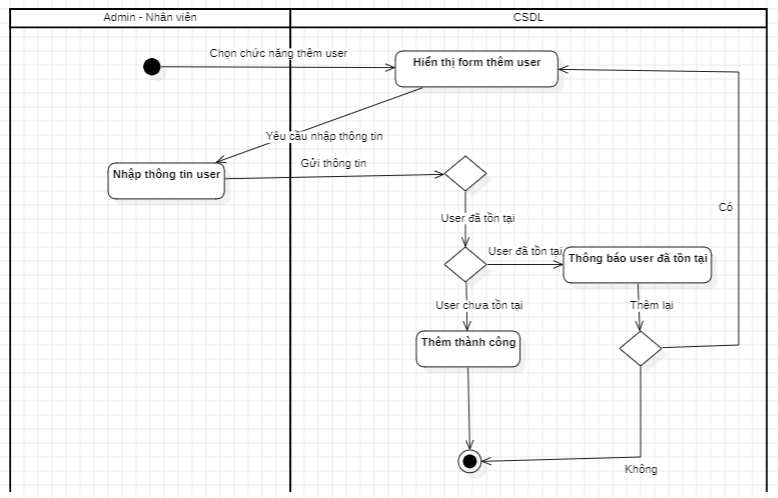
Hình 3.34: Biểu đồ hoạt động sửa sản phẩm

#### 3.4.4.6. Biểu đồ hoạt động xóa sản phẩm



Hình 3.35: Biểu đồ hoạt động xóa sản phẩm

#### 3.4.4.7. Biểu đồ hoạt động thêm user



Hình 3.36: Biểu đồ hoạt động thêm user

#### 3.4.4.8. Biểu đồ hoạt động xóa user



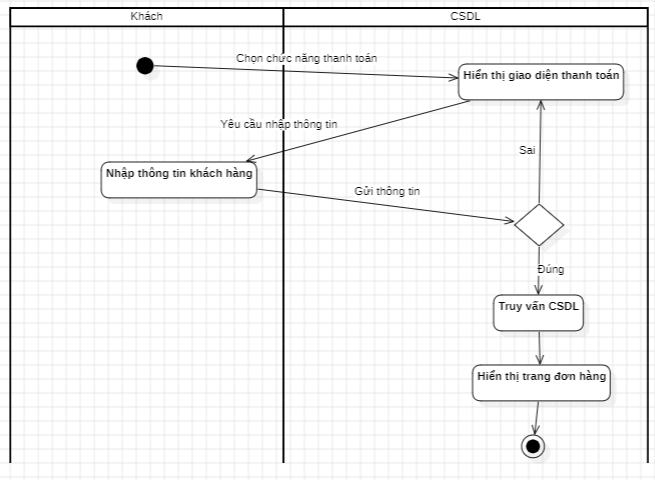
Hình 3.37: Biểu đồ hoạt động xóa user

#### 3.4.4.9. Biểu đồ hoạt động phân quyền user



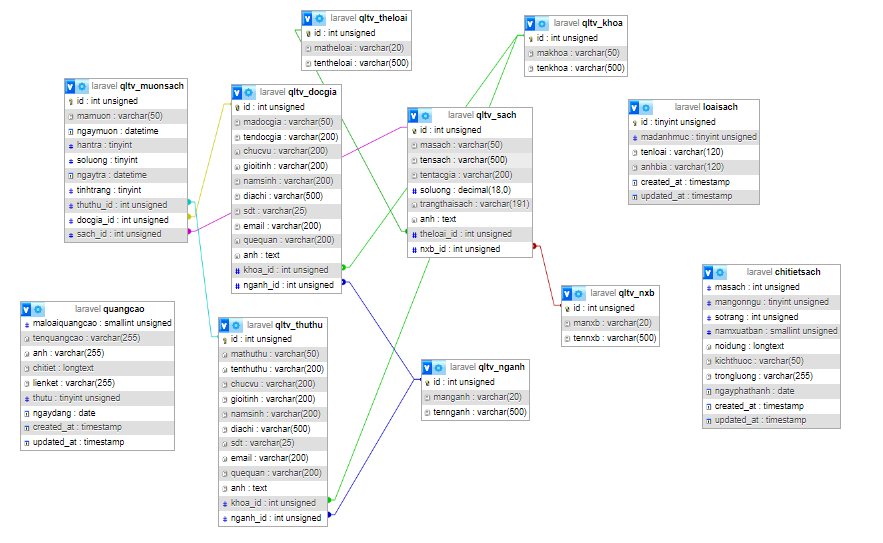
Hình 3.38: Biểu đồ hoạt động phân quyền user

#### 3.4.4.10. Biểu đồ hoạt động phân quyền user



Hình 3.39: Biểu đồ hoạt động thanh toán

### **3.4.5. Biểu đồ Class**



Hình 3.40: Biểu đồ Class

**CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH**

## **4.1. Xây dựng cơ sở dữ liệu**

Bảng 4.1: Bảng chitietsach

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Chú thích** |
| masach | Mã sách | INT | Khóa chính |
| noidung | Nội dung | VARCHAR |  |
| sotrang | Số trang | int |  |
| namxuatban | Năm xuất bản | smallint |  |
| noidung | Nội dung | longtext |  |
| kichthuoc | Kích thước | VARCHAR |  |
| trongluong | Trọng Lượng | VARCHAR |  |

Bảng 4.2: Bảng Độ Giả

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Chú thích** |
| madocgia | Mã độc giả | INT | Khóa chính |
| tendocgia | Tên độc giả | TEXT |  |
| chucvu | Chức vụ | VARCHAR |  |
| gioitinh | Giới tính | VARCHAR |  |
| namsinh | Năm sinh | TEXT |  |
| diachi | Địa chỉ | INT |  |
| sdt | sdt | VARCHAR |  |
| email | email | VARCHAR |  |
| quequan | Quê quán | VARCHAR |  |
| anh | ảnh | Text |  |
| khoa\_id | khoa | int |  |
| Nganh\_id | ngành | int |  |

Bảng 4.3: Bảng Mượn Sách

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Chú thích** |
| mamuon | Mã mượn | INT | Khóa chính |
| ngaymuon | Ngày mượn | INT | Khóa phụ |
| hantra | Hạn trả | VARCHAR |  |
| soluong | Số lượng | VARCHAR |  |
| ngaytra | Ngày trả | VARCHAR |  |
| tinhtrang | Tình trạng | INT |  |
| thuthu\_id | Thủ thư | INT |  |
| docgia\_id | Độc giả | VARCHAR |  |
| sach\_id | sách | INT |  |

Bảng 4.4: Bảng User

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Chú thích** |
| id | Mã | INT | Khóa chính |
| name | Tên | VARCHAR |  |
| email | Email | VARCHAR |  |
| quyen\_tk | Quyền | int |  |
| password | Mật khẩu | VARCHAR |  |

Bảng 4.5: Bảng Sách

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Chú thích** |
| id | Id sách | INT | Khóa chính |
| masach | Ma sách | VARCHAR |  |
| tensach | tên | VARCHAR |  |
| tentacgia | Tên tác giả | VARCHAR |  |
| soluong | Số lượng | decimal |  |
| trangthaisach | Trạng thái | VARCHAR |  |
| anh | ảnh | TEXT |  |
| theloai\_id |  | int |  |
| nxb\_id |  | int |  |

Các bảng còn lại … chứa các thuộc tính tương tự và có chứa các khóa phụ để liên kết giữa các bảng có nhiều quan hệ.

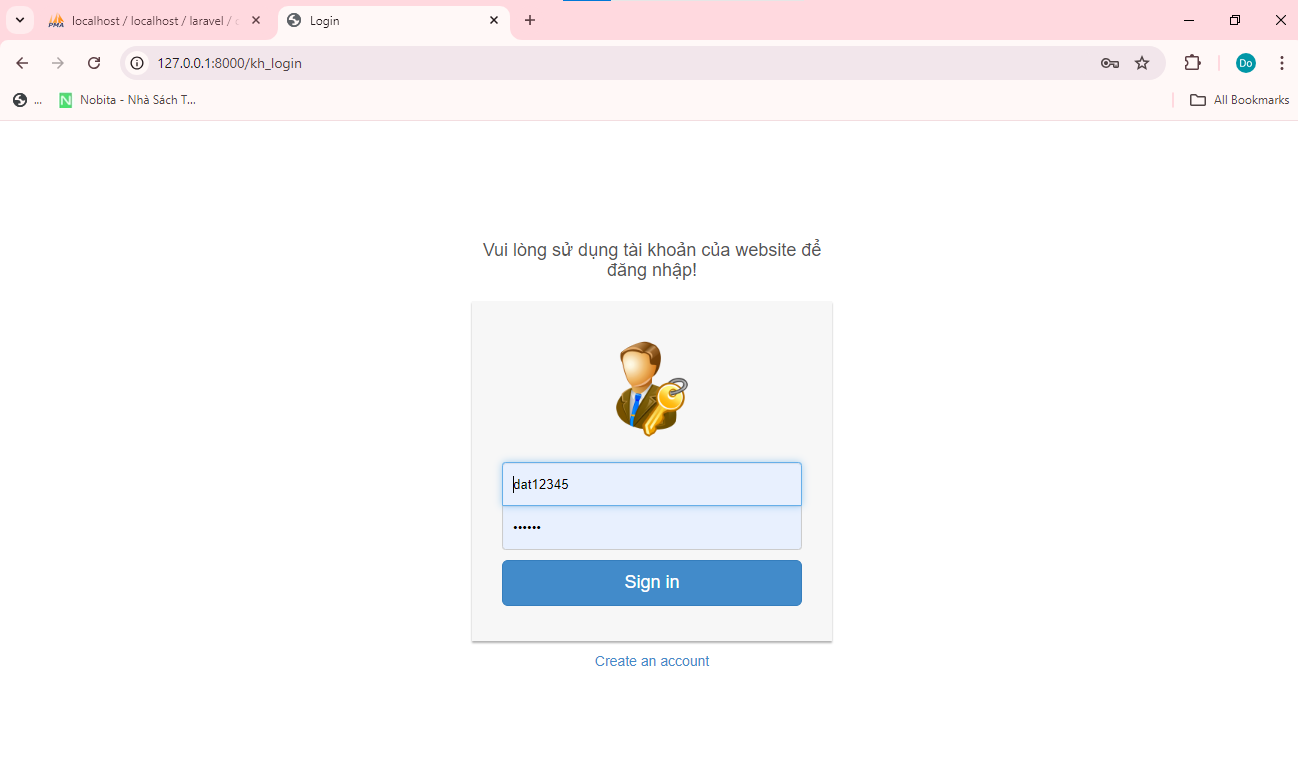
## **4.2. Xây dựng giao diện và chức năng hệ thống**

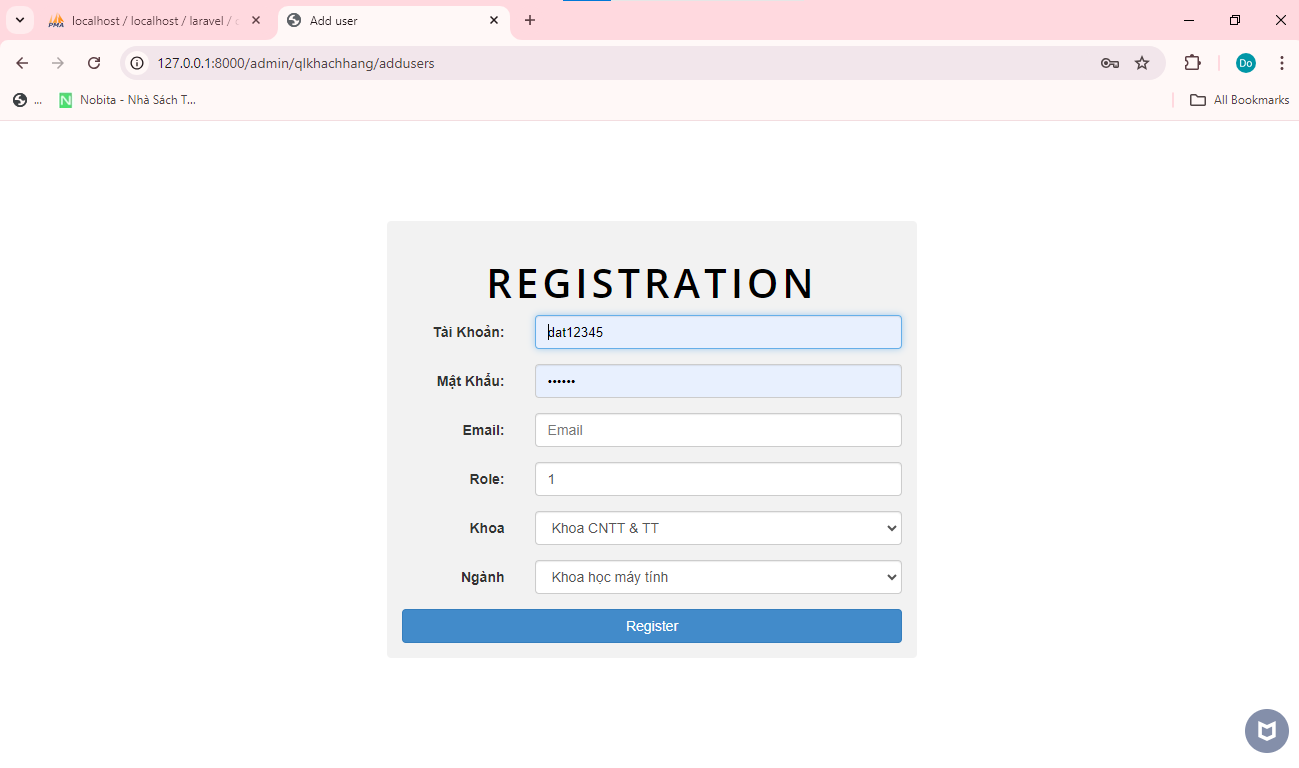
### **4.2.1. Giao diện người dùng**

#### 4.2.1.1. Trang chủ

Từ giao diện trang chủ người dùng có thể thực hiện các chức năng sau:

* Đăng nhập/đăng ký

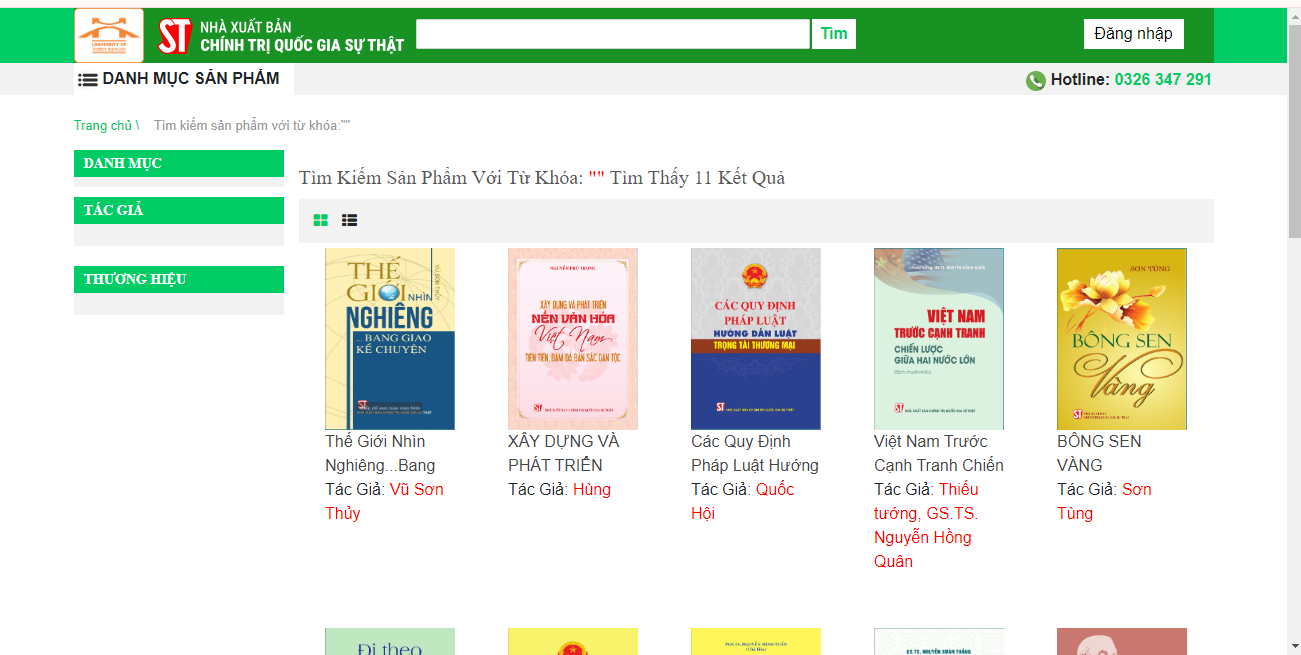




* Xem các sản phẩm trong hệ thống



* Tìm kiếm sản phẩm trong hệ thống



* Các thể loại đang có

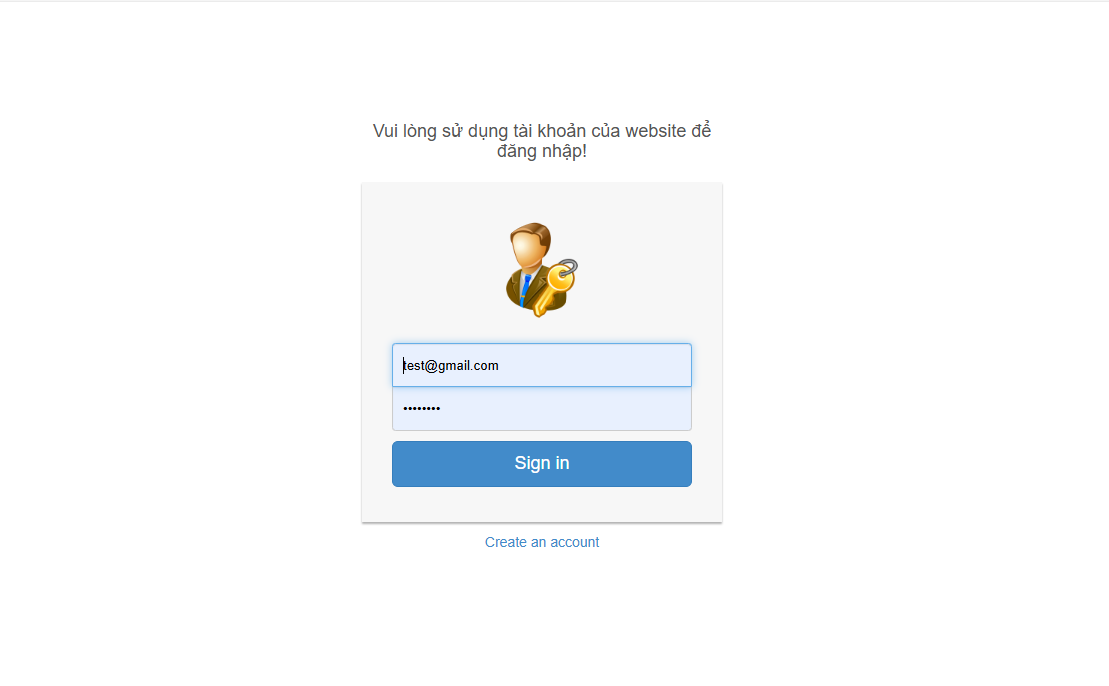


#### Trang đăng nhập/đăng ký

Tại trang Đăng ký, người dùng phải nhập đúng với thông tin cần thiết để có thể tạo ra một tài khoản thành viên bao gồm:

* Nhập họ tên của khách hàng.
* Phần nhập Email phải nhập đúng định dạng email.
* Tài khoản đăng ký không được trùng email với những tài khoản đã được đăng ký trước đấy trên hệ thống, nếu trùng hệ thống sẽ đưa ra thông báo trùng email.
* Mật khẩu có thể nhập tùy ý khách hàng.

Sau khi đăng ký thành công, khách hàng sẽ nhập thông tin vào form Đăng nhập ở trên những thông tin đã đăng ký bao gồm email và mật khẩu.



Hình 4.3: Giao diện người dùng đăng nhập/đăng ký

-

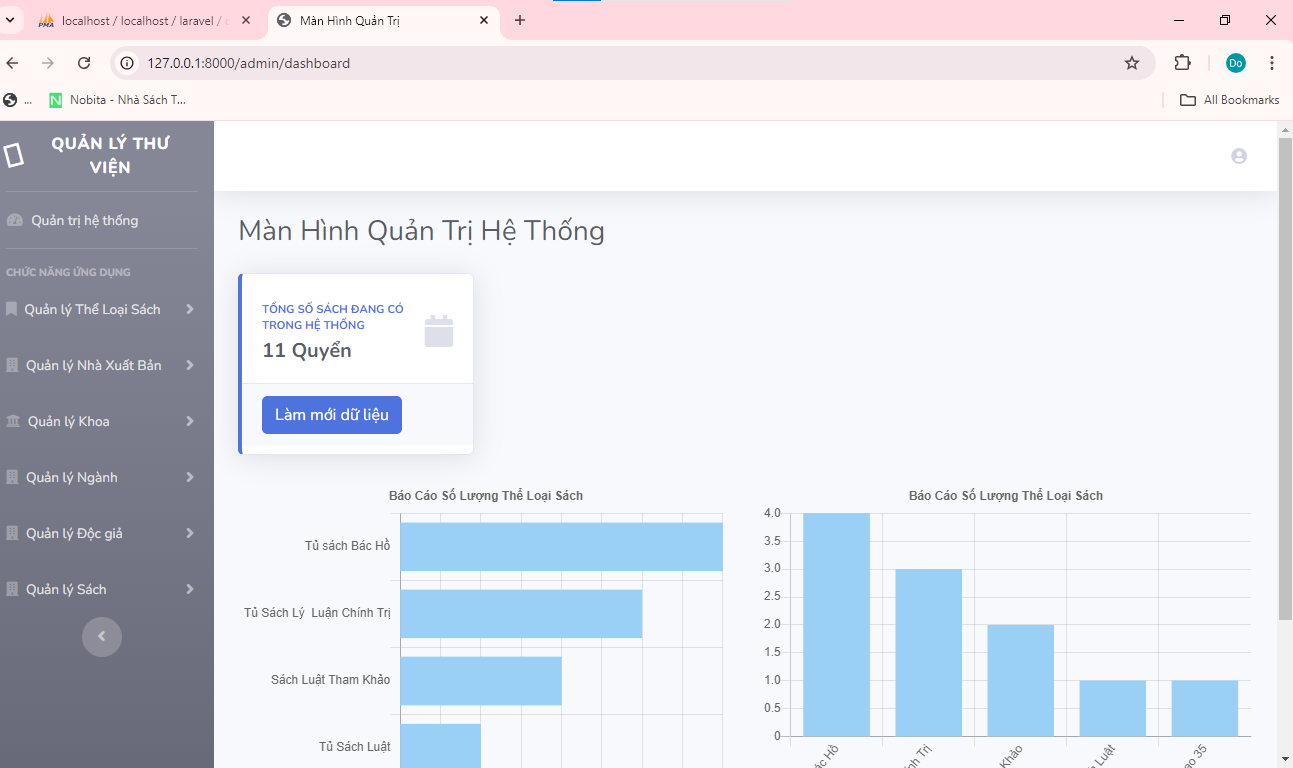
### **Giao diện và chức năng quản trị**

#### 4.2.2.1. Giao diện trang quản trị Dashboard

Tại trang quản trị, quản trị viên và nhân viên có thể thực hiện các chức năng sau:

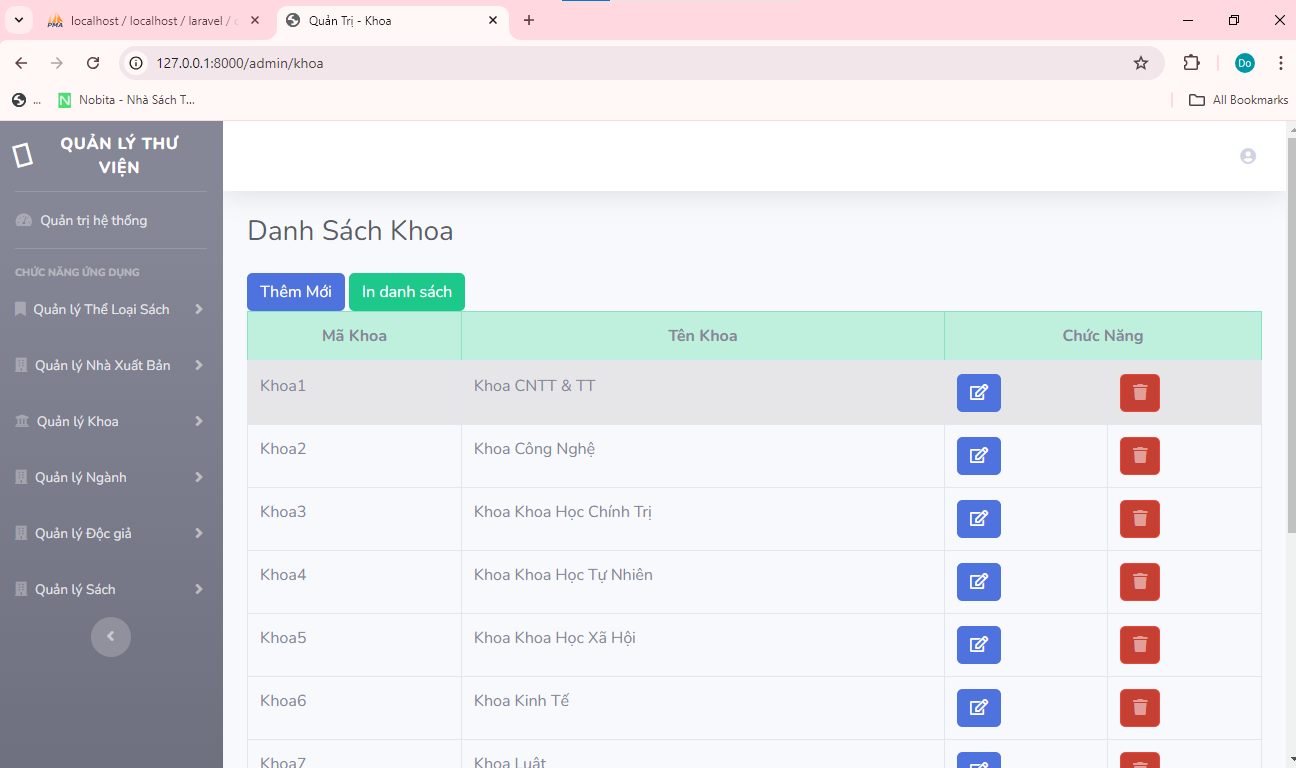
* Xem thống kê ở trang dashboard
* Quản lý slide
* Quản lý danh mục sản phẩm
* Quản lý sách
* Quản lý thủ thư
* Quản lý mượn sách
* ………….

Ngoài ra, có thể xem được thống kê số lượng đơn hàng, sản phẩm, khách hàng, bài viết, xem được số lượng truy cập vào bài viết và sản phẩm.

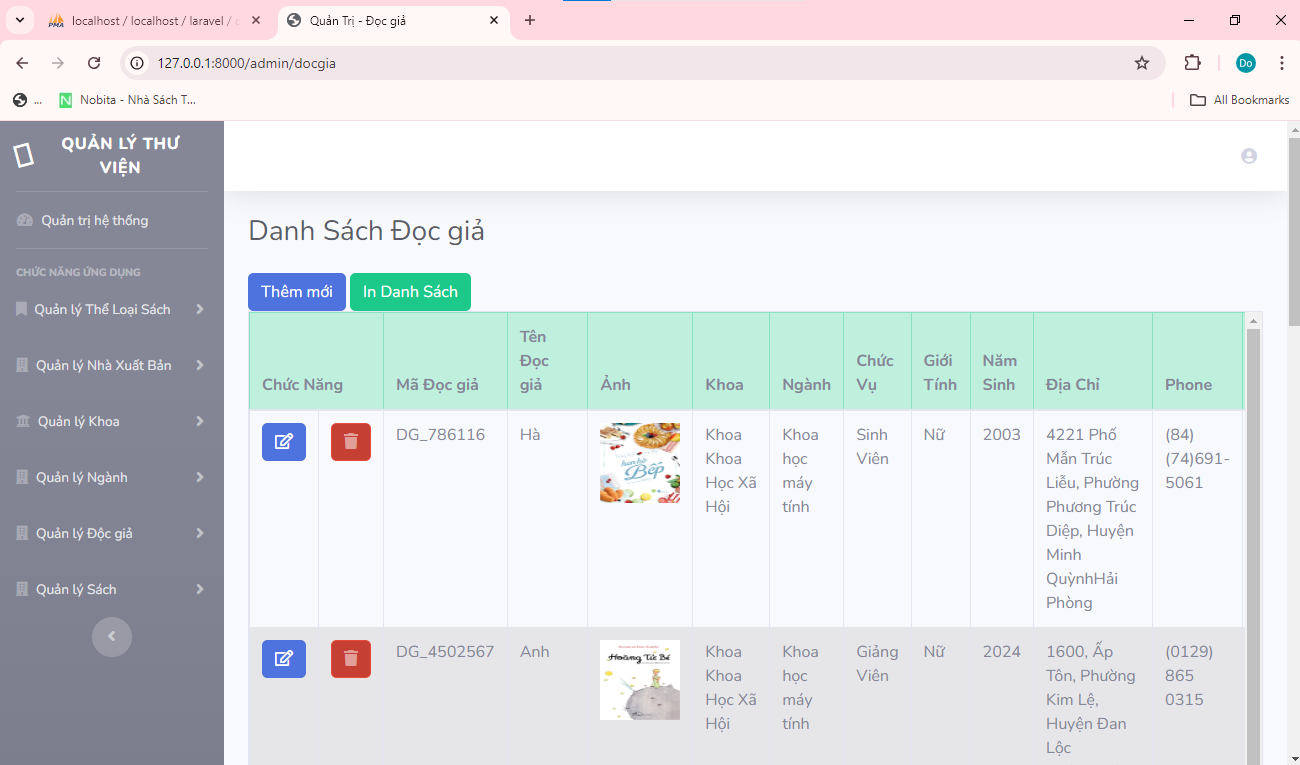


Hình 4.13: Giao diện trang quản trị Dashboard

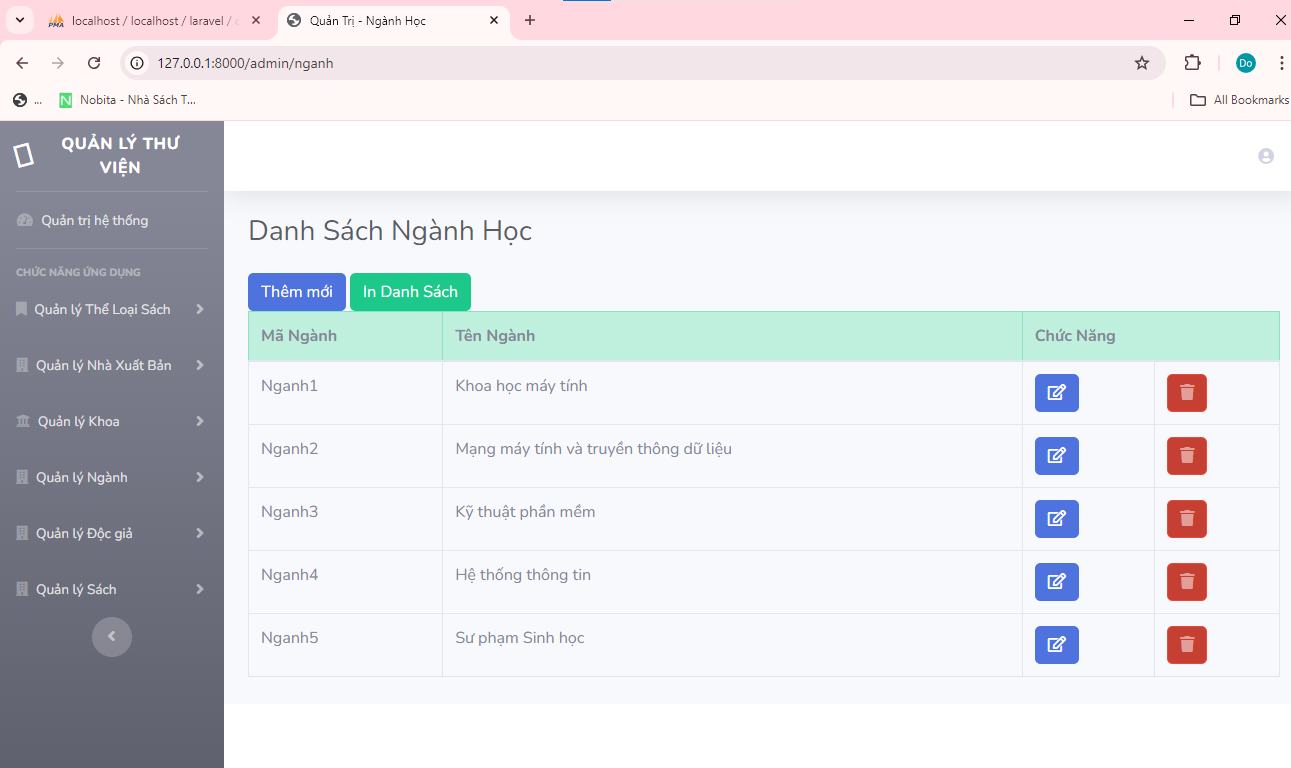
* Giao diện trang danh sách khoa



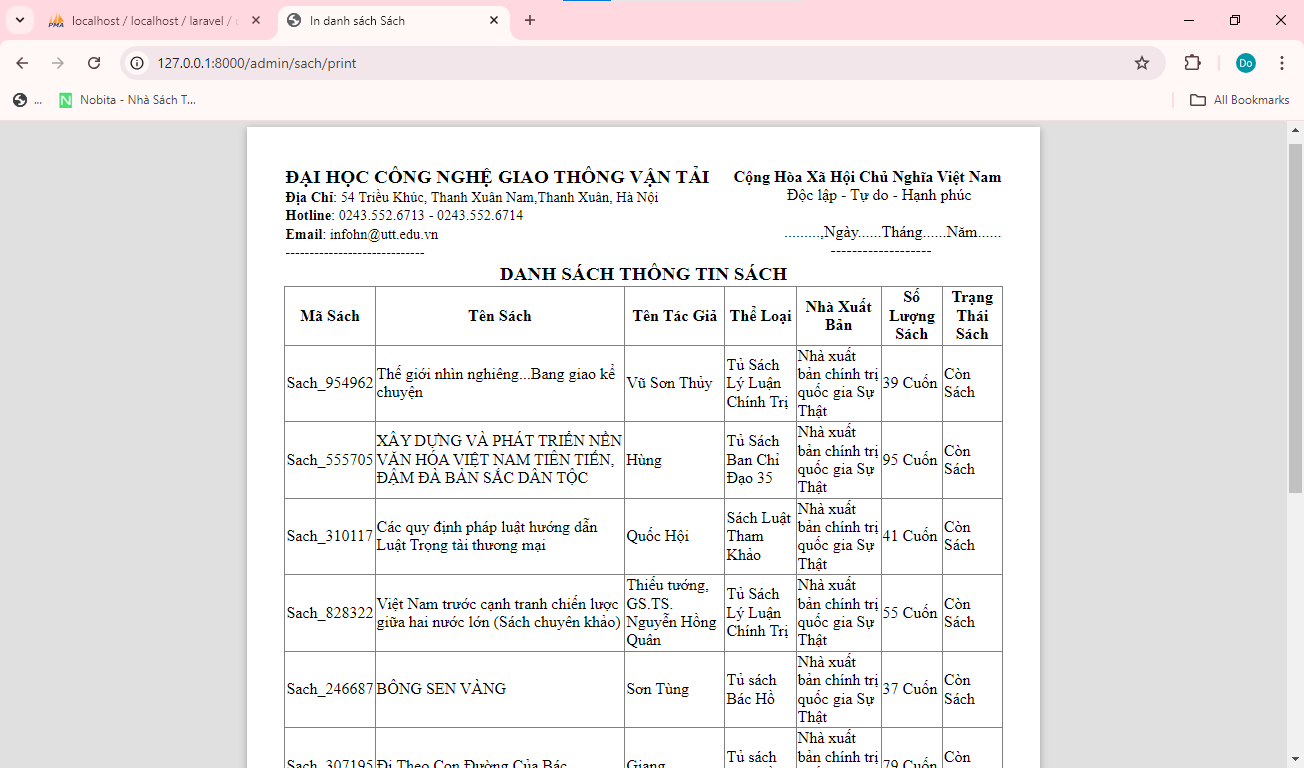
* Giao diện trang độc giả



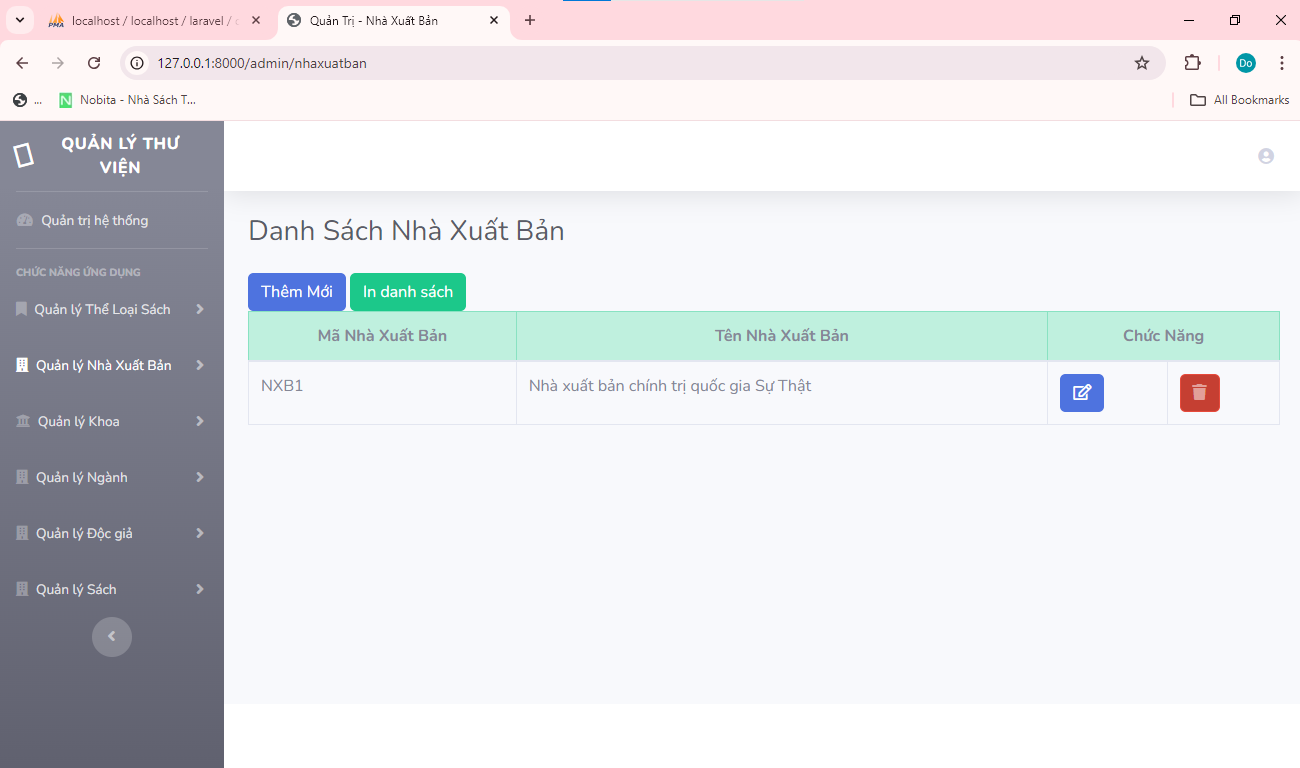
* Giao diện trang danh sách ngành



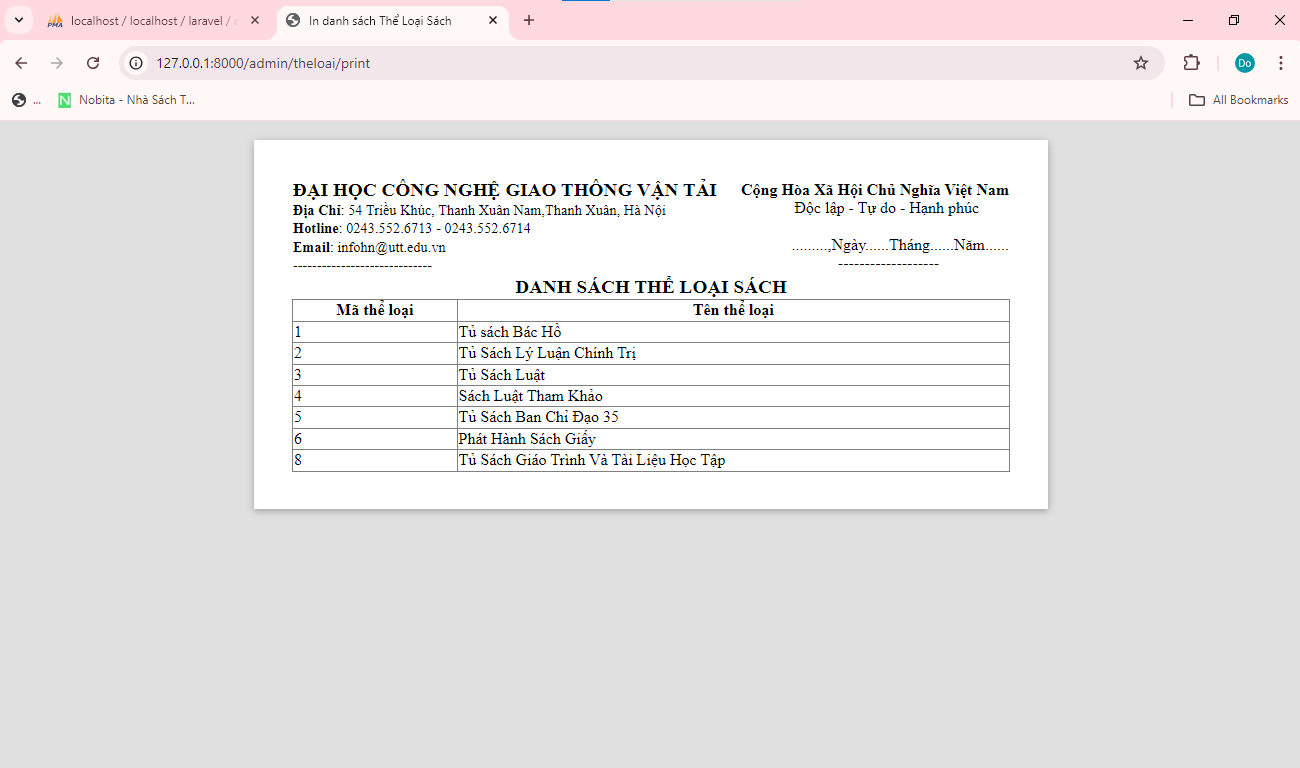
* Giao diện trang in danh sách ngành



* Giao diện trang Nhà xuất bản:

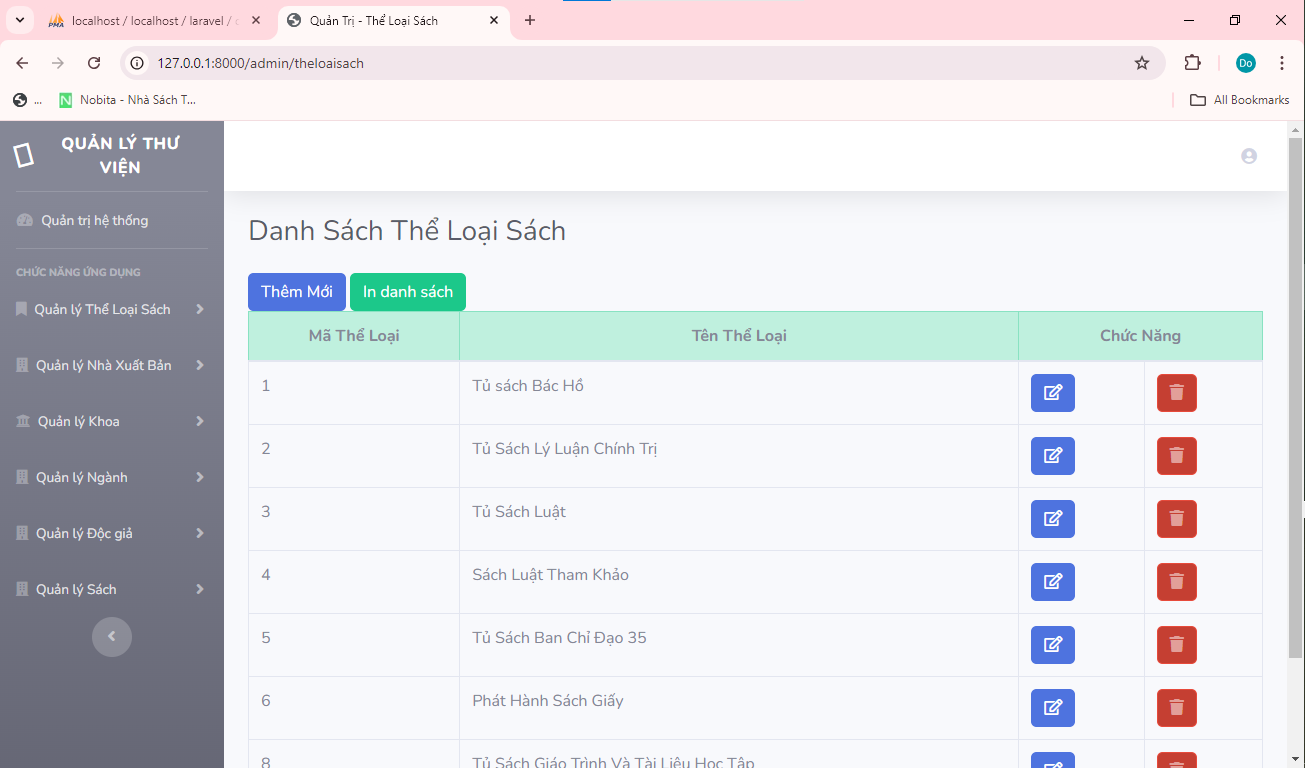


* Giao diện trang In thể loại:

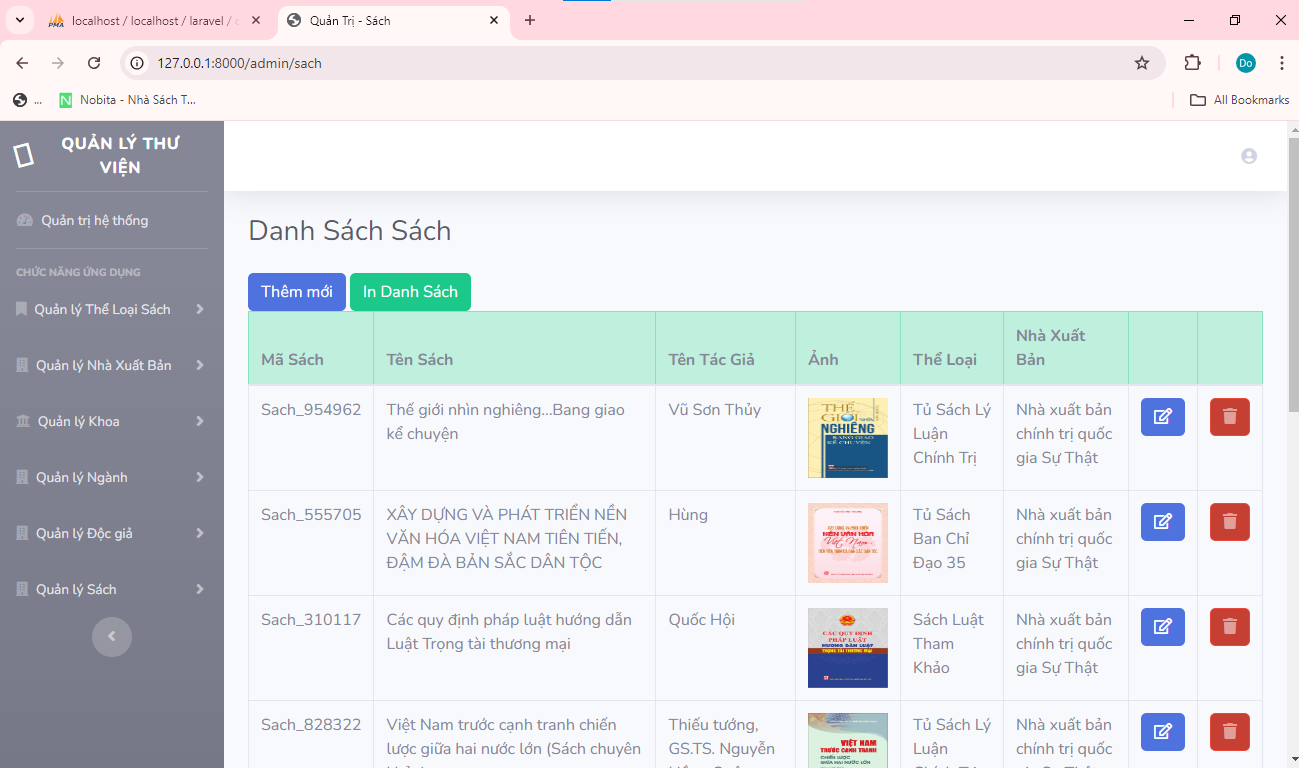


# 

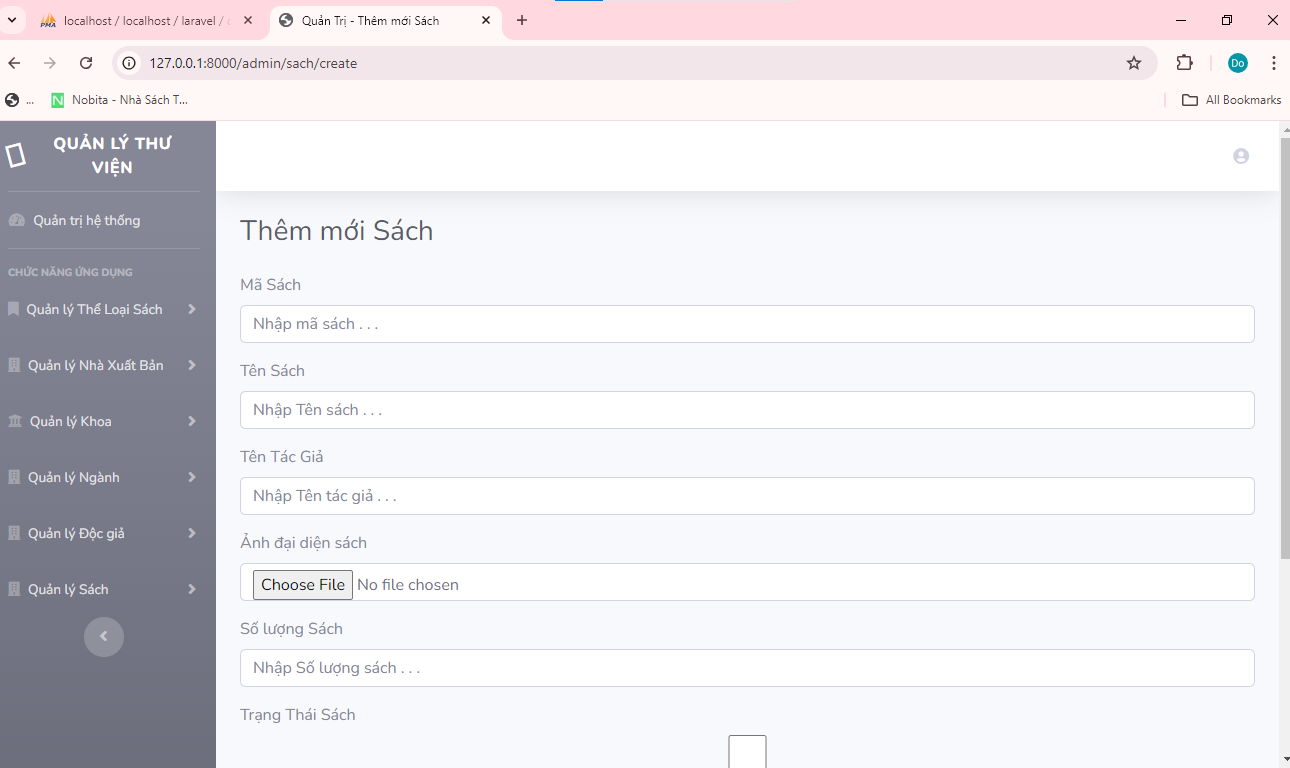
* Giao diện trang quản lý thể loại



-Giao diện trang sách:



-Giao diện trang them mới sách:



# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* 1. **Kết quả đạt được**
* Sau quá trình làm đồ án, em nắm bắt rõ hơn các kiến thức về về quản trị cơ sở dữ liệu, UML, HTML, CSS, Bootstrap và Laravel Framework là nền tảng vững chắc để xây dựng được một hệ thống website.
* Nắm rõ quá trình khảo sát, thiết kết phân tích thiết kế hệ thống, tạo thành các biểu đồ usecase, biểu đồ tuần tự, biểu đồ class, biểu đồ hành động để phục vụ quá trình xây dựng hệ thống có cơ sở chắc chắn và hiệu quả. Việc thiết kế dữ liệu và giao diện người dùng thân thiện cũng là một điểm mạnh quan trọng.
* Xây dựng được các chức năng cơ bản của website phần nào đáp ứng được nhu cầu của người dùng và quản trị.
  1. **Hạn chế**
* Thời gian hạn chế: Việc phát triển một website đòi hỏi một lượng thời gian đáng kể dẫn đến việc phát triển các chức năng còn hạn chế, chưa đầy đủ hoặc không đạt được mức độ hoàn thiện mong muốn.
* Kiến thức về người dùng còn hạn chế: Nếu không đủ thông tin từ người dùng hoặc nếu khảo sát không được thực hiện một cách đầy đủ, có thể dẫn đến việc phát triển hệ thống không đáp ứng đúng nhu cầu thực tế từ người dùng.

- Hiệu suất và tốc độ tải trang: Nếu trang web không được tối ưu hóa đúng cách , nó có thể gặp vấn đề về hiệu suất và tốc độ đọc trang. Điều này ảnh hưởng trực tiếp đến người dùng và làm giảm khả năng tương tác từ người dùng.

* 1. **Hướng phát triển**
* Tối ưu hóa SEO: Tối ưu hóa trang web để các công cụ tìm kiếm giúp tăng khả năng xuất hiện cao trong kết quả tìm kiếm.
* Quảng bá trực tuyến: Sử dụng các phương tiện truyền thông trực tuyến như quảng cáo trả tiền, quảng cáo trên mạng xã hội và nội dung chất lượng để thu hút và giữ chân khách hàng.
* Bảo trì và cập nhật: Luôn duy trì trang web để đảm bảo tính hoạt động được trơn tru và cập nhập các xu hướng mới.
  1. **Kết luận**

Với sự cố gắng không ngừng nghỉ trong việc hoàn thành đồ án và sự giúp đỡ nhiệt tình của các thầy Phan Như Minh cùng với sự hỗ trợ của bạn bè, diễn đàn lập trình trên mạng xã hội em đã hoàn thành được đề tài và hầu hết các mục tiêu được đặt ra. Tuy nhiên, trong quá trình còn có thể xảy ra nhiều thiếu sót, những chức năng chưa được tối ưu, nhiều phần code còn sơ sài, mong thầy cô thông cảm và góp ý để đề tài của em có thể hiện hơn nữa để trong tương lai có thể ứng dụng vào thực tiễn sớm nhất có thể.

# **TÀI LIỆU THAM KHẢO**

**1. Tiếng Việt**

[1] Trương Ninh Thuận & Đặng Đức Hạnh, “Giáo trình phân tích thiết kế hướng đối tượng”, nhà xuất bản Đại học Quốc gia Hà Nội, 2013.

[2] Huỳnh Văn Đức, “Giáo trình nhập môn UML”, nhà xuất bản Lao động Xã hội, 2002.

[3] Nguyễn Văn Ba, “Giáo trình phân tích và thiết kế hệ thống thông tin”, nhà xuất bản Đại học Quốc gia Hà Nội, 2005.

**2. Tiếng Anh**

[4] Lynn Beighley & Michael Morrison, “ Head first php & mysql”, nhà xuất bản

O’Reilly, 2008.

[5] [Elisabeth Robson](https://www.amazon.com/Elisabeth-Robson/e/B007AJ3ZQU/ref=dp_byline_cont_book_1), “ Head first HTML with CSS & XHTML”, nhà xuất bản

Amazon, 2005.

[6] N Douglas Crockford, “JavaScript: The Good Parts” nhà xuất bản Amazon, 2008.

**3. Trang web tham khảo**

[7] Laravel, Framwork PHP, <https://laravel.com/>

[8] Bootstrap Documentation. Đường dẫn: <https://getbootstrap.com/>

[9] CSS Tutorial. Đường dẫn: <https://www.w3schools.com/css/>