

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



## **BÁO CÁO TỐT NGHIỆP**

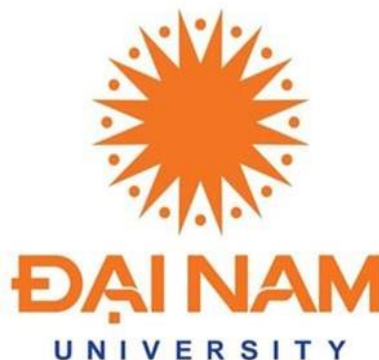
**XÂY DỰNG ỨNG DỤNG WEBSITE QUẢN LÝ THƯ VIỆN  
TẠI TRƯỜNG ĐẠI HỌC ĐẠI NAM**

<b>Sinh viên thực hiện:</b>	<b>TÔ ĐĂNG DŨNG</b>
<b>Mã sinh viên:</b>	<b>1451020044</b>
<b>Khoa:</b>	<b>Công nghệ thông tin</b>

**Hà Nội, 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

---



## **TÔ ĐĂNG DŨNG**

**XÂY DỰNG ỨNG DỤNG WEBSITE QUẢN LÝ THƯ VIỆN**  
**TẠI TRƯỜNG ĐẠI HỌC ĐẠI NAM**

CHUYÊN NGÀNH

: CÔNG NGHỆ THÔNG TIN

TIN MÃ SỐ

: 74.80.201

NGƯỜI HƯỚNG DẪN

: Th.S NGUYỄN HÀ THANH

**Hà Nội, 2024**

## LỜI CAM ĐOAN

Em Tô Đăng Dũng, dưới đây xin ký tên xác nhận rằng toàn bộ nội dung báo cáo và mã nguồn trong xây dựng đồ án tốt nghiệp đề tài: “xây dựng ứng dụng website quản lý thư viện tại trường Đại học Đại Nam” là kết quả nghiên cứu và làm việc độc lập của em.

Em cũng xin cam kết rằng không có sự sao chép hay làm dụng tài nguyên từ nguồn nào khác mà không được chỉ rõ. Mọi tài liệu tham khảo của bên thứ ba đều được em trích dẫn đầy đủ.

Trong quá trình xây dựng trang web, em đã tự tin và chân thành với sự hiểu biết và kỹ năng của mình. Khi thử sức với những điều mới trong học tập thì mọi quyết định và lựa chọn thiết kế đều dựa trên kiến thức và kinh nghiệm của em trong lĩnh vực này.

Hy vọng rằng trang web này sẽ mang lại giá trị và tiện ích cho người sử dụng và là một bước tiến quan trọng trong sự phát triển cá nhân và chuyên môn của em.

Ngày 25, tháng 5 năm 2024

Sinh viên ký tên

## LỜI NÓI ĐẦU

Trong bối cảnh xã hội hiện đại, sự phát triển mạnh mẽ của kinh tế và công nghệ thông tin đang thay đổi nhanh chóng các quy trình quản lý, đặc biệt là trong lĩnh vực giáo dục. Việc lưu trữ, quản lý và xử lý dữ liệu trong các cơ sở giáo dục, đặc biệt là thư viện, ngày càng trở nên quan trọng và cấp thiết. Các trường đại học cần những công cụ quản lý hiệu quả để nâng cao chất lượng dịch vụ và hỗ trợ hoạt động nghiên cứu, học tập.

Sự phát triển của khoa học công nghệ, đặc biệt là công nghệ thông tin, đã làm cho các giải pháp phần mềm ngày càng phổ biến và dễ tiếp cận hơn. Điều này tạo điều kiện thuận lợi cho việc hiện đại hóa các hệ thống quản lý trong nhà trường. Các công nghệ hiện đại không chỉ giúp nâng cao hiệu quả quản lý mà còn mang lại sự tiện lợi và hài lòng cho người dùng.

Nhận thấy tầm quan trọng của việc tối ưu hóa các phương pháp lưu trữ và quản lý dữ liệu trong thư viện, em đã quyết định lựa chọn đề tài "xây dựng ứng dụng website quản lý thư viện tại trường Đại học Đại Nam". Đề tài này hướng đến mục tiêu tiết kiệm thời gian, lưu trữ được nhiều dữ liệu và xử lý thông tin một cách đơn giản, nhanh chóng và hiệu quả, góp phần nâng cao chất lượng quản lý thư viện và hỗ trợ tốt hơn cho hoạt động học tập và nghiên cứu tại trường.

## LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt học kỳ này khi bắt đầu làm đồ án tốt nghiệp em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, các anh chị khóa trên và bạn bè trong và ngoài lớp.

Với lòng biết ơn sâu sắc nhất, em xin chân thành cảm ơn Thầy Nguyễn Hà Thanh đã tận tâm hướng dẫn em qua từng buổi học trên lớp, giải đáp kịp thời các thắc mắc của em. Nếu không có những lời hướng dẫn của thầy thì em nghĩ bài báo cáo của em rất khó để hoàn thành được.

Website được thực hiện trong vòng hai tháng, bước đầu sử dụng ngôn ngữ C# để viết website. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, em rất mong nhận được những ý kiến đóng góp quý báu của Thầy để cho ứng dụng cũng như kiến thức của em được hoàn thiện.

Sau cùng, kính chúc Thầy thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

## DANH MỤC HÌNH ẢNH

Hình 2.1: Hệ thống ASP.NET Core MVC và MySQL .....	5
Hình 2.2: LINQ.....	7
Hình 2.3: Truy vấn LINQ.....	10
Hình 2.4: Mô hình MVC .....	13
Hình 2.5: MySQL.....	16
Hình 3.1: Sơ đồ cơ cấu Quản lý thư viện .....	16
Hình 3.2: Sơ đồ usecase tổng quát .....	37
Hình 3.3: usecase quản lý sách.....	25
Hình 3.4: usecase quản lý độc giả .....	25
Hình 3.5: usecase quản lý phiếu mượn.....	26
Hình 3.6: usecase quản lý phiếu trả.....	26
Hình 3.7: usecase quản lý người dùng .....	27
Hình 3.8: usecase quản lý độc giả .....	27
Hình 3.9: Sơ đồ tuần tự quản lý người dùng .....	28
Hình 3.10: Sơ đồ tuần tự quản lý độc giả.....	29
Hình 3.11: Sơ đồ tuần tự quản lý phiếu trả .....	30
Hình 3.12: Sơ đồ tuần tự quản lý sách .....	31
Hình 3.13: Sơ đồ tuần tự quản lý phiếu mượn .....	32
Hình 3.14: Sơ đồ tuần tự đăng nhập.....	33
Hình 3.15: Mô hình cơ sở dữ liệu.....	34
Hình 4.1: Giao diện trang chủ .....	38
Hình 4.2: Giao diện trang quản lý sách .....	38
Hình 4.3: Giao diện trang quản lý độc giả.....	39
Hình 4.4: Giao diện trang quản lý phiếu mượn.....	39
Hình 4.5: Giao diện trang quản lý phiếu trả .....	40
Hình 4.6: Giao diện trang quản lý người dùng.....	40
Hình 4.7: Giao diện trang thống kê .....	41
Hình 4.8: Giao diện trang đăng nhập.....	41

## **DANH MỤC BẢNG BIỂU**

Bảng 3.1: Mô tả usecase tổng quát.....	23
Bảng 3.2: Danh sách tác nhân nghiệp vụ .....	24
Bảng 3.3: Mô tả bảng DOCGIA.....	34
Bảng 3.4: Mô tả bảng PHIEUMUON .....	35
Bảng 3.5: Mô tả bảng SACH.....	35
Bảng 3.6: Mô tả bảng CHITIETPHIEUMUON.....	36
Bảng 3.7: Mô tả bảng TAIKHOAN .....	37



**DANH MỤC CÁC TỪ VIẾT TẮT**

<b>STT</b>	<b>Từ viết tắt</b>	<b>Nội dung</b>
1	LINQ	Language-Integrated Query
2	ASP.NET Core MVC	ASP.NET Core Model-View-Controller
3	DI	Dependency Injection
4	TDD	Test-Driven Development
5	ACID	Atomicity, Consistency, Isolation, Durability
6	HTTP	HyperText Transfer Protocol
7	MVC	Model-View-Controller
8	URL	Uniform Resource Locator

## MỤC LỤC

CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI .....	1
1.1. Giới thiệu về đề tài .....	1
1.1.1. Lý do chọn đề tài .....	1
1.1.2. Mục đích và nhiệm vụ nghiên cứu .....	1
1.1.3. Đối tượng và phạm vi nghiên cứu .....	1
1.1.4. Phương pháp nghiên cứu .....	1
1.1.5. Đóng góp của đề tài .....	1
1.1.6. Bố cục của đề tài .....	1
1.2. Giới thiệu về thư viện trường Đại học Đại Nam.....	2
1.3. Ứng dụng công nghệ vào bài toán quản lý quy trình mượn trả sách: .....	2
1.3.1. Đăng ký mượn sách: .....	3
1.3.2. Trả sách:.....	3
1.3.3. Thống kê: .....	4
CHƯƠNG II: CƠ SỞ LÝ THUYẾT .....	5
2.1. Tổng quan về hệ thống.....	5
2.1.1. Cơ chế hoạt động: .....	5
2.1.2. Ưu điểm: .....	6
2.1.3. Nhược điểm: .....	6
2.2. LINQ: .....	7
2.2.1. Tổng quan về LINQ (Language-Integrated Query).....	7
2.2.2. Ưu điểm của LINQ .....	7
2.2.3. LINQ trong thực tiễn .....	9
2.2.4. Thực thi truy vấn trong LINQ .....	9
2.2.5. Tóm lược.....	9
2.3. ASP.NET CORE MVC.....	10
2.3.1. Giới thiệu .....	10
2.3.2. Kiến trúc MVC .....	11
2.3.3. Cơ chế và tính năng .....	11
2.3.4. Ưu điểm và nhược điểm .....	12
2.3.5. Sử dụng và ví dụ .....	12
2.4. MySQL:.....	14
2.4.1. Giới thiệu .....	14
2.4.2. Các tính năng nổi bật .....	14

2.4.3. Ưu điểm .....	14
2.4.4. Nhược điểm .....	15
2.4.5. Các trường hợp sử dụng phổ biến.....	15
2.5. Tiểu kết.....	16
CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....	17
3.1. Hiện trạng tổ chức:.....	17
3.2. Quy trình nghiệp vụ: .....	18
3.2.1 Đăng ký độc giả: .....	18
3.2.2 Mượn sách: .....	18
3.2.3 Trả sách:.....	18
3.2.4 Cập nhật tình trạng sách: .....	18
3.2.5 Quản lý thông tin độc giả và thông tin người dùng:.....	19
3.3: Phân tích yêu cầu .....	19
3.3.1. Mục đích: .....	19
3.3.2. Yêu cầu: .....	19
3.4: Mô hình usecase.....	23
3.4.1. Sơ đồ usecase tổng quát:.....	23
3.4.2. Mô tả usecase:.....	23
3.4.3. Danh sách tác nhân nghiệp vụ hệ thống: .....	24
3.4.4. Sơ đồ usecase quản lý sách:.....	25
3.4.5. Sơ đồ usecase quản lý độc giả: .....	25
3.4.6. Sơ đồ usecase quản lý phiếu mượn: .....	26
3.4.7. Sơ đồ usecase quản lý phiếu trả:.....	26
3.4.8. Sơ đồ usecase quản lý người dùng: .....	27
3.4.9. Sơ đồ usecase quản lý độc giả: .....	27
3.5. Sơ đồ tuần tự: .....	28
3.5.1. Sơ đồ tuần tự quản lý người dùng .....	28
3.5.2. Sơ đồ tuần tự quản lý độc giả .....	29
3.5.4. Sơ đồ tuần tự quản lý sách.....	31
3.5.4. Sơ đồ tuần tự quản lý phiếu mượn.....	32
3.5.4. Sơ đồ tuần tự đăng nhập .....	33
3.6: Thiết kế cơ sở dữ liệu.....	34
3.6.1. Mô hình thiết kế dữ liệu: .....	34
3.6.2. Mô tả bảng dữ liệu:.....	34
3.7. Tiểu kết.....	37

CHƯƠNG IV: CÁC KẾT QUẢ ĐẠT ĐƯỢC .....	38
4.1. Giao diện trang chủ .....	38
4.2. Giao diện trang quản lý sách .....	38
4.3. Giao diện trang quản lý độc giả .....	39
4.4. Giao diện trang quản lý phiếu mượn .....	39
4.5. Giao diện trang quản lý phiếu trả .....	40
4.6. Giao diện trang quản lý người dùng: .....	40
4.7. Giao diện trang thống kê .....	41
4.8. Giao diện trang đăng nhập .....	41

## CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI

### 1.1. Giới thiệu về đề tài

#### 1.1.1. Lý do chọn đề tài

Nhận thấy được tầm quan trọng của việc tối ưu hóa những phương pháp lưu trữ dữ liệu để dễ dàng cho việc quản lý và xử lý thông tin, em đã quyết định lựa chọn đề tài xây dựng “Website Quản Lý Thư Viện” với mong muốn sẽ tiết kiệm thời gian lưu trữ được nhiều dữ liệu và xử lý thông tin một cách đơn giản, nhanh chóng và hiệu quả.

#### 1.1.2. Mục đích và nhiệm vụ nghiên cứu

Mục đích của đề tài là khảo sát, phân tích và thiết kế chương trình quản lý sách của thư viện để phục vụ cho việc mượn, trả sách, quản lý độc giả và thống kê, báo cáo tại thư viện sao cho hiệu quả nhất.

#### 1.1.3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài này là các phương pháp và hệ thống lưu trữ, quản lý dữ liệu trong các thư viện hiện đại, đặc biệt là các thư viện Đại học Đại Nam.

#### 1.1.4. Phương pháp nghiên cứu

Nghiên cứu sử dụng các phương pháp khảo sát thực tế, phân tích hệ thống hiện tại và thiết kế chương trình theo các yêu cầu chức năng và phi chức năng đã đặt ra.

#### 1.1.5. Đóng góp của đề tài

Đề tài này đóng góp một hệ thống quản lý thư viện hiện đại, giúp các thủ thư và quản lý thư viện dễ dàng hơn trong việc quản lý sách, độc giả và các hoạt động mượn, trả sách. Đồng thời, hệ thống cũng giúp độc giả dễ dàng tiếp cận và tra cứu thông tin.

#### 1.1.6. Bố cục của đề tài

### **Chương 1: Tổng quan về đề tài**

- Giới thiệu về đề tài
- Giới thiệu về thư viện trường Đại học Đại Nam
- Ứng dụng công nghệ vào bài toán mượn trả sách

### **Chương 2: Cơ sở lý thuyết**

- Tổng quan về hệ thống
- LINQ
- ASP.NET CORE MVC
- MySQL

### **Chương 3: Phân tích và thiết kế hệ thống**

- Hiện trạng tổ chức
- Quy trình nghiệp vụ
- Phân tích yêu cầu
- Các sơ đồ use, tuần tự, cơ sở dữ liệu

### **Chương 4: Các kết quả đạt được**

## **1.2. Giới thiệu về thư viện trường Đại học Đại Nam**

Thư viện của Đại học Đại Nam đã trở thành một trung tâm tri thức vững mạnh và phong phú, là nơi tập trung hàng ngàn tài liệu và nguồn thông tin quý báu. Với diện tích rộng lớn và kiến trúc hiện đại, Thư viện Đại học Đại Nam không chỉ là nơi lưu trữ sách báo mà còn là điểm đến lý tưởng cho sinh viên, giảng viên và nhà nghiên cứu tìm kiếm kiến thức và nghiên cứu.

Thư viện được trang bị các hệ thống công nghệ tiên tiến như hệ thống quản lý tài liệu số, máy tính và kết nối internet tốc độ cao, giúp cho việc tìm kiếm thông tin trở nên dễ dàng và thuận tiện hơn bao giờ hết. Ngoài ra, Thư viện còn có không gian làm việc yên tĩnh và thoáng đãng, các phòng học nhóm và cá nhân, đảm bảo môi trường học tập hiệu quả và sáng tạo cho cộng đồng sinh viên.

Với sứ mệnh trở thành trái tim của hệ thống giáo dục và nghiên cứu của trường Đại học Đại Nam, Thư viện không chỉ đóng vai trò quan trọng trong việc cung cấp tài nguyên tri thức mà còn là nơi gặp gỡ, trao đổi ý kiến và tạo ra những ý tưởng mới mẻ cho sự phát triển của cộng đồng trí thức.

## **1.3. Ứng dụng công nghệ vào bài toán quản lý quy trình mượn trả sách:**

Một thư viện cần quản lý việc đăng ký mượn và trả sách của các bạn đọc giả. Sau đây là phân mô tả theo các nghiệp vụ hàng ngày của thư viện:

Thủ thư gọi sách là đầu sách. Ví dụ: nội dung đầu sách là “Đắc nhân tâm”, “Chinh phục đỉnh cao Toán học”,... Mỗi đầu sách có một mã số gọi là mã số sách để phân biệt với các đầu sách khác, mã số sách được bắt đầu bằng 2 kí tự MS và sau đó là số thứ tự của đầu sách đó (Ví dụ: MS05, MS100, MS201,...). Mỗi đầu sách có một số lượng cho biết tựa sách đó có còn sách để có thể mượn được nữa không. Ngoài ra, thông tin về đầu sách còn có: Tác giả cuốn sách, nhà xuất bản và giá tiền,... giúp cho việc phân biệt và quản lý các đầu sách được dễ dàng và chính xác hơn.

Để trở thành độc giả của thư viện, thì mỗi bạn đọc phải đăng ký và cung cấp thông tin cá nhân bao gồm họ và tên, giới tính, địa chỉ và số điện thoại của mình. Thủ thư sẽ cung cấp cho bạn đọc một thẻ độc giả, trên đó có mã số thẻ chính là mã số bạn đọc để phân biệt các bạn đọc khác. Mã độc giả được bắt đầu bằng 2 kí tự DG và sau đó là số thứ tự của độc giả đó (Ví dụ: DG11, DG32, DG99,...).

#### 1.3.1. Đăng ký mượn sách:

Sau khi bạn đọc đăng ký thẻ độc giả thành công, bạn đọc muốn mượn sách (thư viện cho mượn tối đa 3 cuốn/1 lần mượn), bạn đọc chọn cho mình những cuốn sách cần mượn rồi gặp thủ thư để tiến hành cho bước mượn sách.

Bạn đọc cần trình thẻ độc giả cho thủ thư kiểm tra thông tin và thủ thư tiến hành lập phiếu mượn.

Mỗi cuốn sách có thể được mượn tối đa 30 ngày. Nếu quá hạn, khi trả sách bạn đọc phải đóng tiền lệ phí quá hạn (5.000đ/ngày).

Ngoài ra, chương trình còn hiển thị thông tin về việc mượn trả sách của bạn đọc gồm: mã sách, tên sách, số sách mượn, thông tin ngày mượn, ngày hẹn trả và thông tin của người mượn..

Thông tin về thẻ độc giả của bạn đọc hợp lệ, thì thủ thư sẽ cho mượn sách. Phần mềm sẽ chạy và báo rằng bạn có được mượn sách hay không. Trường hợp bạn không được mượn sách thì phần mềm sẽ thông báo lý do.

#### 1.3.2. Trả sách:

Khi sách được trả, thủ thư kiểm tra mã sách trên sách. Đối chiếu đúng thông tin và tình trạng sách trả, thủ thư cập nhật thông tin vào phần quản lý phiếu trả.

Trường hợp tình trạng sách cũ, hư hỏng so với khi mượn và trường hợp làm mất sách thì bạn đọc phải đền bù theo quy định.

Sau khi hoàn thành việc trả sách, thông tin số lượng sách và tình trạng cuốn sách sẽ được cập nhật trên hệ thống. Những bạn đọc khác có thể mượn sách ngay sau đó.

#### 1.3.3. Thống kê:

Thống kê về số lượng sách.

Thống kê về số độc giả.

Thống kê về số lượng phiếu mượn.

Thống kê về số lượng phiếu trả.

Thống kê doanh thu tiền phạt.

Thống kê số lượng phiếu chưa trả.

Người quản lý sử dụng phần mềm được quyền thực hiện các chức năng sau:

Quản lý sách: thêm mới, sửa thông tin và xóa sách.

Quản lý độc giả: thêm, sửa thông tin và xóa độc giả.

Lập thẻ độc giả.

Lập phiếu mượn sách.

Lập phiếu trả sách.

Tìm kiếm sách.

Thống kê sách.

In thông tin độc giả.

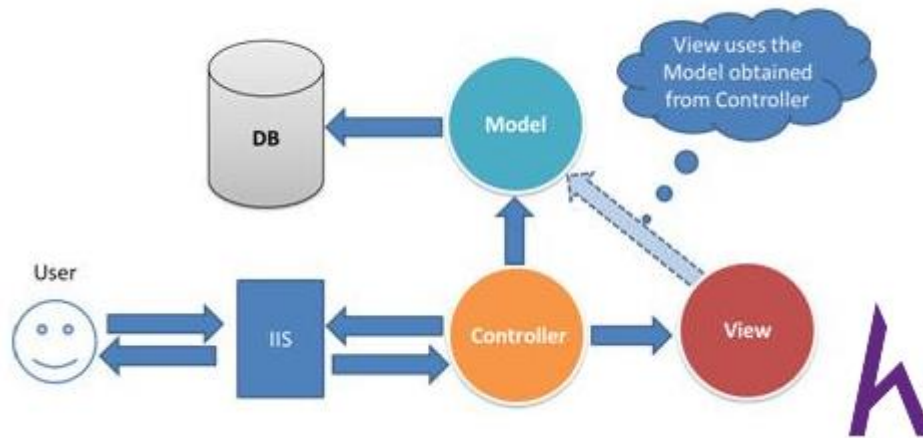
In thông tin danh mục sách.

In thông tin chi tiết việc mượn sách.



## CHƯƠNG II: CƠ SỞ LÝ THUYẾT

### 2.1. Tổng quan về hệ thống



Hình 2.1: Hệ thống ASP.NET Core MVC và MySQL

Hệ thống website sử dụng công nghệ ASP.NET Core MVC và MySQL là một lựa chọn tuyệt vời cho việc phát triển các ứng dụng web hiện đại và mạnh mẽ. ASP.NET Core MVC cung cấp một framework linh hoạt và mạnh mẽ, cho phép phát triển ứng dụng web với cấu trúc tổ chức rõ ràng và dễ dàng quản lý. Bằng cách sử dụng mô hình MVC, chúng ta có thể tách biệt logic ứng dụng, giao diện người dùng và dữ liệu, tạo điều kiện cho việc phát triển và bảo trì ứng dụng dễ dàng hơn.

MySQL, với khả năng lưu trữ dữ liệu mạnh mẽ và ổn định, là một hệ quản trị cơ sở dữ liệu lý tưởng cho các ứng dụng web. Nó cung cấp tính năng đa nhiệm, khả năng xử lý tải cao và hỗ trợ cho các transaction ACID, giúp bảo đảm tính nhất quán và an toàn của dữ liệu.

Sự kết hợp giữa ASP.NET Core MVC và MySQL mang lại cho ứng dụng web sự ổn định, hiệu suất cao và tính linh hoạt, cung cấp một nền tảng mạnh mẽ cho việc phát triển các ứng dụng web hiện đại và chuyên nghiệp.

#### 2.1.1. Cơ chế hoạt động:

ASP.NET Core MVC:

**Model:** Mô hình đại diện cho dữ liệu và logic kinh doanh của ứng dụng. Nó tương tác với cơ sở dữ liệu MySQL để truy xuất và cập nhật dữ liệu.

**View:** Giao diện người dùng mà người dùng cuối tương tác. Nó hiển thị dữ liệu từ Mô hình và tương tác với người dùng thông qua các yêu cầu HTTP.

Controller: Bộ điều khiển xử lý các yêu cầu từ người dùng, tương tác với Mô hình và chọn View phù hợp để hiển thị kết quả.

### MySQL:

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, sử dụng ngôn ngữ truy vấn cấp cao để tương tác với dữ liệu.

Nó lưu trữ dữ liệu trong các bảng có cấu trúc, cho phép truy vấn dữ liệu một cách hiệu quả thông qua các câu lệnh SQL.

### Ưu Nhược Điểm:

#### 2.1.2. Ưu điểm:

**Khả năng Mở Rộng:** ASP.NET Core MVC và MySQL đều hỗ trợ khả năng mở rộng dễ dàng, cho phép ứng dụng phát triển và mở rộng theo nhu cầu.

**Tính Linh Hoạt:** Cả hai công nghệ đều linh hoạt và có thể tùy chỉnh để đáp ứng yêu cầu cụ thể của dự án.

**Hiệu Suất:** MySQL cung cấp hiệu suất cao và ổn định cho việc lưu trữ dữ liệu, trong khi ASP.NET Core MVC cung cấp hiệu suất tốt cho việc xử lý các yêu cầu HTTP.

#### 2.1.3. Nhược điểm:

**Học Hỏi Ban Đầu:** Việc học cách sử dụng cả hai công nghệ có thể tốn nhiều thời gian ban đầu đặc biệt đối với những người mới bắt đầu.

**Quản Lý Phức Tạp:** Do cả hai công nghệ đều mạnh mẽ và linh hoạt, việc quản lý mã nguồn và cấu trúc dự án có thể trở nên phức tạp đối với các dự án lớn.

Tổng cộng, sự kết hợp giữa ASP.NET Core MVC và MySQL cung cấp một cơ sở vững chắc để phát triển các ứng dụng web hiện đại, mặc dù có những thách thức và yếu điểm nhất định.

## 2.2. LINQ:

### 2.2.1. Tổng quan về LINQ (Language-Integrated Query)



Hình 2.2: LINQ

LINQ (Language-Integrated Query) là một công nghệ tích hợp khả năng truy vấn trực tiếp vào ngôn ngữ lập trình C# và Visual Basic. LINQ được giới thiệu cùng với .NET Framework 3.5 và Visual Studio 2008, cung cấp một mô hình nhất quán để làm việc với dữ liệu từ nhiều nguồn khác nhau như cơ sở dữ liệu, tài liệu XML và các bộ sưu tập trong bộ nhớ.

### 2.2.2. Ưu điểm của LINQ

#### \* Cải Thiện Tính Dễ Đọc và Dễ Bảo Trì:

LINQ cho phép các nhà phát triển viết truy vấn trực tiếp trong ngôn ngữ lập trình, làm cho mã nguồn trở nên dễ đọc và dễ bảo trì hơn. Thay vì sử dụng các vòng lặp và điều kiện phức tạp, các truy vấn có thể được diễn đạt một cách rõ ràng và ngắn gọn:

```
var highEarners = employees.Where(e => e.Salary > threshold);
```

#### \* An Toàn Kiểu Dữ Liệu (Type Safety):

LINQ được tích hợp hoàn toàn vào C#, tận dụng hệ thống kiểu tĩnh của ngôn ngữ này. Điều này có nghĩa là các lỗi có thể được phát hiện tại thời gian biên dịch thay vì thời gian chạy, giúp mã nguồn trở nên chắc chắn hơn:

```
var employeeNames = employees.Select(e => e.Name);
```

#### \* Tăng Năng Suất:

LINQ trừu tượng hóa sự phức tạp của việc truy vấn và xử lý dữ liệu, cho phép các nhà phát triển tập trung vào việc triển khai logic nghiệp vụ, dẫn đến chu kỳ phát triển nhanh hơn và năng suất cao hơn.

**\* Linh Hoạt:**

LINQ cung cấp một giao diện truy vấn đồng nhất cho nhiều nguồn dữ liệu khác nhau, bao gồm LINQ to SQL, LINQ to XML, và LINQ to Objects. Điều này cho phép các nhà phát triển chuyển đổi giữa các nguồn dữ liệu với những thay đổi tối thiểu trong logic truy vấn.

**\* Tích Hợp Mạnh Mẽ:**

LINQ là một phần của .NET Framework, có nghĩa là không cần thêm thư viện hay gói nào khác. Điều này đảm bảo sự tích hợp liền mạch với các tính năng và công nghệ khác của .NET.

### 2.2.3. Nhược điểm của LINQ

**\* Hiệu Suất:**

Mặc dù LINQ cung cấp sự tiện lợi và khả năng bảo trì cao, nhưng nó có thể không hiệu quả bằng các truy vấn SQL được tối ưu hóa thủ công, đặc biệt là với các truy vấn phức tạp và lượng dữ liệu lớn.

**\* Học Cách Sử Dụng:**

Đối với những nhà phát triển mới bắt đầu với LINQ, việc học cú pháp và cách sử dụng LINQ có thể đòi hỏi thời gian và nỗ lực nhất định.

**\* Phiên Dịch Truy Vấn:**

Trong LINQ to SQL và Entity Framework, các truy vấn LINQ được chuyển thành SQL. Việc này đôi khi dẫn đến các truy vấn SQL không được tối ưu, làm giảm hiệu suất nếu không được kiểm tra và tối ưu đúng cách.

**\* Cách Thức Hoạt Động của LINQ**

LINQ hoạt động bằng cách cung cấp một tập hợp các phương thức truy vấn chuẩn (Standard Query Operators) để thao tác trên dữ liệu. Các phương thức này bao gồm Select, Where, OrderBy, GroupBy, Join, v.v. Dưới đây là một số ví dụ về cách sử dụng các phương thức này:

Select: Dùng để chiếu mỗi phần tử của một chuỗi thành một hình thức mới.

```
var employeeNames = employees.Select(e => e.Name);
```

Where: Dùng để lọc các phần tử của một chuỗi dựa trên một điều kiện.

```
var highEarners = employees.Where(e => e.Salary > threshold);
```

OrderBy: Dùng để sắp xếp các phần tử của một chuỗi theo một thứ tự cụ thể.

```
var sortedEmployees = employees.OrderBy(e => e.Name);
```

GroupBy: Dùng để nhóm các phần tử của một chuỗi có cùng một khóa.

```
var employeesByTitle = employees.GroupBy(e => e.JobTitle);
```

### 2.2.3. LINQ trong thực tiễn

LINQ đặc biệt hữu ích trong các ứng dụng thực tiễn, nơi việc thao tác và xử lý dữ liệu là phổ biến. Ví dụ, trong một ứng dụng thương mại điện tử, bạn có thể sử dụng LINQ để dễ dàng tìm ra các sản phẩm bán chạy nhất, tính toán trung bình, lọc dữ liệu dựa trên các điều kiện cụ thể, và nhiều hơn nữa. Dưới đây là một ví dụ về cách sử dụng LINQ để tìm 10 sản phẩm bán chạy nhất trong tháng vừa qua:

```
var topProducts = salesData
    .Where(sale => sale.Date >= DateTime.Now.AddMonths(-1))
    .GroupBy(sale => sale.ProductId)
    .Select(group => new { Product = group.Key, Sales = group.Sum(sale =>
sale.Quantity) })
    .OrderByDescending(x => x.Sales)
    .Take(10);
```

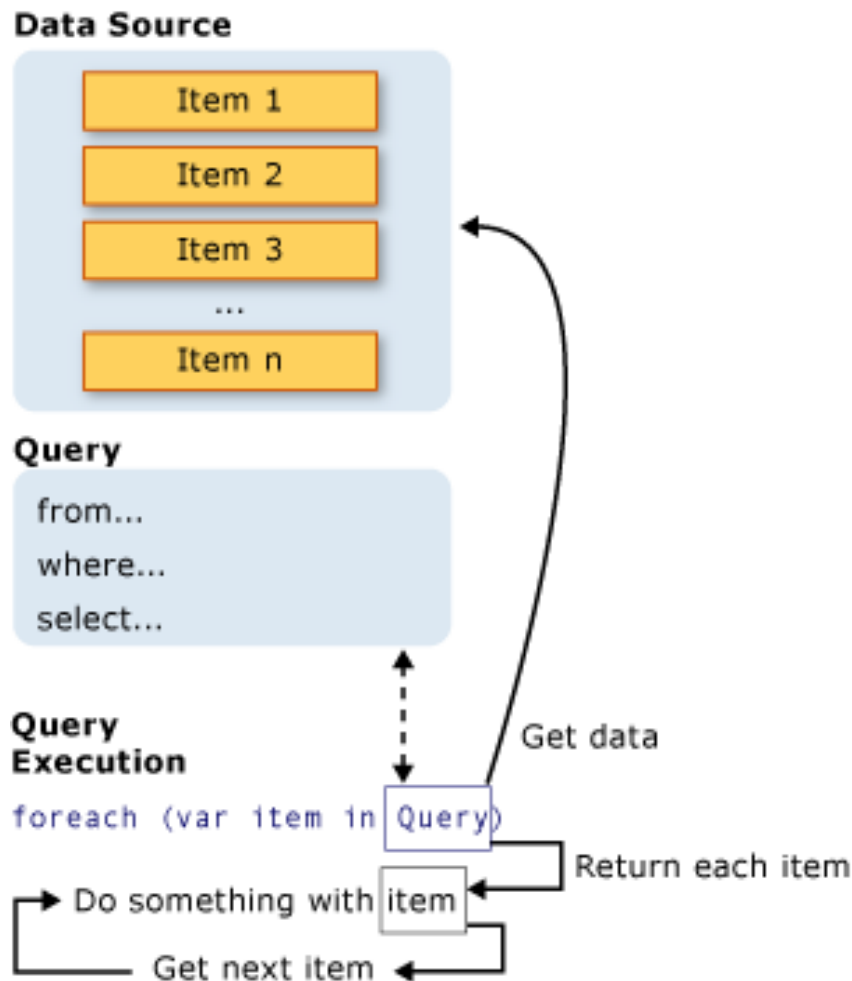
### 2.2.4. Thực thi truy vấn trong LINQ

LINQ hỗ trợ cả thực thi truy vấn trì hoãn (deferred execution) và thực thi truy vấn ngay lập tức (immediate execution). Thực thi trì hoãn chỉ thực thi truy vấn khi kết quả được yêu cầu, giúp tối ưu hóa hiệu suất trong nhiều trường hợp. Ngược lại, thực thi ngay lập tức thực thi truy vấn và lưu trữ kết quả ngay tại thời điểm truy vấn được xác định.

### 2.2.5. Tóm lược

LINQ là một công nghệ mạnh mẽ và linh hoạt, mang lại nhiều lợi ích cho các nhà phát triển bằng cách đơn giản hóa và hợp nhất các thao tác truy vấn dữ liệu. Mặc dù có một số nhược điểm về hiệu suất và yêu cầu học tập, nhưng với sự tiện lợi và tính bảo trì cao mà

LINQ cung cấp, nó đã trở thành một công cụ không thể thiếu trong phát triển ứng dụng .NET. Link tham khảo: [11]



Hình 2.3: Truy vấn LINQ

Link tham khảo: [4], [5], [6]

## 2.3. ASP.NET CORE MVC

### 2.3.1. Giới thiệu

ASP.NET Core MVC là một framework nhẹ, mã nguồn mở, và có khả năng kiểm tra cao, được tối ưu hóa để sử dụng với ASP.NET Core. Nó cung cấp một cách tiếp cận dựa trên mẫu để xây dựng các trang web động, cho phép phân tách rõ ràng các mối quan tâm và hỗ trợ phát triển hướng kiểm thử (TDD). ASP.NET Core MVC sử dụng các tiêu chuẩn web mới nhất và cho phép kiểm soát toàn bộ phần đánh dấu.

### 2.3.2. Kiến trúc MVC

MVC (Model-View-Controller) là một mô hình kiến trúc phần mềm chia ứng dụng thành ba thành phần chính:

- **Model:** Chứa logic nghiệp vụ và trạng thái ứng dụng. Model tương tác với cơ sở dữ liệu và xử lý các thao tác dữ liệu.
- **View:** Chịu trách nhiệm hiển thị dữ liệu và giao diện người dùng. View lấy dữ liệu từ Model và hiển thị nó cho người dùng cuối.
- **Controller:** Xử lý các yêu cầu từ người dùng, tương tác với Model để thực hiện các thao tác và quyết định View nào sẽ hiển thị.

### 2.3.3. Cơ chế và tính năng

- **Routing:** ASP.NET Core MVC sử dụng một hệ thống định tuyến mạnh mẽ để ánh xạ URL đến các hành động của controller. Có hai loại định tuyến là định tuyến theo quy ước (convention-based routing) và định tuyến theo thuộc tính (attribute routing).
- **Model Binding:** Chuyển đổi dữ liệu yêu cầu từ phía khách hàng (giá trị form, tham số query string, v.v.) thành các đối tượng mà controller có thể xử lý.
- **Model Validation:** Hỗ trợ kiểm tra dữ liệu bằng cách sử dụng các thuộc tính xác thực (data annotation) trên các đối tượng model.
- **Dependency Injection (DI):** Tích hợp sẵn trong ASP.NET Core, cho phép controllers và các thành phần khác yêu cầu các dịch vụ cần thiết thông qua constructor.
- **Filters:** Cho phép thực thi logic tiền và hậu xử lý cho các hành động của controller, giúp quản lý các mối quan tâm như xác thực và xử lý lỗi.
- **Areas:** Giúp chia ứng dụng lớn thành các nhóm chức năng nhỏ hơn, mỗi nhóm có mô hình, view, và controller riêng.
- **Web APIs:** ASP.NET Core MVC có thể được sử dụng để xây dựng các API web, cung cấp các tính năng như kiểm tra kiểu dữ liệu mạnh mẽ và tích hợp tốt với các công nghệ web hiện đại.

- **Razor View Engine:** Cho phép sử dụng cú pháp Razor để tạo ra các view động mạnh mẽ và dễ bảo trì.

#### 2.3.4. Ưu điểm và nhược điểm

##### Ưu điểm:

- **Hiệu năng cao:** ASP.NET Core MVC được thiết kế để tối ưu hóa hiệu suất.
- **Mã nguồn mở:** Cho phép cộng đồng đóng góp và mở rộng.
- **Phân tách rõ ràng các mối quan tâm:** Giúp mã nguồn dễ dàng bảo trì và mở rộng.
- **Hỗ trợ kiểm thử tốt:** Phù hợp với các phương pháp phát triển hướng kiểm thử (TDD).
- **Tích hợp các công nghệ web hiện đại:** Hỗ trợ tốt cho các tiêu chuẩn và công nghệ mới như HTML5, CSS3, và các thư viện JavaScript.

##### Nhược điểm:

- **Độ phức tạp:** Có thể phức tạp hơn để thiết lập và học so với các framework đơn giản hơn.
- **Cần kiến thức nền tảng tốt:** Yêu cầu hiểu biết sâu về MVC, các mẫu thiết kế, và ASP.NET Core.

#### 2.3.5. Sử dụng và ví dụ

Để bắt đầu với ASP.NET Core MVC, bạn cần cài đặt .NET SDK và một IDE như Visual Studio. Dưới đây là các bước cơ bản để tạo một ứng dụng ASP.NET Core MVC:

##### 1. Tạo dự án mới:

- Mở Visual Studio và chọn "Create a new project".
- Chọn "ASP.NET Core Web Application" và nhấn "Next".
- Đặt tên cho dự án và chọn "Create".
- Chọn "ASP.NET Core Web App (Model-View-Controller)" và chọn phiên bản .NET Core phù hợp.

##### 2. Thêm controller:

- Trong thư mục Controllers, thêm một controller mới.



- Ví dụ, tạo một HomeController với một phương thức Index để trả về một view.

### 3. Tạo view:

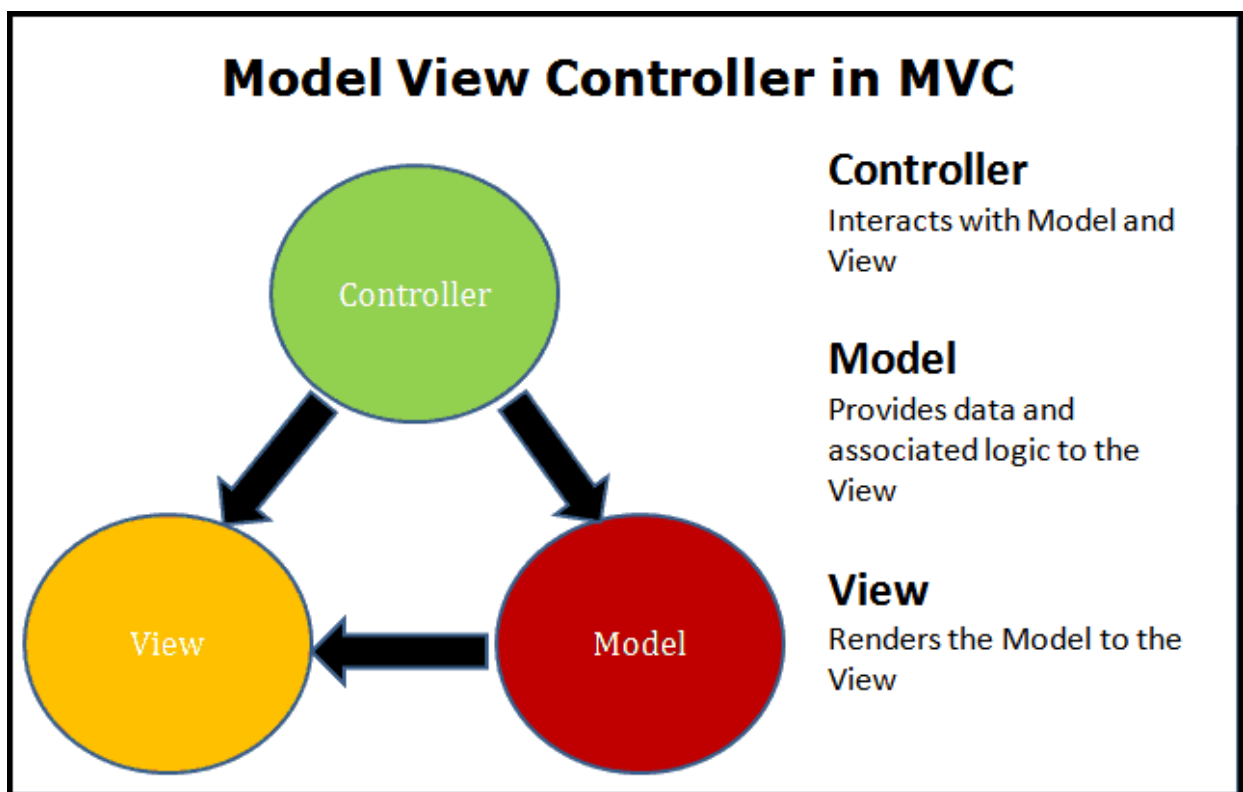
- Tạo một view cho phương thức Index trong thư mục Views/Home.
- Sử dụng cú pháp Razor để tạo nội dung HTML động.

### 4. Chạy ứng dụng:

- Nhấn Ctrl+F5 để chạy ứng dụng và kiểm tra kết quả.

Bằng cách làm theo các bước này, bạn có thể bắt đầu xây dựng các ứng dụng web động với ASP.NET Core MVC.

Link tham khảo: [7], [8], [9], [10], [11]



Hình 2.4: Mô hình MVC

## 2.4. MySQL:

### 2.4.1. Giới thiệu

MySQL là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến nhất thế giới, được phát triển bởi Michael Widenius và David Axmark từ năm 1994 và phát hành lần đầu năm 1995. MySQL nổi tiếng với hiệu suất cao, độ ổn định và dễ sử dụng, thích hợp cho các ứng dụng web và xử lý dữ liệu lớn

### 2.4.2. Các tính năng nổi bật

- **Hỗ trợ SQL:** MySQL sử dụng ngôn ngữ SQL (Structured Query Language), cho phép tạo và quản lý cơ sở dữ liệu dễ dàng cũng như thực hiện các truy vấn và thao tác dữ liệu phức tạp
- **Khả năng mở rộng:** MySQL có thể xử lý lượng dữ liệu lớn và mức độ đồng thời cao, phù hợp cho các ứng dụng web lớn
- **Sao chép (Replication):** Hỗ trợ sao chép dữ liệu giữa các máy chủ, giúp cải thiện hiệu suất và đảm bảo tính dự phòng dữ liệu
- **Stored Procedures và Triggers:** Hỗ trợ các thủ tục lưu trữ và triggers, giúp tự động hóa các tác vụ như kiểm tra và xử lý dữ liệu
- **Bảo mật dữ liệu:** MySQL cung cấp nhiều tính năng bảo mật như kiểm soát truy cập, mã hóa và kiểm toán để bảo vệ dữ liệu và tuân thủ các quy định về bảo mật dữ liệu

### 2.4.3. Ưu điểm

- **Mã nguồn mở:** MySQL là phần mềm mã nguồn mở, miễn phí sử dụng, phân phối và chỉnh sửa
- **Đa nền tảng:** MySQL có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux, và macOS
- **Hiệu suất cao:** MySQL nổi tiếng với hiệu suất cao, phù hợp cho các ứng dụng web có lưu lượng truy cập lớn và các kho dữ liệu lớn
- **Dễ sử dụng:** Giao diện đơn giản và thân thiện với người dùng
- **Cộng đồng lớn:** MySQL có một cộng đồng người dùng và nhà phát triển lớn, cung cấp nhiều tài liệu, hướng dẫn và hỗ trợ

#### 2.4.4. Nhược điểm

- **Xử lý cơ sở dữ liệu lớn kém hiệu quả:** MySQL có thể gặp khó khăn khi xử lý các cơ sở dữ liệu rất lớn
- **Công cụ phát triển và gỡ lỗi hạn chế:** Không có các công cụ phát triển và gỡ lỗi tốt như các cơ sở dữ liệu trả phí khác
- **Thiếu hỗ trợ các ràng buộc kiểm tra (check constraints):** MySQL không hỗ trợ ràng buộc kiểm tra trong SQL, gây khó khăn trong việc đảm bảo tính toàn vẹn dữ liệu
- **Đễ bị hỏng dữ liệu khi xử lý giao dịch:** MySQL có thể dễ bị hỏng dữ liệu khi xử lý các giao dịch lớn và phức tạp

#### 2.4.5. Các trường hợp sử dụng phổ biến

- **Ứng dụng web:** MySQL là lựa chọn phổ biến cho các ứng dụng web nhờ vào hiệu suất và khả năng mở rộng. Nhiều hệ thống quản lý nội dung (CMS) như WordPress, Joomla, và Drupal sử dụng MySQL
- **Kho dữ liệu:** MySQL thường được sử dụng làm kho dữ liệu do khả năng xử lý lượng dữ liệu lớn và kết hợp với các công cụ BI để phân tích dữ liệu
- **Phát triển game:** MySQL được sử dụng trong phát triển game do khả năng xử lý mức độ đồng thời cao và hỗ trợ stored procedures và triggers
- **Viễn thông:** MySQL được sử dụng rộng rãi trong ngành viễn thông để lưu trữ và quản lý dữ liệu khách hàng, thông tin thanh toán và các loại dữ liệu khác
- **Chính phủ và khu vực công:** Nhiều tổ chức chính phủ và khu vực công sử dụng MySQL để lưu trữ và quản lý dữ liệu nhạy cảm như thông tin công dân, dữ liệu tài chính và các loại dữ liệu khác



*Hình 2.5: MySQL*

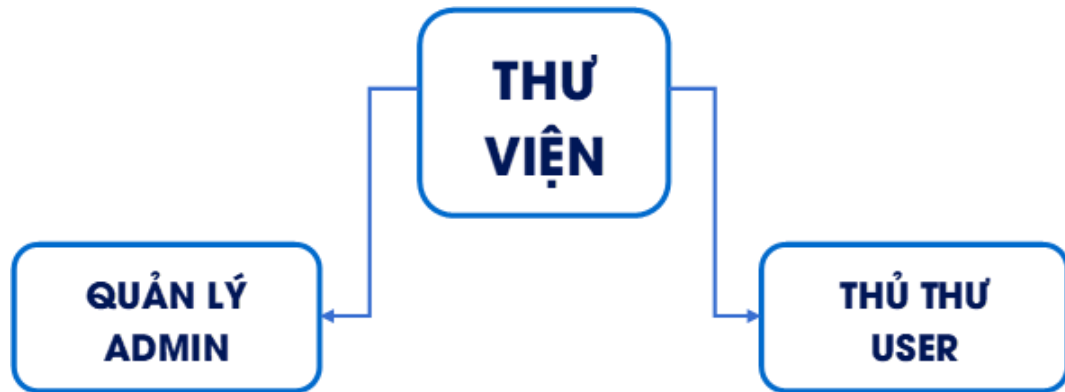
## **2.5. Tiểu kết**

Chương II đã cung cấp cái nhìn toàn diện về các khái niệm và công nghệ quan trọng cho hệ thống, bao gồm cơ chế hoạt động, ưu điểm và nhược điểm. LINQ (Language-Integrated Query) được giới thiệu cùng các ưu điểm và ứng dụng thực tiễn. NET CORE MVC được trình bày với kiến trúc, cơ chế, tính năng, ưu và nhược điểm, cùng với các ví dụ sử dụng cụ thể. MySQL được phân tích về các tính năng nổi bật, ưu và nhược điểm, cùng các trường hợp sử dụng phổ biến. Tóm lại, chương này giúp chúng ta hiểu rõ hơn về nền tảng công nghệ và chuẩn bị cho việc triển khai hệ thống hiệu quả.

Link tham khảo: [12], [13], [14]

### CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

#### 3.1. Hiện trạng tổ chức:



Hình 3.1: Sơ đồ cơ cấu Quản lý thư viện

Sơ đồ cơ cấu Quản lý thư viện

Mô tả chức năng từng vị trí:

**Quản lý – Admin:** Trực tiếp điều hành các hoạt động diễn ra trong thư viện, chịu trách nhiệm quản lý tài nguyên, dữ liệu của thư viện. Chịu trách nhiệm quản lý, hỗ trợ nhân viên hoàn thành chuyên môn và nghiệp vụ theo đúng quy trình, tránh sơ sót, rủi ro trong quá trình hoạt động của thư viện. Đây cũng là vị trí kiểm tra, theo dõi hoạt động của thư viện, quản lý thông kê và kịp thời đưa ra biện pháp xử lý khi gặp rủi ro, sự cố. Là bộ phận trực tiếp theo dõi công việc, nghiệp vụ và kiểm tra năng lực thủ thư thư viện, phản ánh tình trạng của thư viện.

**Thủ thư – User:** Bộ phận có chức năng tương tác trực tiếp với khách hàng, tư vấn và hướng dẫn khách hàng sử dụng các chức năng nghiệp vụ của thư viện. Ngoài ra, khách hàng có thể thông qua bộ phận này để thực hiện một số chức năng như đăng ký thẻ đọc giả để thuận lợi cho việc mượn sách, trả sách,... Là bộ phận đón khách hàng và hướng dẫn khách hàng làm thủ tục khi tới mượn sách, trả sách. Ngoài ra bộ phận còn có chức năng thanh toán chi phí cho độc giả các khoản liên qua. Bộ phận lập các phiếu mượn, phiếu trả đồng thời lưu lại thông tin của khách vào hệ thống quản lý và chịu trách nhiệm quản lý các thông tin của các loại phiếu mà khách hàng sử dụng. Bộ phận chịu trách nhiệm báo cáo tình hình phát triển của thư viện cho quản lý.

### **3.2. Quy trình nghiệp vụ:**

#### **3.2.1 Đăng ký thẻ độc giả:**

Bạn đọc muốn mượn sách thì phải đăng ký thẻ độc giả. Bạn đọc cần cung cấp thông tin cá nhân gồm họ và tên, giới tính, địa chỉ và số điện thoại của mình. Thủ thư sẽ cung cấp cho bạn đọc một thẻ độc giả, trên đó có mã số thẻ chính là mã số bạn đọc để phân biệt các bạn đọc khác. Thẻ độc giả là bắt buộc phải có dùng để mượn sách.

Sau khi bạn đọc đăng ký thẻ độc giả thành công, bạn đọc muốn mượn sách (thư viện cho mượn tối đa 3 cuốn/1 lần mượn), bạn đọc chọn cho mình những cuốn sách cần mượn rồi gập thủ thư để tiến hành cho bước mượn sách.

#### **3.2.2 Mượn sách:**

Bạn đọc trình thẻ độc giả cho thủ thư kiểm tra thông tin và thủ thư tiến hành lập phiếu mượn.

Ngoài ra, chương trình còn hiển thị thông tin về việc mượn trả sách của bạn đọc gồm: mã sách, tên sách, số sách mượn, thông tin ngày mượn, ngày hẹn trả và thông tin của người mượn..

Thông tin về thẻ độc giả của bạn đọc hợp lệ, thì thủ thư sẽ cho mượn sách. Phần mềm sẽ chạy và báo rằng bạn có được mượn sách hay không. Trường hợp bạn không được mượn sách thì phần mềm sẽ thông báo lý do.

Mỗi cuốn sách có thể được mượn tối đa 30 ngày. Nếu quá hạn, khi trả sách bạn đọc phải đóng tiền lệ phí quá hạn (5.000đ/ngày).

#### **3.2.3 Trả sách:**

Khi sách được trả, thủ thư kiểm tra mã sách trên sách. Đối chiếu đúng thông tin và tình trạng sách trả, thủ thư cập nhật thông tin vào phần quản lý phiếu trả.

Trường hợp tình trạng sách cũ, hư hỏng so với khi mượn và trường hợp làm mất sách thì bạn đọc phải đền bù theo quy định.

#### **3.2.4 Cập nhật tình trạng sách:**

Sau khi hoàn thành việc trả sách, thông tin số lượng sách và tình trạng cuốn sách sẽ được thủ thư cập nhật trên hệ thống.

Những bạn đọc khác có thể mượn sách ngay sau đó.

### 3.2.5 Quản lý thông tin độc giả và thông tin người dùng:

#### 3.2.5.1. Quản lý thông tin độc giả:

Thông tin của độc giả sẽ được lưu lại khi khách hàng đăng ký thẻ độc giả để thực hiện cho việc mượn sách.

Thông tin của người dùng sẽ được quản lý – admin có quyền truy cập, thêm, chỉnh sửa và quản lý trên hệ thống.

### **3.3: Phân tích yêu cầu**

#### 3.3.1. Mục đích:

Mục đích của đề tài là khảo sát phân tích thiết kế chương trình quản lý sách của thư viện để hiểu biết và phục vụ cho việc mượn trả sách, quản lý độc giả và thống kê báo cáo tại thư viện sao cho có hiệu quả nhất.

#### 3.3.2. Yêu cầu:

Cần phải hiểu biết rõ và nắm được các nghiệp vụ của việc quản lý thư viện từ đó đi đến khảo sát, phân tích hệ thống quản lý thư viện theo đúng yêu cầu, cuối cùng là phải thiết kế được chương trình với các chức năng chỉ rõ ở bước phân tích hệ thống.

##### 3.3.2.1 Yêu cầu chức năng:

Website Quản lý thư viện cần đáp ứng các yêu cầu chức năng cơ bản sau:

\* Yêu cầu lưu trữ:

Hệ thống đáp ứng các yêu cầu lưu trữ thông tin cần thiết về người dùng, sách, độc giả, phiếu mượn và phiếu trả...của thư viện.

\*Thông tin về người dùng:

Tài khoản

Mật khẩu

Họ và tên

Số điện thoại

Số CMND/căn cước

Phân quyền

\* Thông tin về sách:

Mã sách

Tên sách

Tác giả

Nhà xuất bản

Thể loại

Số lượng

Giá

Hình ảnh

Giới thiệu sách

\* Thông tin về độc giả:

Mã độc giả

Tên độc giả

Số điện thoại

Địa chỉ

Giới tính

Số lần mất sách

\* Thông tin về phiếu mượn:

Mã phiếu

Mã độc giả

Tên độc giả

Ngày mượn

Ngày hẹn trả

Sách mượn

Thủ thư

\* Thông tin về phiếu trả:



Mã phiếu

Mã độc giả

Mã sách

Ngày hẹn trả

Ngày trả

Tình trạng sách mượn

Tình trạng sách trả

Thủ thư nhận sách

Ghi chú

Tiền phạt

\* Yêu cầu tra cứu:

Tra cứu thông tin độc giả

Tra cứu thông tin sách

Tra cứu thông tin mượn sách

Yêu cầu nghiệp vụ:

Thêm độc giả

Thêm sách mới

Tra cứu sách

Lập phiếu mượn

Tra cứu phiếu mượn

Thêm người dùng

Nhận trả sách

Thống kê

### 3.3.2.1 Yêu cầu phi chức năng:

1 Tốc độ

Chương trình khi hoạt động không mất nhiều thời gian, tốc độ tra cứu thông tin phải nhanh.

## 2 Khối lượng lưu trữ

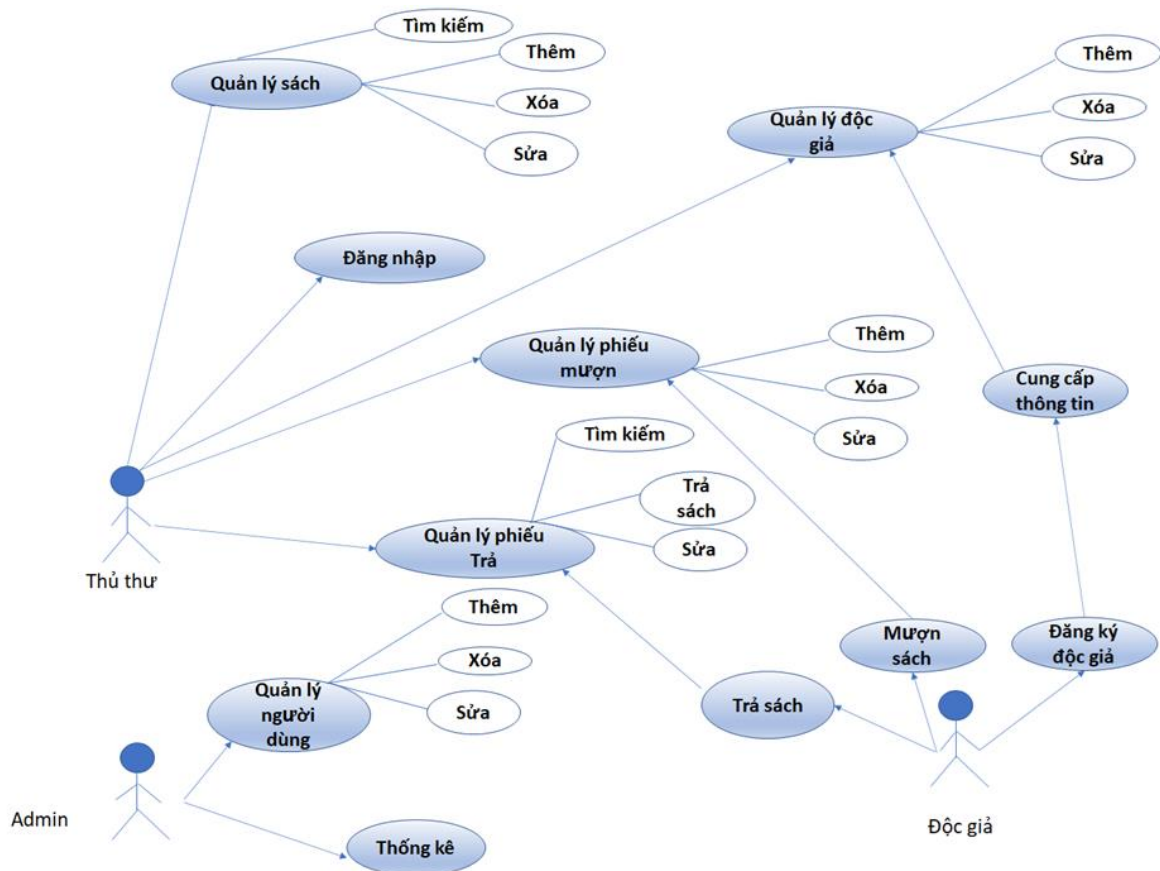
Phần mềm phải gọn nhẹ, không chiếm quá nhiều dung lượng ổ cứng.

## 3 An toàn, bảo mật

Dễ sử dụng, đảm bảo chất lượng, và chống người khác xâm nhập ngoài chủ sở hữu.

### 3.4: Mô hình usecase

#### 3.4.1. Sơ đồ usecase tổng quát:



Hình 3.2: Sơ đồ usecase tổng quát

#### 3.4.2. Mô tả usecase:

Bảng 3.1: Mô tả usecase tổng quát

STT	Tên usecase	Ý nghĩa
1.	Đăng nhập	Hỗ trợ đăng nhập vào hệ thống.( dành cho nhân viên và admin)
2.	Quản lý người dùng	Giúp người quản trị quản lý được các( thủ thư) nhân viên. ( dành cho admin)
3.	Quản lý độc giả	Quản lý các thông tin liên quan đến độc giả
4.	Quản lý sách	Quản lý các thông tin liên quan đến sách.

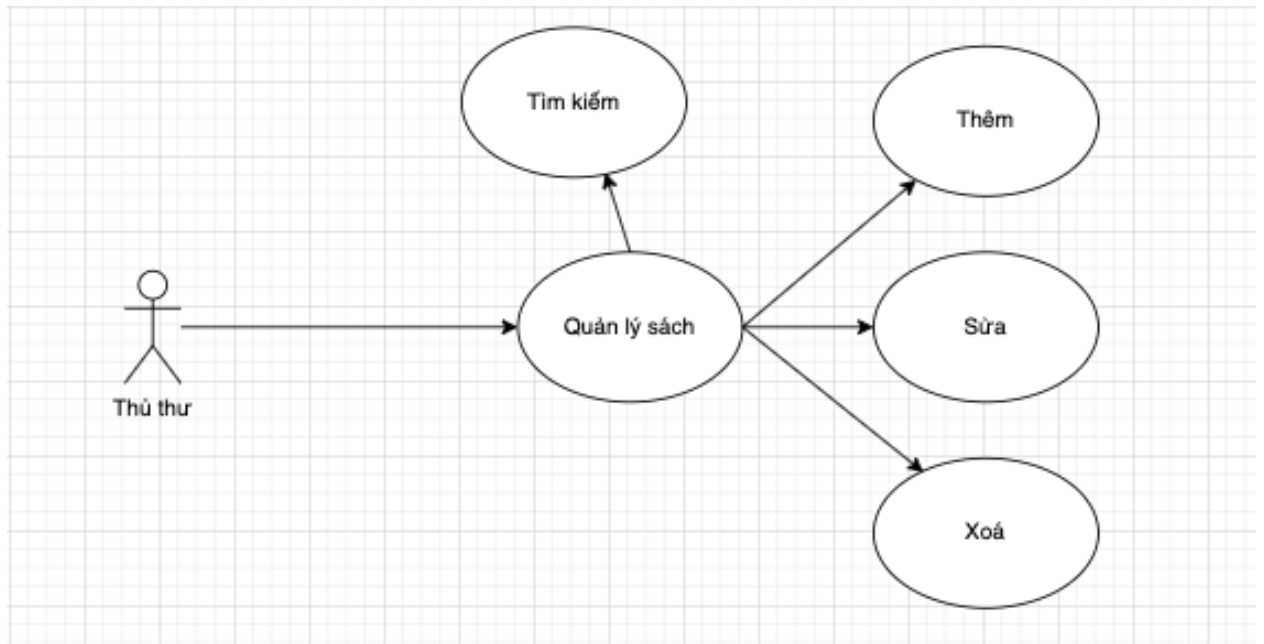
5.	Quản lý phiếu mượn	Quản lý các thông tin liên quan đến việc mượn sách.
6.	Quản lý phiếu trả	Quản lý các thông tin liên quan đến việc trả sách ( mất sách, hư hỏng, đúng hạn, trễ hạn,...).
7.	Thống kê	Thống kê những thông tin ( sách, phiếu mượn , phiếu trả, độc giả,...). ( chỉ dành cho admin)
8.	Mượn sách	Độc giả tìm kiếm sách muốn mượn và liên hệ thủ thư để mượn sách.
9.	Trả sách	Đến hạn trả sách, độc giả đến và trả sách.( Nếu độc giả làm mất sách, hư sách, trễ hạn sẽ bị phạt tiền theo qui định)
10.	Đăng ký độc giả	Người đọc muốn mượn sách phải cung cấp cho thông tin cho thủ thư để trở thành độc giả.

### 3.4.3. Danh sách tác nhân nghiệp vụ hệ thống:

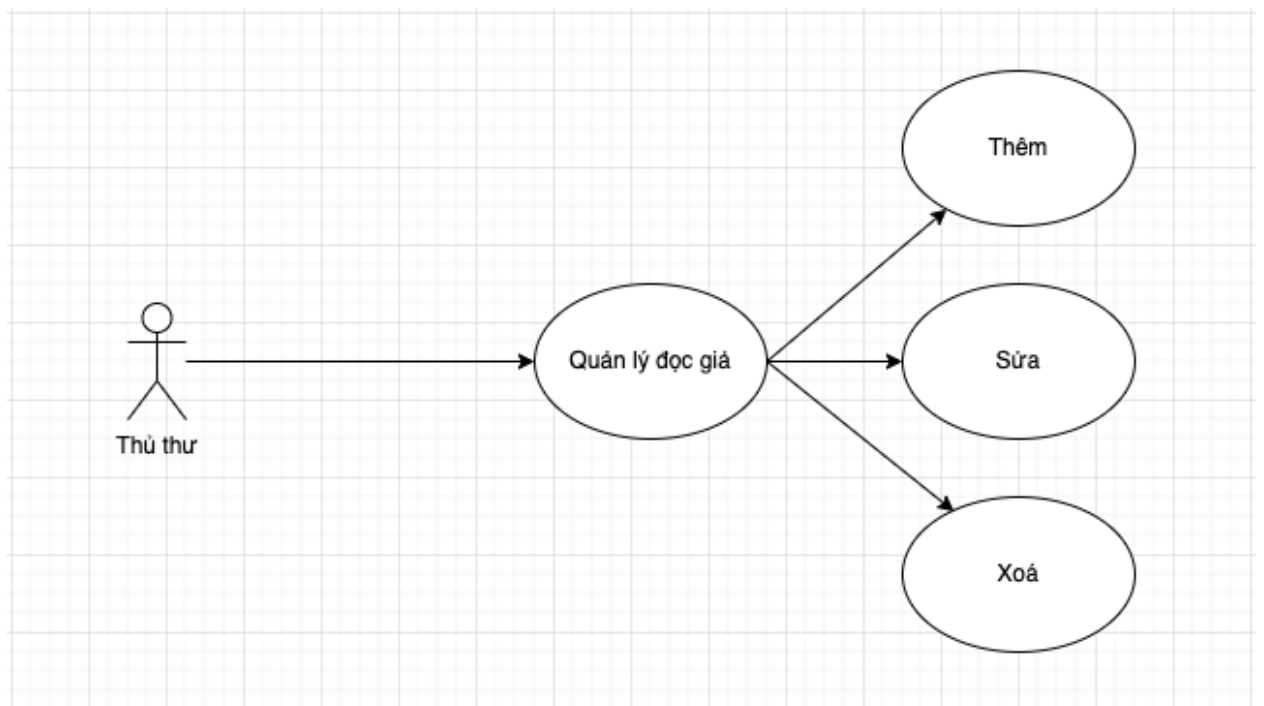
*Bảng 3.2: Danh sách tác nhân nghiệp vụ*

STT	Tác nhân hệ thống	Ý nghĩa
1	Độc giả	Là người đến thư viện mượn sách.
2	Thủ thư	Là người quản lý (sách, cho mượn sách, nhận sách, độc giả).
3	Admin	Người quản lý hệ thống.

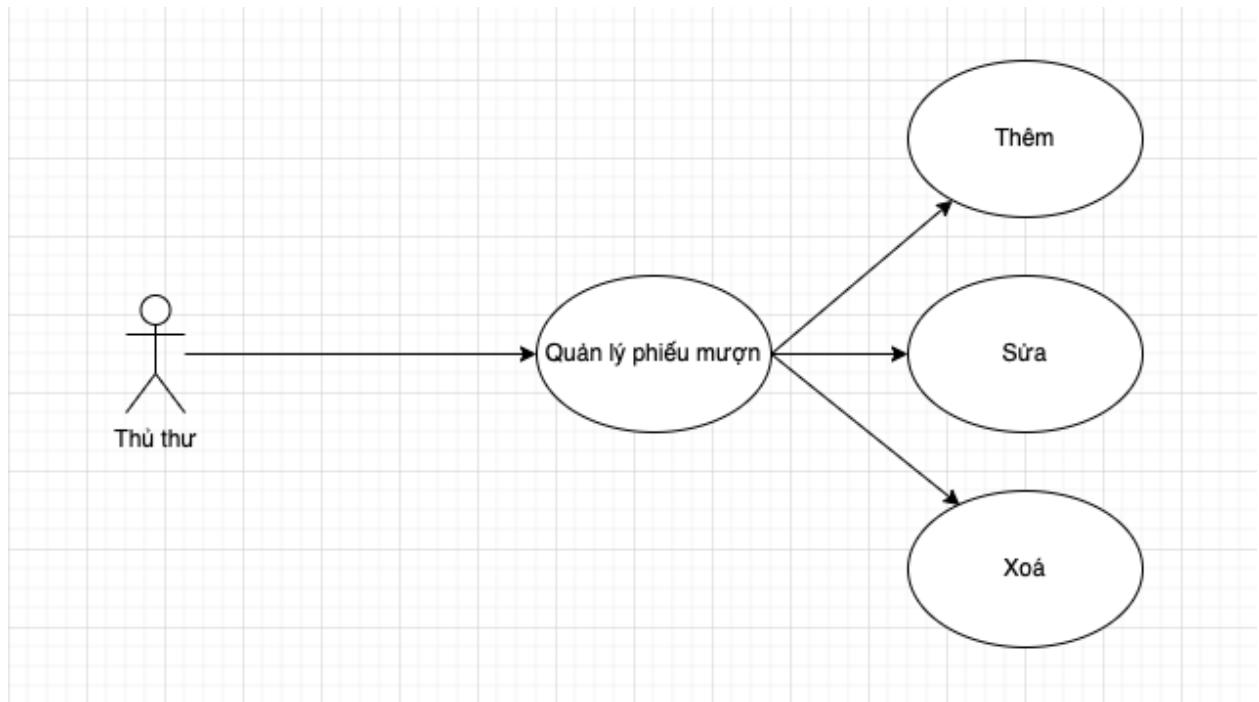
## 3.4.4. Sơ đồ usecase quản lý sách:

*Hình 3.3: usecase quản lý sách*

## 3.4.5. Sơ đồ usecase quản lý độc giả:

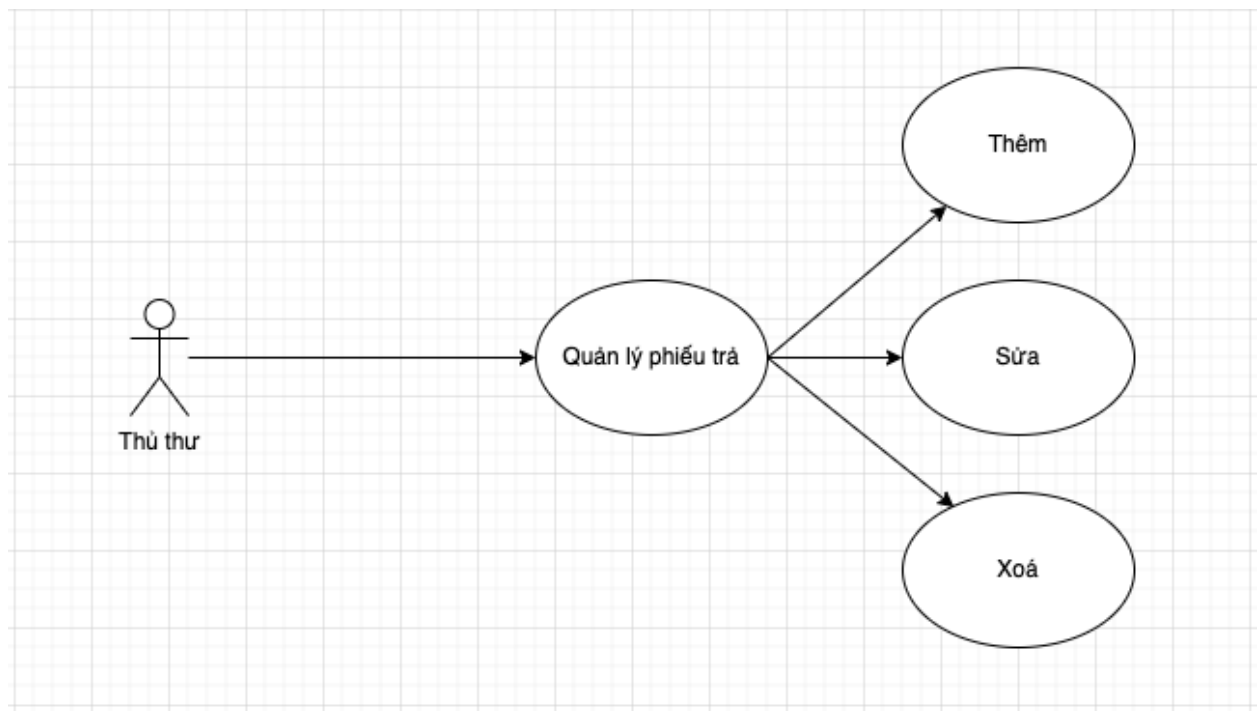
*Hình 2: usecase quản lý độc giả*

#### 3.4.6. Sơ đồ usecase quản lý phiếu mượn:



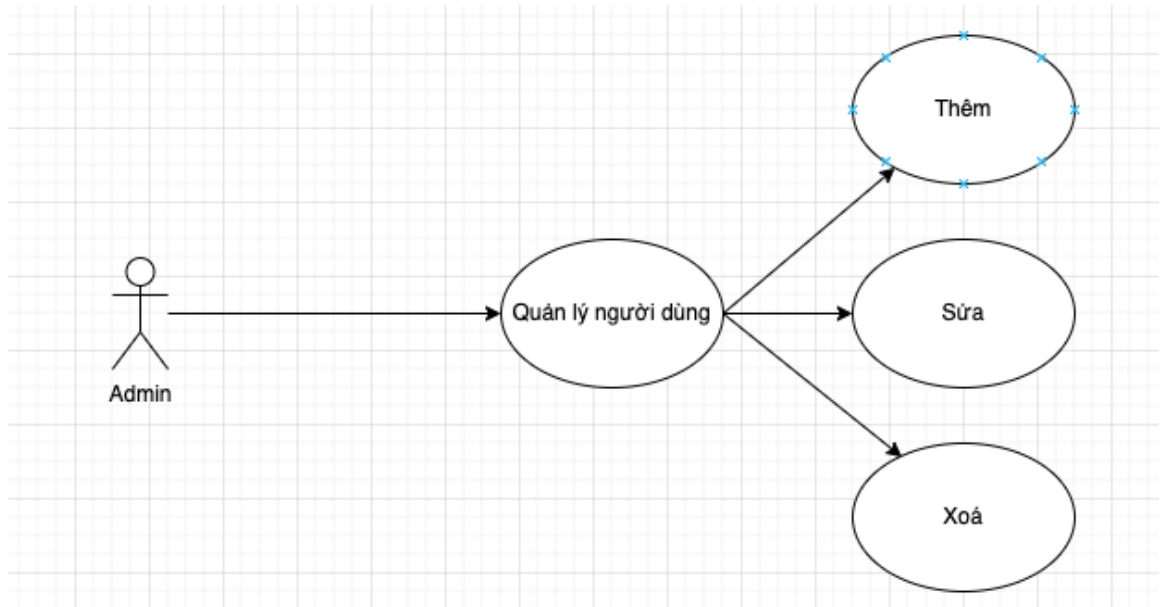
Hình3.4: usecase quản lý phiếu mượn

#### 3.4.7. Sơ đồ usecase quản lý phiếu trả:



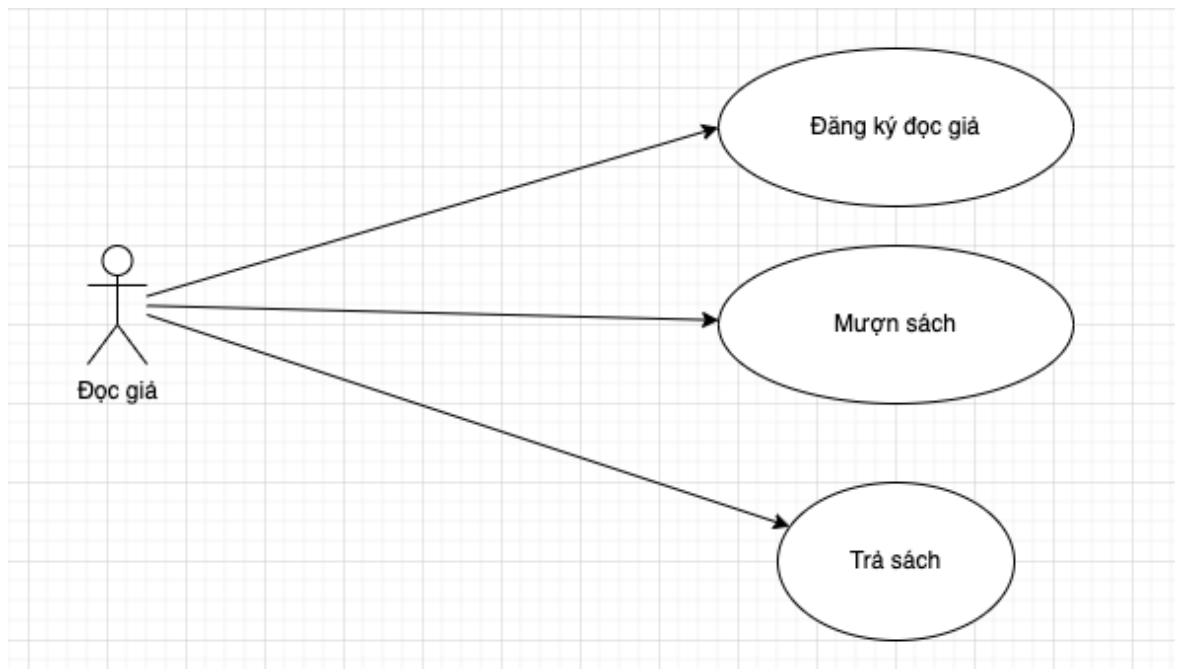
Hình 3.5: usecase quản lý phiếu trả

#### 3.4.8. Sơ đồ usecase quản lý người dùng:



Hình 3.6: usecase quản lý người dùng

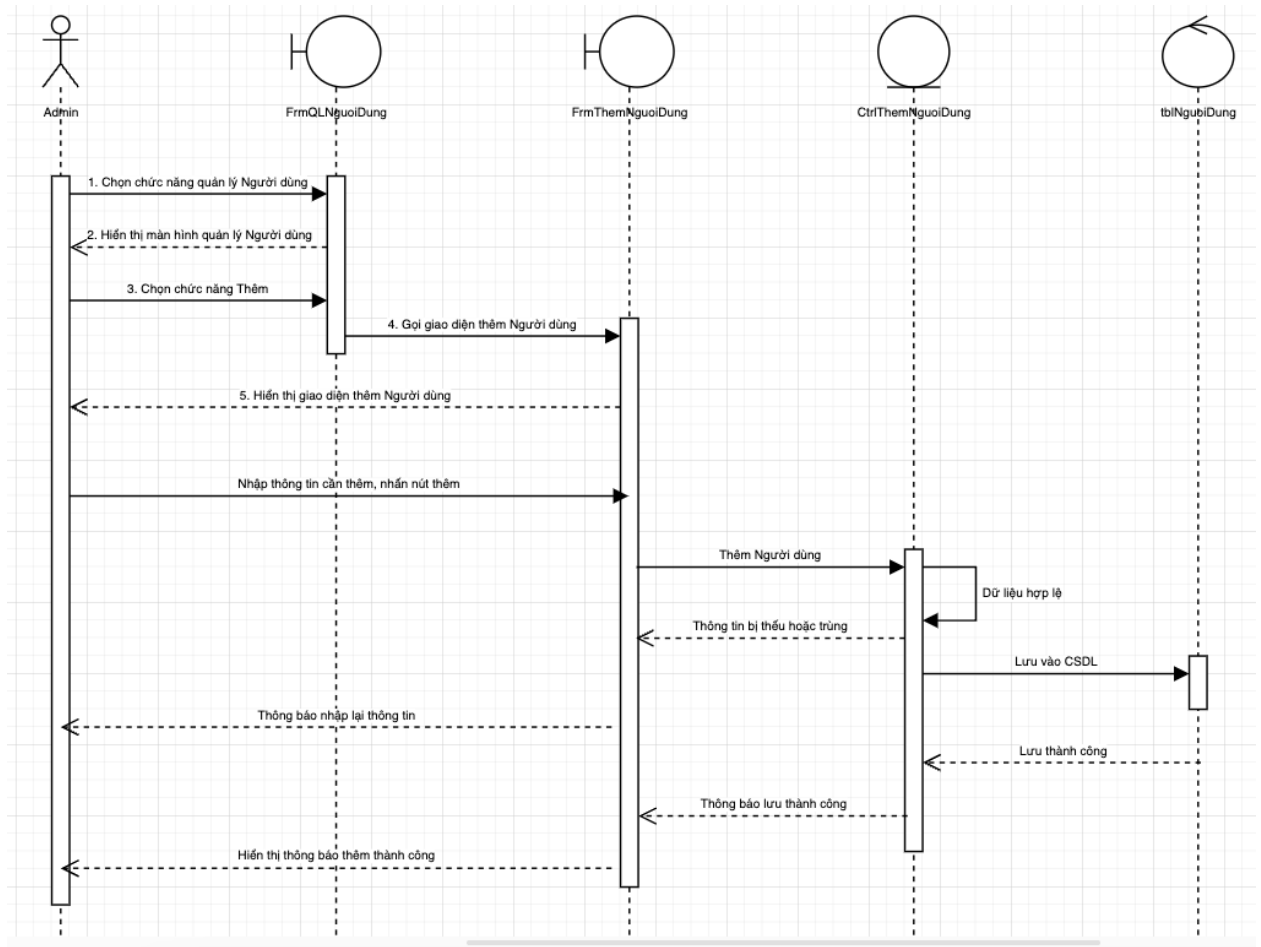
#### 3.4.9. Sơ đồ usecase quản lý độc giả:



Hình 3.7: usecase quản lý độc giả

### 3.5. Sơ đồ tuần tự:

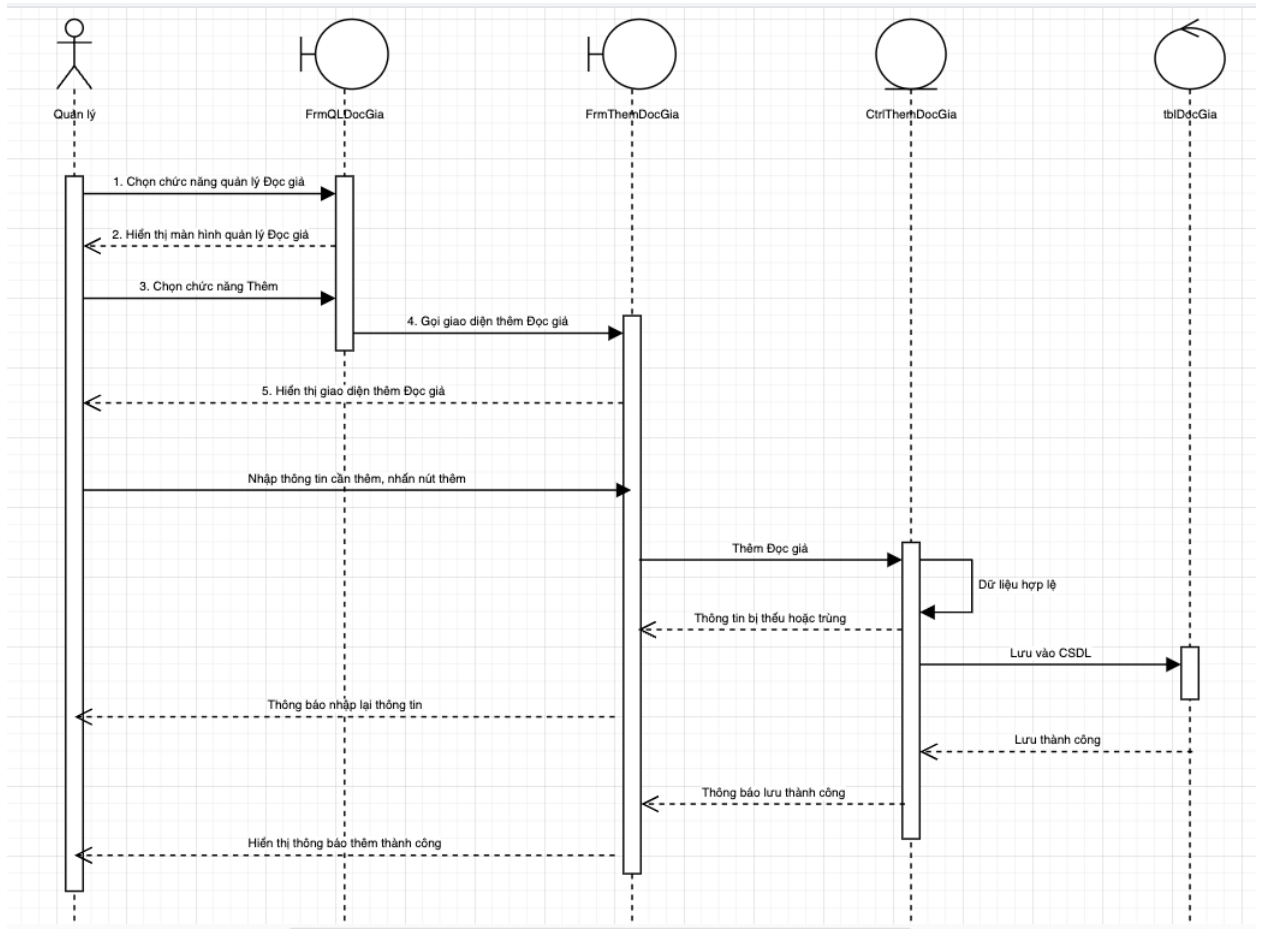
#### 3.5.1. Sơ đồ tuần tự quản lý người dùng



Hình 3.8: Sơ đồ tuần tự quản lý người dùng

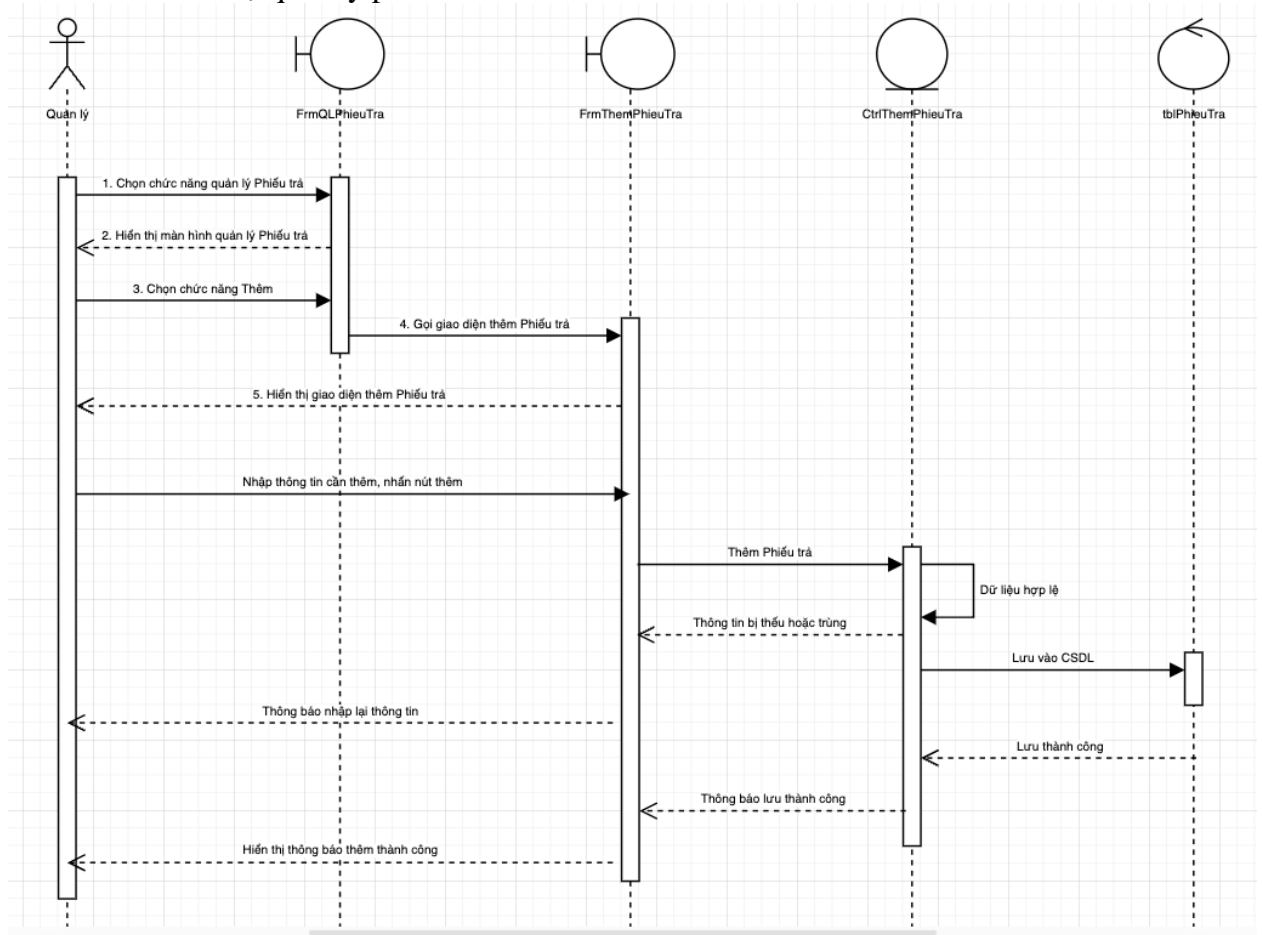


### 3.5.2. Sơ đồ tuần tự quản lý độc giả



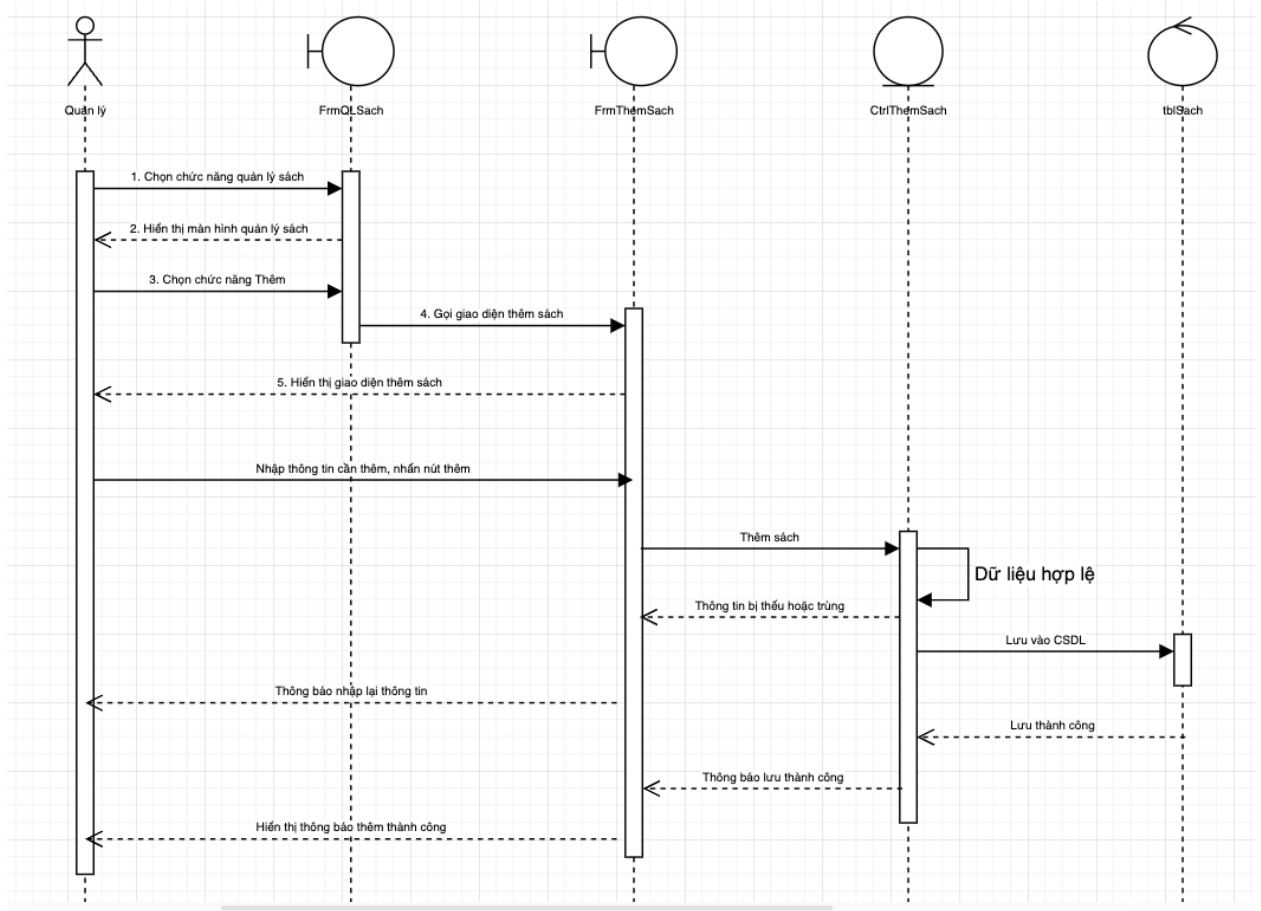
Hình 3.9: Sơ đồ tuần tự quản lý độc giả

## 3.5.3. Sơ đồ tuần tự quản lý phiếu trả



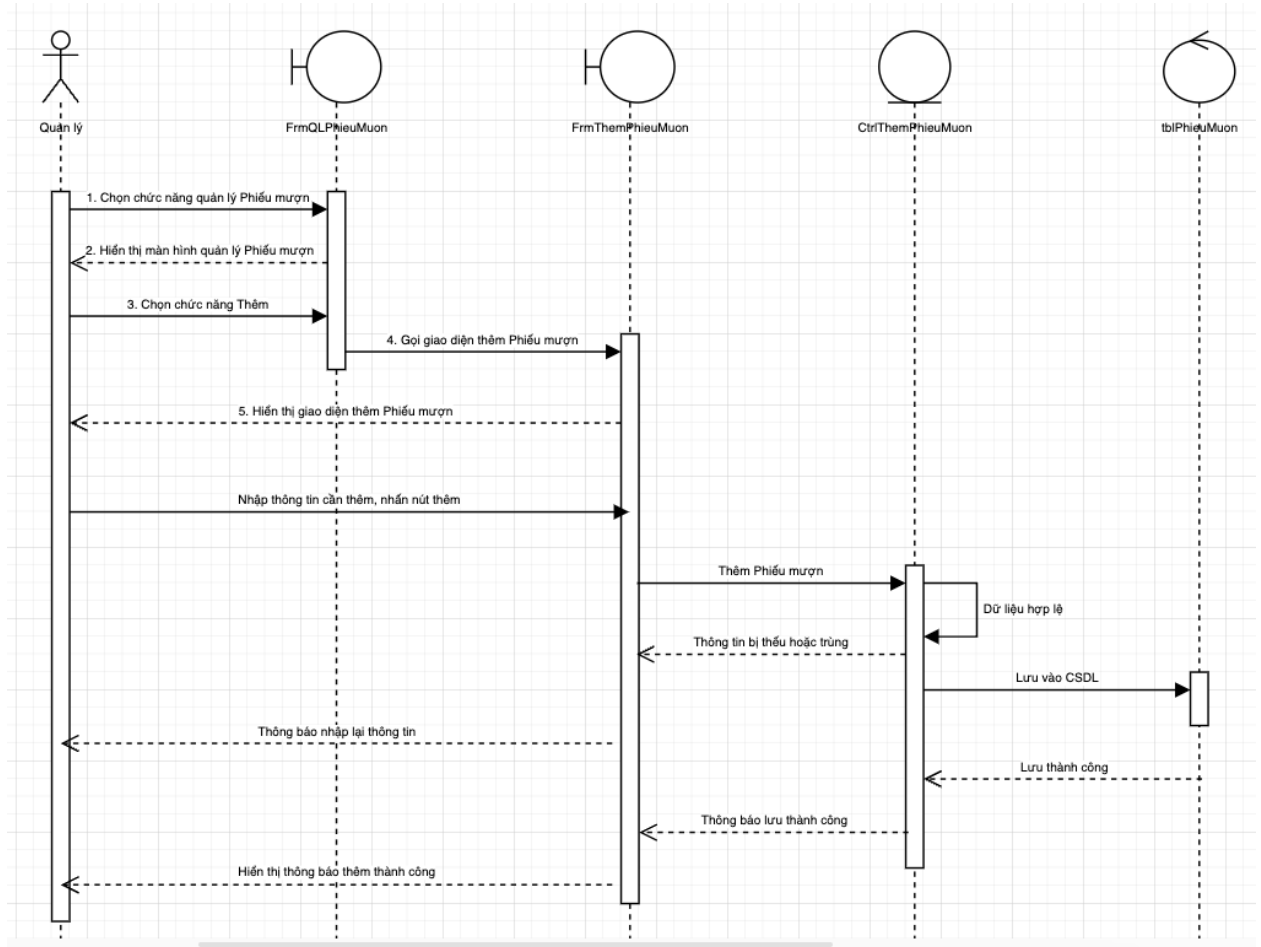
Hình 3.10: Sơ đồ tuần tự quản lý phiếu trả

### 3.5.4. Sơ đồ tuần tự quản lý sách



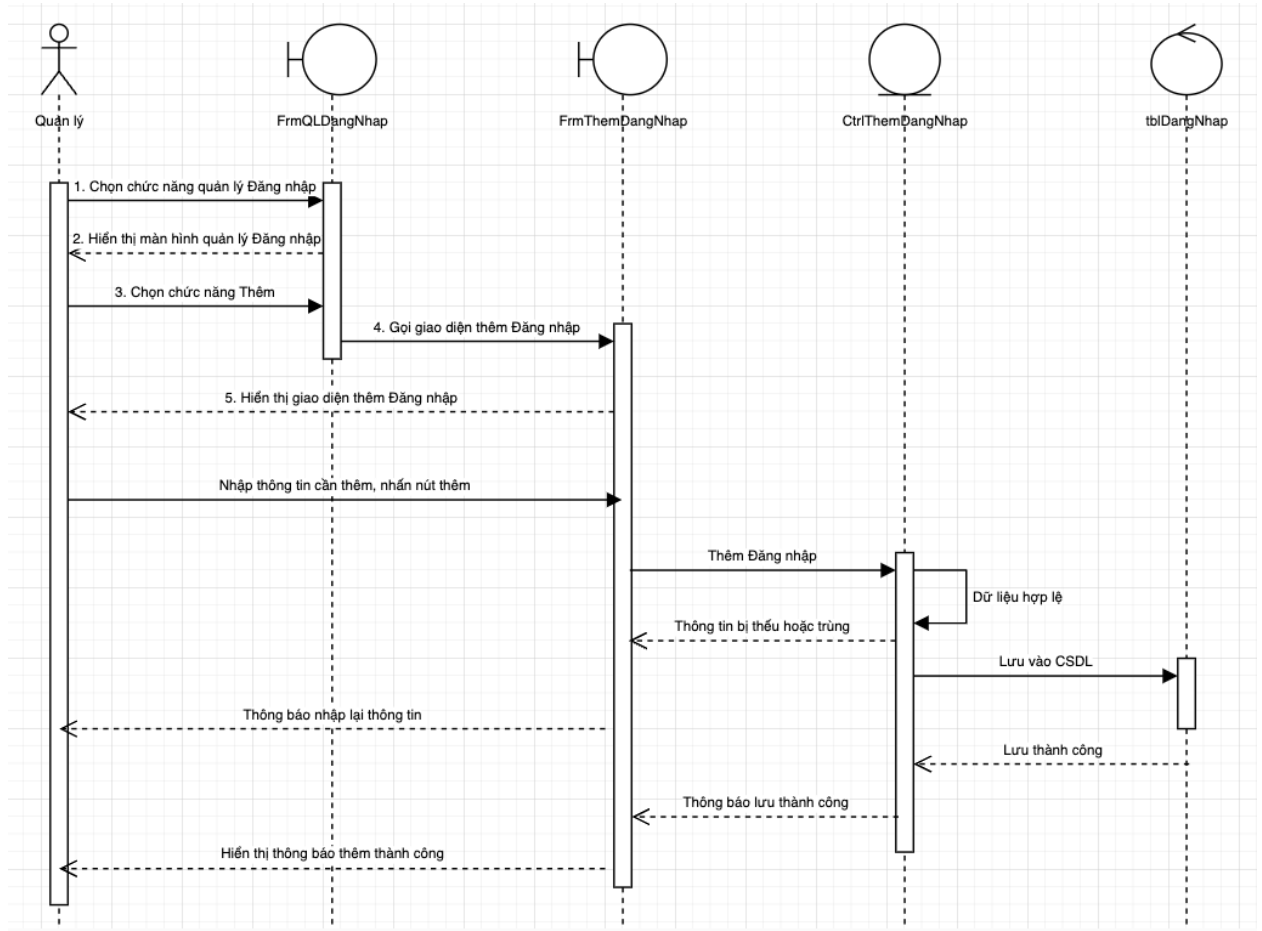
Hình 3.11: Sơ đồ tuần tự quản lý sách

### 3.5.4. Sơ đồ tuần tự quản lý phiếu mượn



Hình 3.12: Sơ đồ tuần tự quản lý phiếu mượn

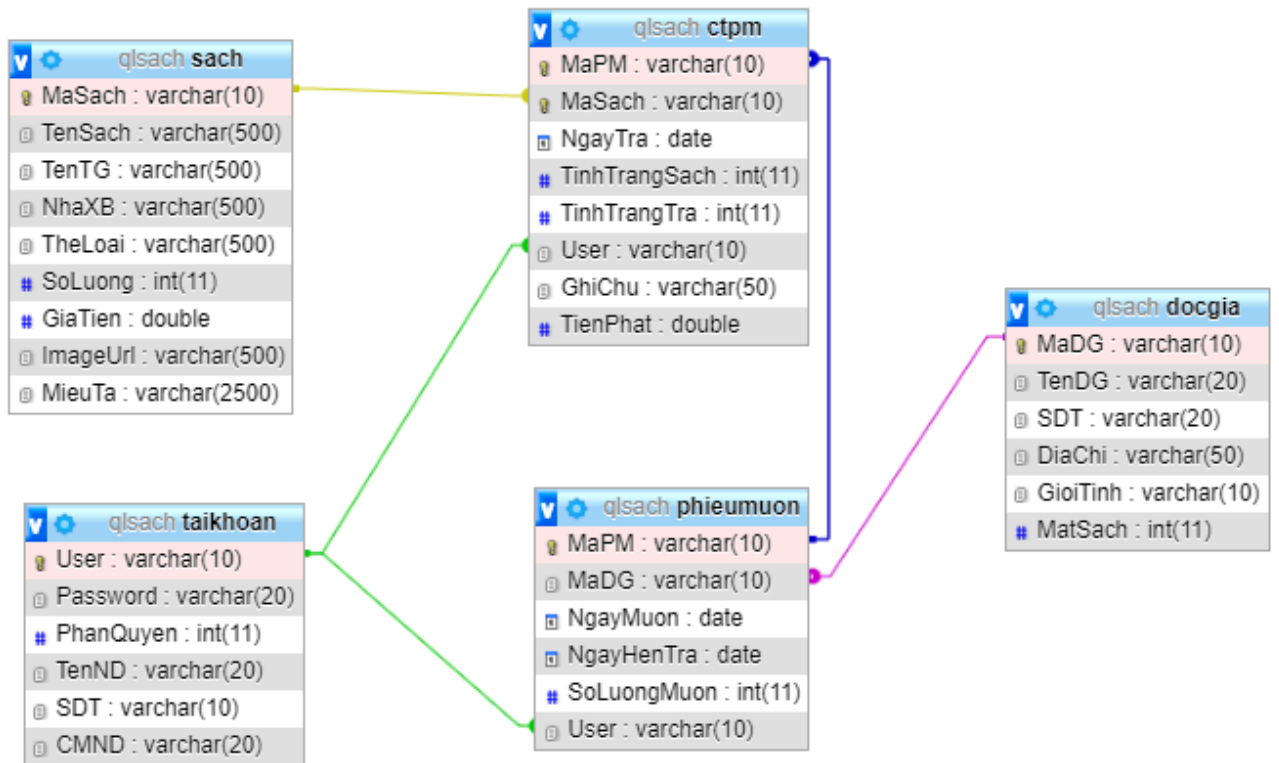
### 3.5.4. Sơ đồ tuần tự đăng nhập



Hình 3.13: Sơ đồ tuần tự đăng nhập

### 3.6: Thiết kế cơ sở dữ liệu

#### 3.6.1. Mô hình thiết kế dữ liệu:



Hình 3.14: Mô hình cơ sở dữ liệu

#### 3.6.2. Mô tả bảng dữ liệu:

##### 3.6.2.1 Bảng DOCGIA

Chức thông tin về các quy định của độc giả gồm : mã độc giả, tên độc giả, số điện thoại, địa chỉ, giới tính, số lần mất sách.

Bảng 3.3: Mô tả bảng DOCGIA

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	MaDocGia	Text		Là khóa chính để phân biệt với các mã độc giả khác.
2	TenDocGia	Text		Tên của độc giả.
3	SoDienThoai	Text		Số điện thoại của độc giả không quá 10 số.

4	DiaChi	Text		Địa chỉ của độc giả.
5	GioiTinh	Text		Giới tính của độc giả.
6	MatSach	Int		Là số lần mất sách. Mất quá 3 lần thì sẽ không được mượn sách nữa.

### 3.5.2.2 Bảng PHIEUMUON

Chứa các thông tin quy định mượn sách gồm:

*Bảng 3.4: Mô tả bảng PHIEUMUON*

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn Giải
1	MaPhieuMuon	Text		Là khóa chính để phân biệt với các mã mượn khác.
2	MaDocGia	Text		Mã của độc giả (là khóa ngoại).
3	NgayMuon	Date		Ngày độc giả mượn.
4	NgayHenTra	Date		Ngày độc giả hẹn trả.
5	SoLuong	Number		Số lượng sách độc giả mượn.
6	User	Text		Thủ thư cho mượn sách.

### 3.5.2.3 Bảng SACH

Chứa các thông tin về cuốn sách gồm: mã cuốn sách, tên cuốn sách, tên tác giả, tên nhà xuất bản, thể loại, số lượng tồn của sách, giá tiền.

*Bảng 3.5: Mô tả bảng SACH*

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn Giải
1	MaSach	Text		Là khóa chính để phân biệt với các mã cuốn sách khác.

2	TenSach	Text		Tên cuốn sách.
3	TenTG	Text		Tên tác giả.
4	NhaXB	Text		Tên nhà xuất bản.
5	TheLoai	Text		Tên thể loại.
6	SoLuong	Int		Số lượng tồn.
7	GiaTien	Double		Giá tiền của sách
8	Image	Text		Đường dẫn hình ảnh cuốn sách
9	MieuTa	Text		Miêu tả nội dung sách

#### 3.5.2.4. Bảng CTPM

Chức các thông tin về chi tiết phiếu mượn gồm : Mã phiếu mượn, mã sách, ngày trả, tình trạng lúc mượn sách, tình trạng trả sách, thủ thư nhận sách, ghi chú.

*Bảng 3.6: Mô tả bảng CHITIETPHIEUMUON*

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	MaPM	Text		Là khóa chính để phân biệt với các mã cuốn sách khác.
2	MaSach	Text		Là khóa chính để phân biệt với các mã cuốn sách khác.
3	NgayTra	Date		Ngày trả của độc giả.
4	TinhTrangSach	Int		Tình trạng sách lúc mượn.
5	TinhTrangTra	Int		Tình trạng sách lúc trả.
6	User	Text		Thủ thư nhận sách.
7	GhiChu	Text		Ghi chú vào lúc nhận sách (hư, trễ hạn...)



8	TienPhat	Double		Số tiền mà độc giả phải trả khi làm hư sách, mất sách hoặc trả trễ so với hạn.
---	----------	--------	--	--

### 3.5.2.5 Bảng TAIKHOAN

Chức các thông tin về chi tiết phiếu mượn gồm: tài khoản đăng nhập, mật khẩu, vai trò, tên người dùng, số điện thoại, chứng minh nhân dân.

*Bảng 3.7: Mô tả bảng TAIKHOAN*

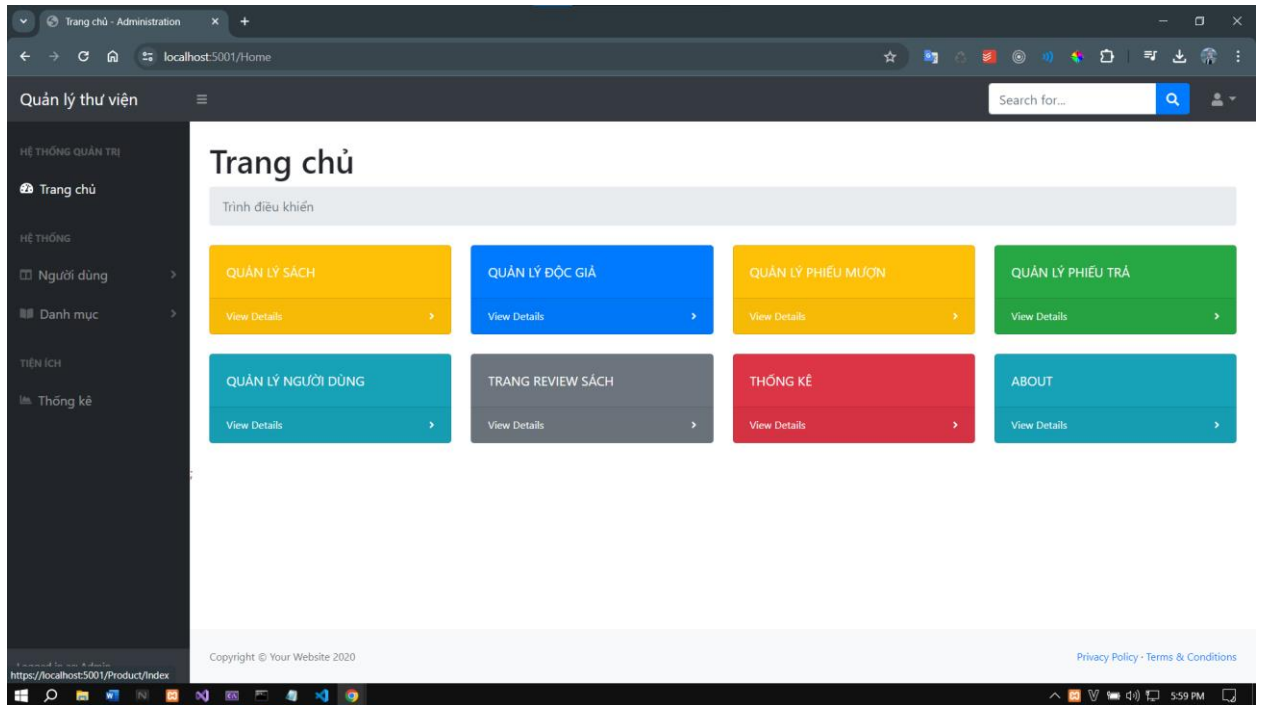
STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	User	Text		Là khóa chính để phân biệt với các người dùng khác.
2	Password	Text		Mật khẩu để đăng nhập vào phần mềm.
3	PhanQuyen	Date		Vai trò của người dùng.
4	TenND	Int		Tên người dùng.
5	SDT	Int		Số điện thoại của người dùng.
6	CMND	Text		CMND của người dùng.

## 3.7. Tiểu kết

Chương này đã cung cấp cái nhìn tổng quan về hiện trạng tổ chức và quy trình nghiệp vụ của hệ thống quản lý thư viện. Qua việc phân tích yêu cầu và thiết kế cơ sở dữ liệu, chúng ta đã xác định được các chức năng cần thiết và mô hình hóa hệ thống qua các sơ đồ usecase và mô tả bảng dữ liệu. Những kết quả này là nền tảng để triển khai hệ thống trong chương tiếp theo.

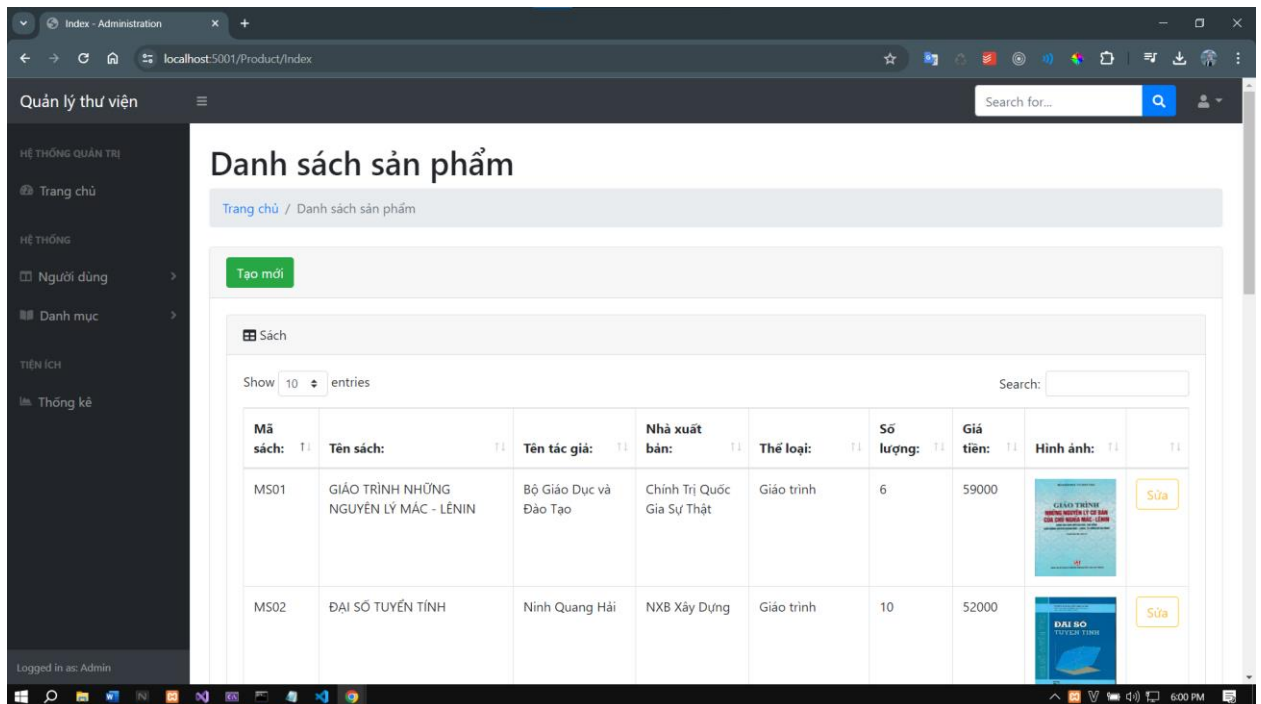
## CHƯƠNG IV: CÁC KẾT QUẢ ĐẠT ĐƯỢC

### 4.1. Giao diện trang chủ



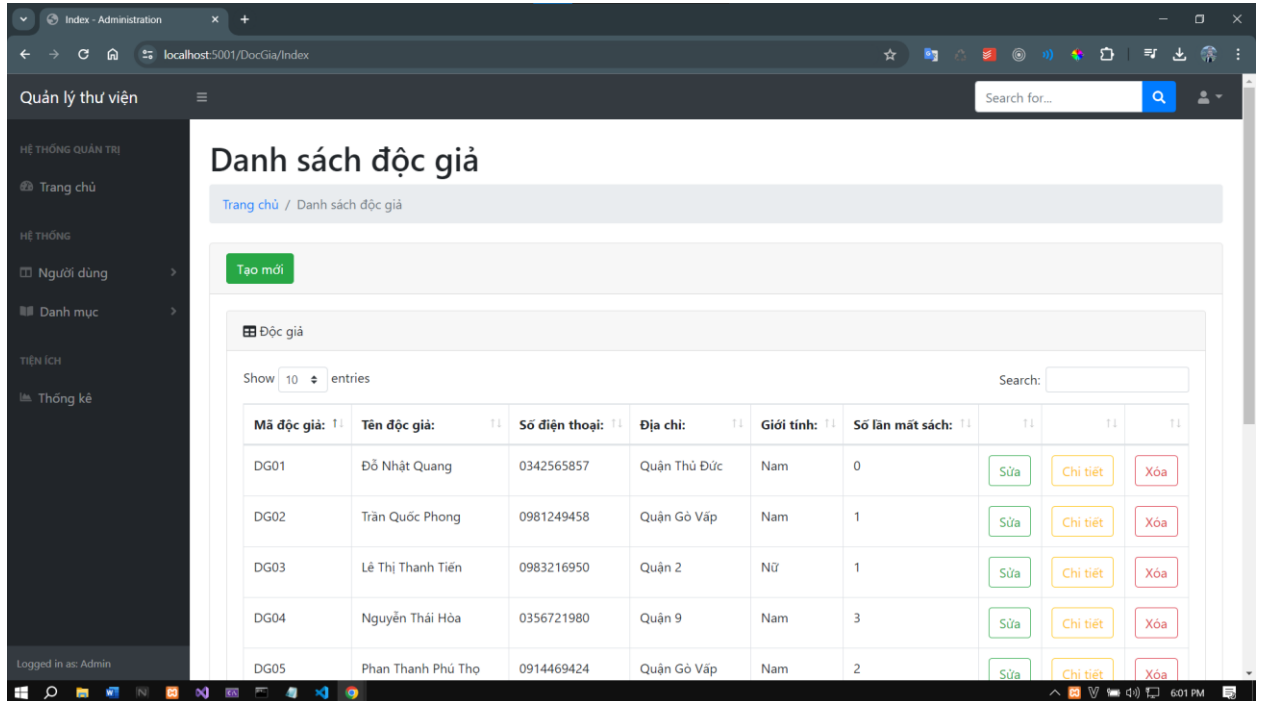
Hình 4.1: Giao diện trang chủ

### 4.2. Giao diện trang quản lý sách



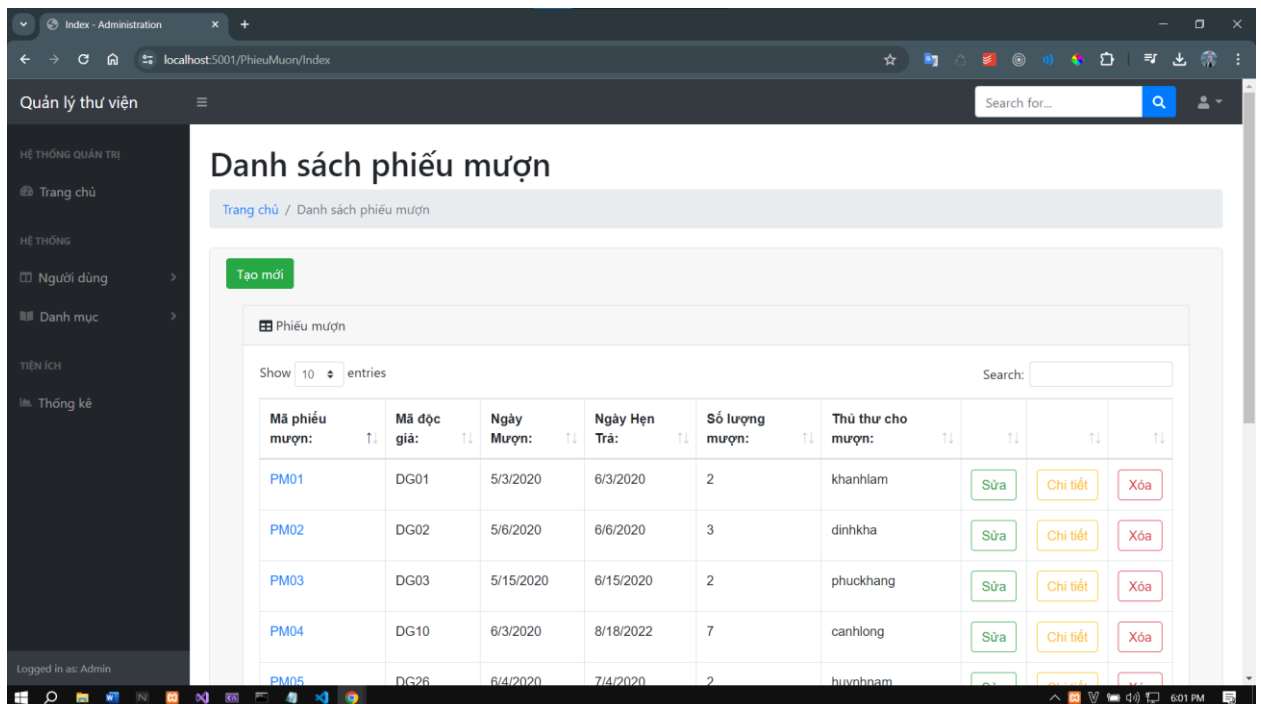
Hình 4.2: Giao diện trang quản lý sách

### 4.3. Giao diện trang quản lý độc giả



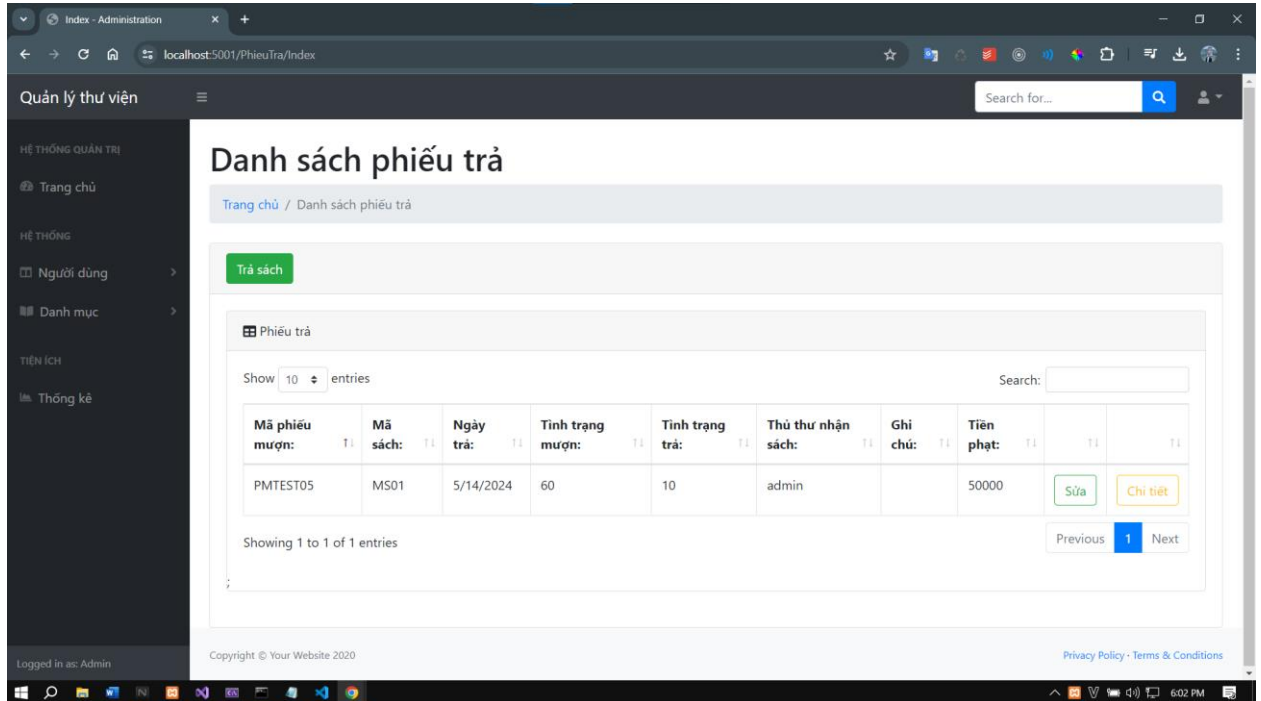
Hình 4.3: Giao diện trang quản lý độc giả

### 4.4. Giao diện trang quản lý phiếu mượn



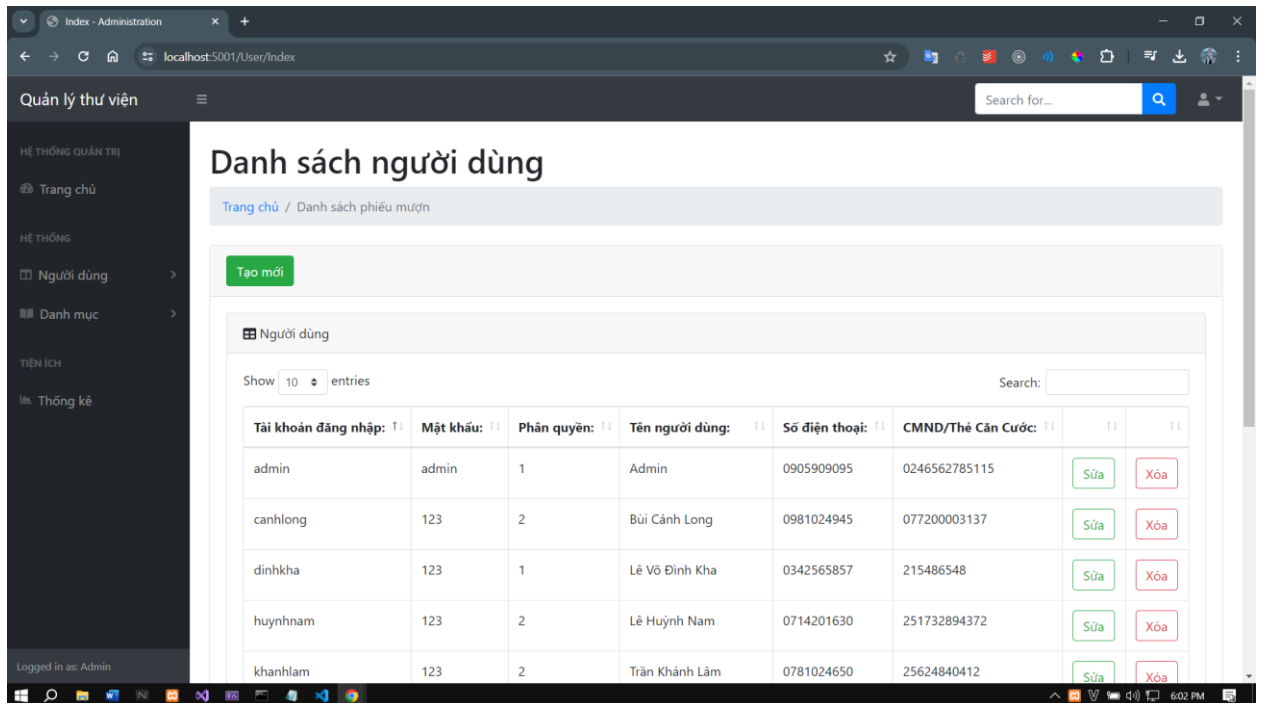
Hình 4.4: Giao diện trang quản lý phiếu mượn

## 4.5. Giao diện trang quản lý phiếu trả



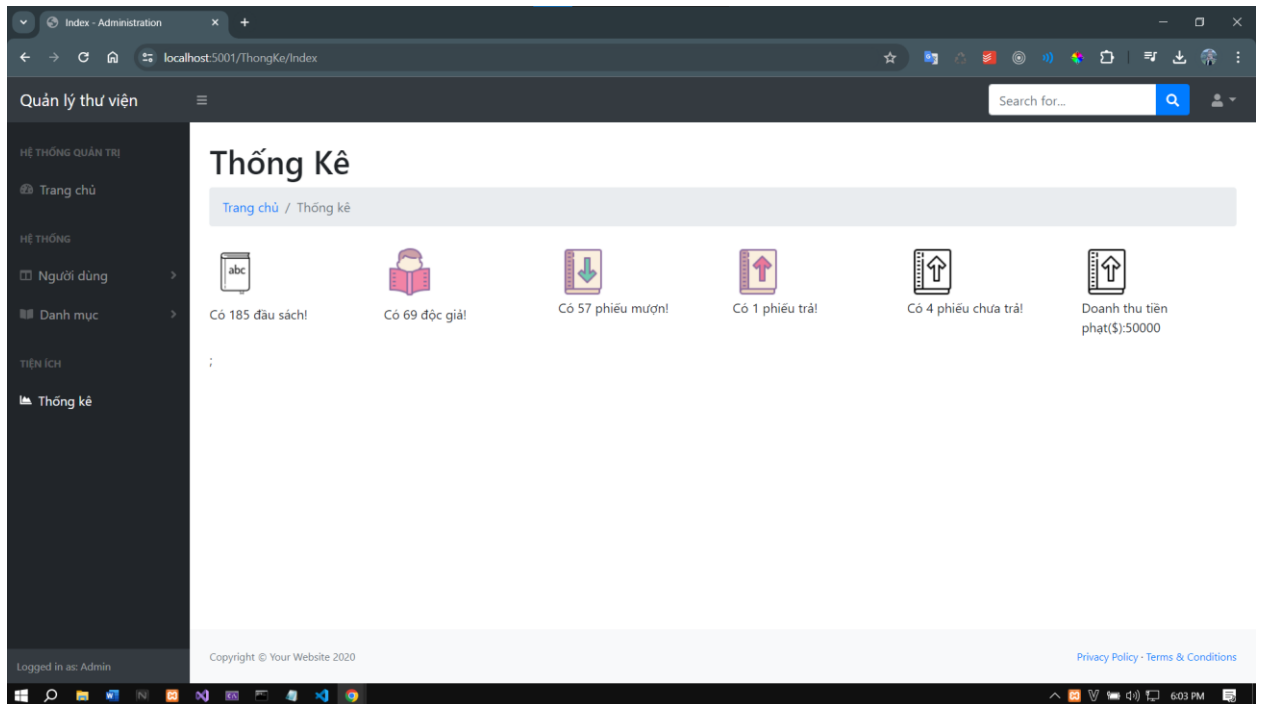
Hình 4.5: Giao diện trang quản lý phiếu trả

## 4.6. Giao diện trang quản lý người dùng:



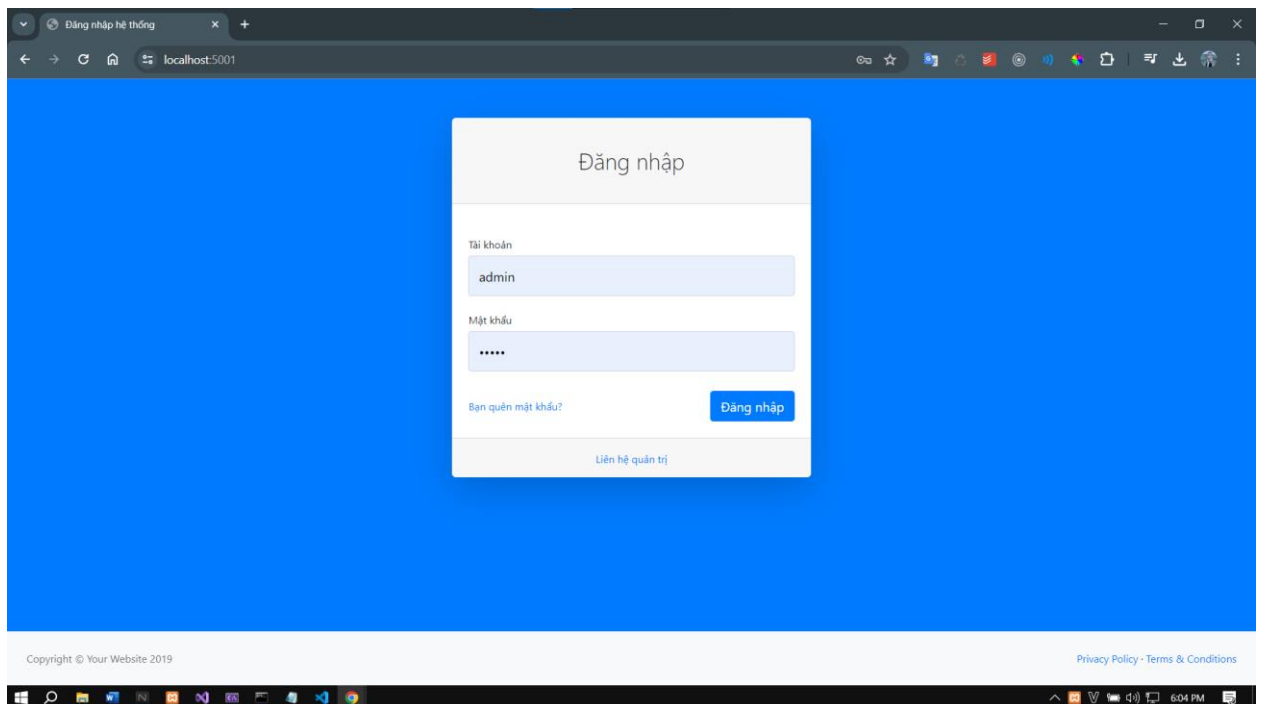
Hình 4.6: Giao diện trang quản lý người dùng

## 4.7. Giao diện trang thống kê



Hình 4.7: Giao diện trang thống kê

## 4.8. Giao diện trang đăng nhập



Hình 4.8: Giao diện trang đăng nhập

## KẾT LUẬN

Trong quá trình thực hiện đề tài "Xây dựng ứng dụng website quản lý thư viện cho trường Đại học Đại Nam," em đã đạt được nhiều thành tựu quan trọng trong việc ứng dụng các công nghệ tiên tiến vào quản lý thư viện. Với sự kết hợp của các công nghệ như LINQ, NET CORE MVC và MySQL, hệ thống đã được thiết kế để đảm bảo hiệu quả và độ tin cậy cao trong việc quản lý các quy trình mượn và trả sách, đăng ký độc giả, và thống kê dữ liệu.

Ứng dụng này không chỉ đơn giản hóa quy trình quản lý thư viện mà còn nâng cao trải nghiệm người dùng. Người dùng có thể dễ dàng tìm kiếm và mượn sách, quản lý thông tin cá nhân và theo dõi lịch sử mượn sách của mình. Hệ thống cũng hỗ trợ các chức năng thống kê, giúp ban quản lý thư viện nắm bắt được tình hình sử dụng sách và đưa ra các quyết định kịp thời để cải thiện dịch vụ.

Trong tương lai, hệ thống quản lý thư viện này có tiềm năng phát triển và mở rộng để đáp ứng nhu cầu ngày càng cao của người dùng. Một số hướng phát triển có thể kể đến như:

1. Tích Hợp Công Nghệ RFID: Việc tích hợp công nghệ RFID sẽ giúp tự động hóa quy trình mượn và trả sách, giảm thiểu sai sót và tiết kiệm thời gian cho người dùng lẫn nhân viên thư viện. Công nghệ này cũng giúp theo dõi chính xác vị trí của từng cuốn sách trong thư viện.

2. Phát Triển Ứng Dụng Di Động: Với sự phổ biến của điện thoại thông minh, việc phát triển ứng dụng di động cho hệ thống quản lý thư viện sẽ mang lại tiện ích lớn cho người dùng.

3. Hệ Thống Gợi Ý Sách: Áp dụng các thuật toán học máy để phân tích sở thích và lịch sử mượn sách của người dùng, từ đó đề xuất những cuốn sách phù hợp.

Những hướng phát triển này sẽ không chỉ giúp hệ thống quản lý thư viện đáp ứng tốt hơn nhu cầu hiện tại mà còn mở rộng phạm vi ứng dụng, nâng cao trải nghiệm người dùng và hỗ trợ tích cực cho hoạt động học tập và nghiên cứu tại trường Đại học Đại Nam. Việc không ngừng cải tiến và cập nhật công nghệ sẽ đảm bảo hệ thống luôn đáp ứng được yêu cầu ngày càng cao của người dùng và bắt kịp với sự phát triển của công nghệ.

**TÀI LIỆU THAM KHẢO**

- [1]. Trần Minh Tiến. (2019). Hệ Thống Quản Lý Thư Viện Sử Dụng MySQL. Hội thảo Quốc tế về Công nghệ Thông tin.
- [2]. Phạm Thị Lan Anh. (2017). Thiết Kế Cơ Sở Dữ Liệu và Ứng Dụng Trong Quản Lý Thư Viện. Tạp chí Khoa học và Công nghệ.
- [3]. Phạm Thị Lan Phương. (2020). Phân Tích và Thiết Kế Hệ Thống Thông Tin. Nhà xuất bản Đại học Quốc gia Hà Nội.
- [4]. Understanding LINQ: A Comprehensive Guide to Language-Integrated Query
- [5]. Exploring The Wonders Of Language Integrated Query (LINQ)
- [6]. Learn LINQ using Step-by-Step Tutorials
- [7]. Microsoft Learn. (2023). Get started with ASP.NET Core MVC. Microsoft Documentation.
- [8]. Joe Stevens. (2022). Exploring ASP.NET Core MVC: A Comprehensive Guide. Code Maze.
- [9]. Robert Half. (2021). Understanding ASP.NET Core MVC. Robert Half Blog.
- [10]. David Fowler, Scott Hanselman. (2020). ASP.NET Core in Action. Manning Publications.
- [11]. Tim Corey. (2019). ASP.NET Core MVC: From Zero to Overkill. Tim Corey Training.
- [12]. Datamation. (2021). 8 Big Advantages of Using MySQL.
- [13]. Factober. (2024). Exploring MySQL: A Comprehensive Guide to Features, Advantages, and Use Cases.
- [14]. W3Schools . (2023). MySQL advantages and disadvantages.