



# Đồ Án Tốt Nghiệp-Bùi Xuân Hùng-CNTT1-K20

Công nghệ thông tin (Đại học Hàng hải Việt Nam)

**TRƯỜNG ĐẠI HỌC HẢI PHÒNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP**  
**CHUYÊN NGÀNH CÔNG NGHỆ THÔNG TIN**  
**ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ THƯ VIỆN**

**Giảng viên hướng dẫn** : ThS. Nguyễn Thị Thúy  
**Sinh viên thực hiện** : Bùi Xuân Hùng  
**Ngày sinh** : 20/09/2001  
**Lớp** : CNTT1.K20  
**Mã sinh viên** : 193148201108

**Hải Phòng – 2023**

## LỜI CẢM ƠN

Để có thể hoàn thành được bài luận này, em xin gửi lời cảm ơn chân thành đến Ths.Nguyễn Thị Thuý đã tận tình trực tiếp hướng dẫn giúp đỡ em hoàn thành đồ án tốt nghiệp này.

Em xin chân thành cảm ơn sự giúp đỡ của toàn thể các giảng viên khoa công nghệ thông tin Trường Đại học Hải Phòng đã dạy dỗ em về cả kiến thức chuyên môn và tinh thần học tập độc lập, sáng tạo để em có được kiến thức thực hiện đồ án của mình.

Trong quá trình thực tập cũng như làm tiểu luận, do còn nhiều thiếu sót về kiến thức, kỹ năng cũng như kinh nghiệm nên không thể tránh khỏi những sai sót. Em mong nhận được sự góp ý của thầy cô để bài luận của em ngày càng hoàn thiện hơn.

*Em xin chân thành cảm ơn!*

## MỤC LỤC

LỜI CẢM ƠN.....	i
MỤC LỤC.....	ii
DANH MỤC CÁC TỪ VIẾT TẮT.....	iv
DANH MỤC CÁC BẢNG.....	v
DANH MỤC CÁC HÌNH.....	vi
LỜI NÓI ĐẦU.....	1
<b>Chương I: CƠ SỞ LÝ THUYẾT.....</b>	<b>3</b>
<b>1. CÁC CÔNG NGHỆ SỬ DỤNG.....</b>	<b>3</b>
<b>1.1. Angular Framework.....</b>	<b>3</b>
<b>1.1.1. Các phiên bản của Angular.....</b>	<b>3</b>
<b>1.1.2. Các Thành phần của Angular.....</b>	<b>5</b>
2. PostgreSQL.....	9
3. ASP.NET Core.....	10
4. Ngôn ngữ thiết kế web.....	12
<b>4.2. CSS.....</b>	<b>13</b>
<b>CHƯƠNG II: PHÂN TÍCH HỆ THỐNG.....</b>	<b>15</b>
<b>2.1. Khảo xác hiện trạng.....</b>	<b>15</b>
2.1.1. Giới thiệu chung.....	15
<b>2.1.2. Các tác nhân của hệ thống.....</b>	<b>16</b>
2.2. Mô tả các quy trình quản lý tại thư viện.....	16
2.2.7. Các vấn đề đặt ra cho hệ thống mới.....	18
2.2.8. Phạm vi nghiên cứu.....	18
2.2.9. Giới hạn hệ thống.....	18
2.2.10. Các chức năng của hệ thống mới.....	18
2.3. Thiết kế sơ đồ.....	21
<b>2.3.4. Biểu đồ phân rã với chức năng đăng kí.....</b>	<b>22</b>
<b>2.3.5. Biểu đồ phân rã với chức năng tìm kiếm.....</b>	<b>23</b>
<b>2.4. Sơ đồ tuần tự.....</b>	<b>25</b>
2.4.1. Sơ đồ tuần tự chức năng đăng nhập.....	25
2.4.2. Sơ đồ tuần tự chức năng đăng kí.....	25
2.4.3. Sơ đồ tuần tự chức năng thêm sản phẩm(độc giả hoặc sách).....	26

2.4.4. Sơ đồ tuần tự chức năng sửa thông tin sản phẩm (độc giả hoặc sách).....	26
2.4.5. Sơ đồ tuần tự chức năng xoá sản phẩm (độc giả hoặc sách).....	27
<b>2.4.6. Sơ đồ tuần tự chức năng tìm kiếm sản phẩm (độc giả hoặc sách).....</b>	<b>27</b>
<b>CHƯƠNG III: THỰC NGHIỆM XÂY DỰNG HỆ THỐNG.....</b>	<b>28</b>
<b>3. Một số giao diện của website.....</b>	<b>28</b>
3.1. Form đăng nhập.....	28
3.2. Form đăng kí.....	29
3.3. Form main.....	29
<b>KẾT LUẬN.....</b>	<b>34</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>36</b>

## DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Giải thích
API	Application Programming Interface
CSDL	Cơ sở dữ liệu
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
IDE	Integrated Development Environment
UC	Use Case
WS	WebStorms
RD	JetBrain Rider
pgA	Postgres

## DANH MỤC CÁC BẢNG

<b>Bảng 2.1: Bảng đặc tả UseCase đăng nhập.....</b>	<b>22</b>
<b>Bảng 2.2: Bảng đặc tả Use Case đăng kí.....</b>	<b>23</b>
<b>Bảng 2.3: Bảng đặc tả UseCase tìm kiếm.....</b>	<b>23</b>

## DANH MỤC CÁC HÌNH

Hình 1.1: Mô hình web api.....	11
Hình 1.2: Mô hình Web API.....	11
Hình 2.1: Biểu đồ chức năng nghiệp vụ.....	21
Hình 2.2: Biểu đồ lớp.....	22
Hình 2.3: Biểu đồ phân rã chức năng đăng nhập.....	22
Hình 2.4: Biểu đồ phân rã chức năng đăng kí.....	22
Hình 2.5: Biểu đồ phân rã chức năng tìm kiếm.....	23
Hình 2.6: Biểu đồ tuần tự chức năng đăng nhập.....	25
Hình 2.7: Biểu đồ tuần tự chức năng đăng kí.....	25
Hình 2.8: Biểu đồ tuần tự chức năng thêm sản phẩm(độc giả hoặc sách).....	26
Hình 2.9: Biểu đồ tuần tự chức năng sửa thông tin sản phẩm(độc giả hoặc sách).....	26
Hình 2.10: Biểu đồ tuần tự chức năng xóa sản phẩm(độc giả hoặc sách).....	27
Hình 2.11: Biểu đồ tuần tự chức năng tìm kiếm sản phẩm(độc giả hoặc sách).....	27
Hình 3.1 Giao diện đăng nhập.....	28
Hình 3.2: Giao diện đăng nhập.....	29
Hình 3.3: Giao diện quản lí độc giả (danh sách khoa).....	29
Hình 3.4: Giao diện khi thêm khoa mới.....	30
Hình 3.5: Giao diện quản lí độc giả (danh sách lớp).....	30
Hình 3.6: Giao diện quản lí độc giả (danh sách người đọc).....	31
Hình 3.7: Giao diện khi thêm người đọc.....	31
Hình 3.8: Giao diện quản lí sách (danh sách tác giả).....	32
Hình 3.9: Giao diện quản lí sách (danh sách phân loại).....	32
Hình 3.10: Giao diện quản lí sách (danh sách nhà xuất bản).....	33
Hình 3.11: Giao diện quản lí mượn trả sách.....	33



## LỜI NÓI ĐẦU

Từ trước tới nay, các thư viện dù lớn hay nhỏ đều chỉ hoạt động dựa theo cách thủ công do con người phụ trách. Nhưng xu thế hiện nay, khi sự phát triển mạnh mẽ của ngành công nghệ thông tin dẫn đến việc đòi hỏi ra đời các hệ thống để trợ giúp con người trong việc quản lý. Thông qua lĩnh vực phân tích và thiết kế hệ thống thông tin đã giúp việc quản lý đơn giản và nhanh chóng hơn.

Xuất phát từ nhu cầu thực tế cần có một hệ thống trợ giúp quản lý việc xuất nhập, thao tác trong các khâu của thư viện. Em đã thực hiện đồ án “Quản lý thư viện sách” mục tiêu để tìm hiểu sâu hơn trong lĩnh vực pháp tích thiết kế hệ thống thông tin. Xin chân thành cảm ơn thầy đã nhiệt tình giúp đỡ và truyền đạt cho em nhiều kiến thức quý giá về bộ môn phân tích và thiết kế hệ thống thông tin để em có đầy đủ khả năng thực hiện tốt đồ án này. Đây là 1 đồ án đầu tay nên khó tránh khỏi còn nhiều sai sót, khuyết điểm mong thầy thông cảm, hết lòng góp ý và chỉ cho em ưu, khuyết điểm để giúp em có thể hoàn thiện tốt hơn sau này.

Em xin chân thành cảm ơn !

### **Ý nghĩa đề tài :**

Trước hết, đề tài xây dựng website quản lý thư viện là một website được xây dựng như một đề tài thể hiện việc áp dụng những kiến thức quý báu đã được thầy cô của trường Đại Học Hải Phòng nhiệt tình truyền thụ lại cho em. Và đặc biệt là sự theo dõi và quan tâm giúp đỡ của cô Nguyễn Thị Thuý trong suốt thời gian em thực hiện đề tài này.

Thứ hai là những kinh nghiệm quý báu mà em có được trong quá trình thực hiện đề tài sẽ là hành trang tuyệt vời giúp ích rất nhiều trong công việc của em sau này.

Thứ ba trong thời đại CNTT phát triển như vũ bão ngày nay, công nghệ không ngừng được cải tiến, những công nghệ mới ra đời đòi hỏi người lập trình viên phải cố gắng nghiên cứu và tìm hiểu để áp dụng vào thực tiễn. Điều này cũng được thể hiện qua sự cố gắng khi quyết định thực hiện đề tài này với công nghệ được coi là mạnh nhất và được dự báo sẽ phát triển mạnh mẽ trong tương lai của em.

Do vậy em rất hy vọng sẽ tạo nên một sản phẩm hoàn chỉnh nhất có thể đáp lại sự nhiệt tình giúp đỡ của thầy cô và cả tâm huyết của em khi thực hiện đề tài này.

Nội dung báo cáo gồm 3 chương:

### **Chương 1: Cơ sở lý thuyết**

- Tìm hiểu về công nghệ .Net khái quát lên ưu nhược điểm và tầm quan trọng của công việc này trong việc xây dựng các hệ thống website hiện nay.

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

- Từ việc khảo sát hệ thống ban đầu, nội dung của phần này tập trung xác định các yêu cầu của hệ thống cần đáp ứng, xác định các chức năng cần có, luồng dữ liệu và các thực thể trong hệ thống, từ đó thiết kế CSDL, xử lý các chức năng thiết kế chương trình.

### **Chương 3: Thực nghiệm xây dựng hệ thống**

- Từ dữ liệu có được qua phần phân tích thiết kế hệ thống và ứng dụng từ việc tìm hiểu công nghệ .Net, nội dung chương 3 tập trung xác định môi trường triển khai và xác định hệ thống, đưa ra kết luận và rút kinh nghiệm

## Chương I: CƠ SỞ LÝ THUYẾT

### 1. CÁC CÔNG NGHỆ SỬ DỤNG

#### 1.1. Angular Framework

Angular là một javascript framework do google phát triển để xây dựng các Single Page Application (SPA) bằng JavaScript , HTML và TypeScript . Angular cung cấp các tính năng tích hợp cho animation , http service và có các tính năng như auto-complete , navigation , toolbar , menus ,... Code được viết bằng TypeScript , biên dịch thành JavaScript và hiển thị tương tự trong trình duyệt.

Để học được angular thì bạn cần biết các kiến thức cơ bản sau:

- HTML
- CSS
- JavaScript
- TypeScript
- Document Object Model (DOM)

##### *1.1.1. Các phiên bản của Angular*

**Angular js :** Phiên bản đầu tiên của angular là AngularJS được bắt đầu từ năm 2009 và được ra mắt vào 20/10/2010, do lập trình viên Misko Hevery tại Google viết ra như là một dự án kiểu “viết cho vui”. Lúc đó angular js được viết theo mô hình MVC (Model-View-Controller) trong đó :

- Model là thành phần trung tâm thể hiện hành vi của ứng dụng và quản lý dữ liệu.
- View được tạo ra dựa trên thông tin của Model .
- Controller đóng vai trò trung gian giữa Model và View và để xử lý logic .

##### **Angular 2 :**

• Sau phiên bản angular js thì vào tháng 3 năm 2015 phiên bản angular 2 ra đời nhằm thay thế Angular Js với các khái niệm mới nhằm đơn giản hóa và tối ưu cho quá trình phát triển sử dụng framework này. Angular 2 thay đổi hoàn toàn so với angular js bằng việc thay Controllers và \$scope ( Angular js ) bằng components và directives . Components = directives + template , tạo nên view của ứng dụng và xử lý các logic trên view. Angular 2 hoàn toàn được viết bằng Typescript. Angular 2 nhanh hơn angular js ,hỗ trợ đa nền tảng đa trình duyệt, cấu trúc code được tổ chức đơn giản và dễ sử dụng hơn.

#### **Angular 4 :**

- Ra mắt vào tháng 3/2017 đây là một phiên bản nâng cấp từ Angular 2 nên kiến trúc không thay đổi nhiều ngoài việc giảm thiểu code được tạo ra từ đó giảm kích thước tệp được đóng gói xuống 60%, đẩy nhanh quá trình phát triển ứng dụng.

#### **Angular 5 :**

- Đã được phát hành vào ngày 1 tháng 11 năm 2017 với mục tiêu thay đổi về tốc độ và kích thước nên nó nhanh hơn và nhỏ hơn angular 4. Các tính năng mới so với angular 4:

- Sử dụng HTTPClient thay vì sử dụng HTTP : bởi vì nó nhanh, an toàn và hiệu quả hơn.

- Với phiên bản Angular 5 mặc định sử dụng RxJs 5.5

- Multiple export aliases : Một component có thể được xuất bằng nhiều bí danh (aliases) để giảm bớt quá trình di chuyển.

- Internationalized Pipes for Number, Date, and Currency: Các pipe mới được giới thiệu để tiêu chuẩn hóa tốt hơn.

- Tối ưu hóa build production bằng việc sử dụng công cụ build optimizer được tích hợp sẵn vào trong CLI. Công cụ này tối ưu tree shake và loại bỏ code dư thừa.

- Cải thiện tốc độ biên dịch bằng việc dùng TypeScript transforms, giờ đây khi build sẽ sử dụng lệnh “ng serve –aot”. AOT sẽ cải thiện performance khi load page và nó được dùng để deploy app lên production

#### **Angular 6:**

- Cập nhật CLI, command line interface: thêm 1 số lệnh mới như ng-update để chuyển từ version trước sang version hiện tại; ng-add để thêm các tính năng của ứng dụng để trở thành một ứng dụng web tiến bộ.

- Angular Element: Cho phép các component của Angular được triển khai dưới dạng component web, sau đó có thể được sử dụng trong bất kỳ trang HTML nào một cách dễ dàng.

- Multiple Validators: cho phép nhiều Validators được áp dụng trên form builder.

- Tree-shakeable providers: giúp loại bỏ mã code chết.

- Sử dụng RxJS 6 với syntax thay đổi.

### **Angular 7 :**

- Được phát hành vào 18 tháng 10 năm 2018 với những thay đổi như :
- ScrollingModule : Để scroll load dữ liệu.
- Drag and Drop: Chúng ta có thể dễ dàng thêm tính năng kéo và thả vào một mục
- Angular 7.0 đã cập nhật RxJS 6.3

### **Angular 8 :**

- Ra mắt 28 tháng 5 năm 2019 với CLI workflow improvements, Dynamic imports for lazy routes ...

### **Angular 9 :**

- Ra mắt mới đây 6 tháng 2 năm 2020, Angular 9 di chuyển tất cả các ứng dụng để sử dụng trình biên dịch Ivy và thời gian chạy theo mặc định. Angular đã được cập nhật để hoạt động với TypeScript 3.6 và 3.7

### **Angular 10:**

Angular 10, phiên bản lớn gần đây nhất của Angular được ra mắt vào Tháng 6 Năm 2020, chỉ sau Angular 9 4 tháng, bao gồm các tính năng mới: Bổ sung các thành phần Angular Material mới, hỗ trợ TypeScript 3.9, TSlib làm mới sang v2.0 và chức năng tùy chỉnh trình duyệt mặc định mới.

### **Angular 11:**

Ra mắt vào ngày 11/11/2020.

### **Angular 12:**

Ra mắt vào ngày 12/5/2021.

#### ***1.1.2. Các Thành phần của Angular***

##### ***1.1.2.1. Component***

Components là một khối code trong app Angular. Nó là sự kết hợp của bộ template html và những kèm code TypeScript (hoặc Javascript). Các components là độc lập với nhau và độc lập với hệ thống. Nó có thể được cài vào hoặc tháo ra khỏi hệ thống dễ dàng. Một component có thể hiểu như một control trên màn hình hiển thị, gồm giao diện html và code logic xử lý sự kiện đi kèm control đó. Một component cũng có thể to lớn như là cả 1 màn hình chứa nhiều control hoặc một nhóm nhiều màn hình. Tức là một component cũng có thể chứa và gọi được nhiều component khác nối vào

Cấu trúc của Component:

Từ khóa **@Component** sẽ giúp định nghĩa bộ khung html cho nó. Và bên dưới là một class `CategoryListComponent` dùng để viết code logic. Trong định nghĩa bộ khung html, chúng ta có một số thuộc tính cần chú ý sau đây:

- **selector** : Là tên được đặt để gọi một component trong code html. Ở ví dụ vừa rồi, từ khóa `hello-ng-world` được đặt tên cho component này. Khi cần gọi component này ra ở màn hình html cha, ta sẽ gọi bằng html tag `<app-category-list></app-category-list>`. Gọi như vậy thì component con sẽ được render ra component cha.\*\*

- **template** : Là tự định nghĩa khung html cho component .

- **templateUrl** : Là đường dẫn url tới file html bên ngoài để load file đó vào làm khung html cho component này.

- **styles** : Là viết style css luôn vào file component này. Cách này chỉ dùng cho component đơn giản.

- **styleUrls** : Là đường dẫn url đến file style css độc lập cho component này.

#### 1.1.2.2. Binding

Angular có cách code Binding (kết nối giữa html và data) dữ liệu theo kiểu 2 chiều, nghĩa là html input thay đổi thì biến javascript sẽ ngay lập tức nhận được giá trị trả về và ngược lại, giá trị trong js thay đổi thì ngay lập tức màn hình html thay đổi theo. Để bind một chuỗi ra ngoài màn hình html thì rất đơn giản sử dụng 2 dấu ngoặc nhọn `{{TenBien}}` Ví dụ chúng ta có một Component đơn giản như sau:

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-category-list',
  templateUrl: './category-list.component.html',
})

export class CategoryListComponent {
  title = 'Xin chào';
}
```

```
<h1>Hello {{title}}</h1>
```

### 1.1.2.3 . Xử lý sự kiện

Để gắn 1 sự kiện của control html với một hàm javascript, ta viết như sau:

```
<button type="button" class="btn default" (click)="goBack()">
  <i class="fa fa-undo" aria-hidden="true"></i>Back
</button>
```

Khi chúng ta muốn (click) tức là muốn bind sự kiện click chuột của người dùng ở control này gọi là goBack().

### 1.1.2.4. NgModule

Module là một khái niệm rộng nhất của Angular. Một module có thể bao gồm chứa các components, directives, pipes, v.v. Module có thể được biên dịch (compile) dưới dạng ahead-of-time (AoT). Nghĩa là biên dịch ra mã thực thi để hiện ra luôn trên trình duyệt không cần vẽ vời gì từ đầu. Hãy tưởng tượng component có html và js viết riêng, khi load trang thì 2 thứ này mới nhào nặn chung để hiển thị html+data lên màn hình. AoT là thứ html+data đã nhào sẵn. Module cũng có thể gọi module con và bắt tay được với các module khác. Ví dụ về module chúng ta có thể bắt gặp ngay ở trong category.module.ts

```
import { NgModule } from '@angular/core';

@NgModule({
  imports: [ ],
  declarations: [ ],
  bootstrap: [ ]
})
export class CategoryModule { }
```

Các thuộc tính của module được định nghĩa như sau:

- imports: Định nghĩa sự phụ thuộc (Dependency) của module này, module phụ thuộc sẽ được load trước rồi module này mới load.
- declarations: Định nghĩa tất cả các component sẽ được dùng trong module này. Nếu chưa định nghĩa thì các component trong module sẽ không thể gọi nhau vì không tìm thấy nhau.
- bootstrap: Mỗi ứng dụng Angular đều cần một module gốc, module này sẽ có một component gốc chứa layout gốc sẽ được render ra ở file index.html.

### 1.1.2.5. Service

Để tạo ra một service thì chúng ta cần import và mô tả một class với từ khóa **@injectable** lấy từ **@angular/core** module.

```
import { Injectable } from '@angular/core';

@Injectable()
export class CategoryService {

}
```

### 1.1.2.6. Các tính năng nổi bật của Angular

Là một phần thiết yếu của hệ sinh thái Javascript, Angular là công cụ lập trình web front-end phổ biến nhất hiện nay. Mỗi phiên bản Angular đều có rất nhiều tính năng và được thường xuyên nâng cấp để trở nên thông minh hơn.

Chúng ta hãy cùng tìm hiểu về các tính năng nổi bật của Angular:

#### *Các tính năng chung*

- Liên nền tảng (Cross-platform): Với Angular bạn có thể phát triển các ứng dụng web cấp tiến (Progressive Web Applications – PWA). PWA cũng cung cấp trải nghiệm như ứng dụng (app-like) cho người dùng sử dụng các công nghệ web tân tiến. Tùy theo nhu cầu thì bạn có thể triển khai một ứng dụng gốc cũng như cấp tiến. Bộ công cụ phát triển phần mềm kết hợp được gọi là Ionic có thể đưa ứng dụng của bạn lên app store hoặc triển khai nó lên mobile web như một PWA. Bạn cũng có thể phát triển ứng dụng máy tính bằng Angular.

- Tốc độ cao và tối ưu hiệu năng: Angular định nghĩa lại máy ảo JavaScript hiện đại bằng cách chuyển các template thành code. Bởi vậy, code viết tay của bạn cũng có thể được tận dụng trong framework này. Điều tuyệt vời nhất của Angular đó là bạn có thể render code thành HTML & CSS và có cái nhìn đầu tiên về ứng dụng trên các nền tảng khác như Node.js, .NET, PHP,..v.v..

Thời gian tải của Angular nhanh hơn bất kỳ front-end framework nào ở thời điểm hiện tại. Angular tải cực kì nhanh với bất kỳ router thành phần mới nào. Hơn nữa, code sẽ tự động được chia ra đối với mỗi khi người dùng load và yêu cầu render.

- Các ứng dụng Angular dành cho mọi người: Angular là một nền tảng front-end kì diệu không chỉ cho phép lập trình các ứng dụng ấn tượng, mà nó còn có thể tạo ra các hoạt ảnh cao cấp để nâng cao trải nghiệm người dùng, Các API của Angular linh



hoạt đến mức các lập trình viên có thể phát triển các bố trí giao diện phức tạp cũng như các hoạt ảnh với sử dụng ít code.

Hơn nữa, với các framework kiểm thử đơn vị thông minh như Jasmine và Karma, bạn có thể sửa lại code bất kỳ lúc nào, Angular có hơn 11 module kiểm thử được tích hợp sẵn để đảm bảo code không dính lỗi.

## 2 . PostGresSQL



PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ – đối tượng, được phát triển bởi Khoa Điện toán, Đại học California – Hoa Kỳ dựa trên Postgres bản 4.2. Chương trình này đã mở đường cho nhiều khái niệm về hệ quản trị dữ liệu thương mại sau này.

Ban đầu, hệ quản trị được thiết kế để chạy trên các nền tảng tương tự như Unix. Sau này, PostgreSQL được điều chỉnh trở nên linh động và chạy trên nhiều nền tảng khác nhau như Windows, Mac OS X, Solaris với nhiều tính năng và đặc điểm nổi bật.

PostgreSQL là mã nguồn mở miễn phí, được xây dựng theo chuẩn SQL99. Người dùng có thể tự do sử dụng, chỉnh sửa và phân bổ PostgreSQL theo nhiều hình thức khác nhau.

So với nhiều hệ quản trị cơ sở dữ liệu khác, PostgreSQL không quá yêu cầu về công tác bảo trì bởi tính ổn định cao, có thể phát triển nhiều ứng dụng khác nhau với chi phí tương đối thấp.

PostgreSQL sở hữu một hệ tính năng đa dạng giúp hỗ trợ các nhà phát triển xây dựng app, các nhà quản trị bảo vệ toàn vẹn dữ liệu, và tạo ra một môi trường chịu lỗi fault-tolerant giúp bạn quản lý dữ liệu bất kể tập dữ liệu lớn hay nhỏ. Bên cạnh hệ thống nguồn mở và miễn phí, PostgreSQL cũng có khả năng mở rộng tuyệt vời. Ví dụ, bạn có thể định nghĩa các kiểu dữ liệu riêng của bạn, xây dựng các hàm tùy chỉnh, hay viết mã từ các ngôn ngữ lập trình khác nhau mà không cần biên dịch lại cơ sở dữ liệu!

PostgreSQL tuân theo tiêu chuẩn SQL nhưng không mâu thuẫn với các tính năng truyền thống hay có thể dẫn đến các quyết định kiến trúc gây hại. Nhiều tính

năng theo tiêu chuẩn SQL được hỗ trợ, tuy nhiên đôi khi có thể có cú pháp hoặc hàm hơi khác một chút.

### 3 . ASP.NET Core

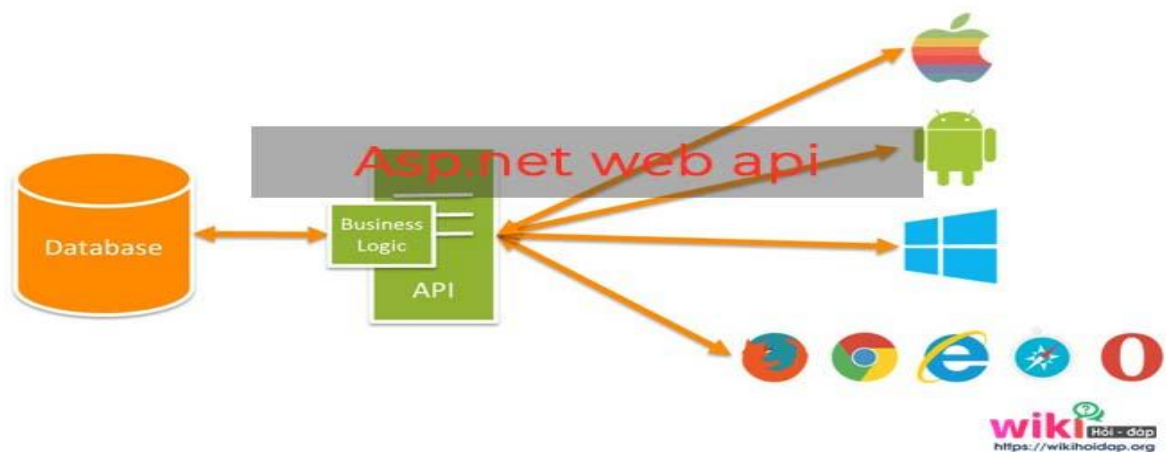


ASP.NET Core là tập hợp thư viện như một Framework mới được xây dựng ứng dụng web khi có kết nối internet, nó là một sản phẩm của Microsoft khá nổi tiếng trong cộng đồng lập trình hiện nay khi có rất nhiều blogger, vblog về công nghệ thể hiện sự chú ý và quan tâm đến nó. Ngay khi vừa xuất hiện đã có hàng loạt các ASP.NET Core tutorial, bài viết so sánh, hướng dẫn, thảo luận được đưa ra mổ xẻ.

Với ASP.NET các bạn có thể áp dụng để làm nhiều mục đích làm việc khác nhau của bạn ví dụ như: Xây dựng các ứng dụng web, các dịch vụ, ứng dụng IoT và các phần backen mobile; sử dụng các công cụ phát triển ưa thích trên Windows, macOS và Linux; Triển khai trên đám mây hoặc ngay tại chỗ. Với quá nhiều tác vụ như vậy thì bản chất thực sự của ASP.NET Core là gì ?ASP.NET Core không phải là phiên bản tiếp theo của ASP.NET. Nó là một cái tên mới được xây dựng từ đầu. Nó có một sự thay đổi lớn về kiến trúc và kết quả là nó gọn hơn, phân chia module tốt hơn.

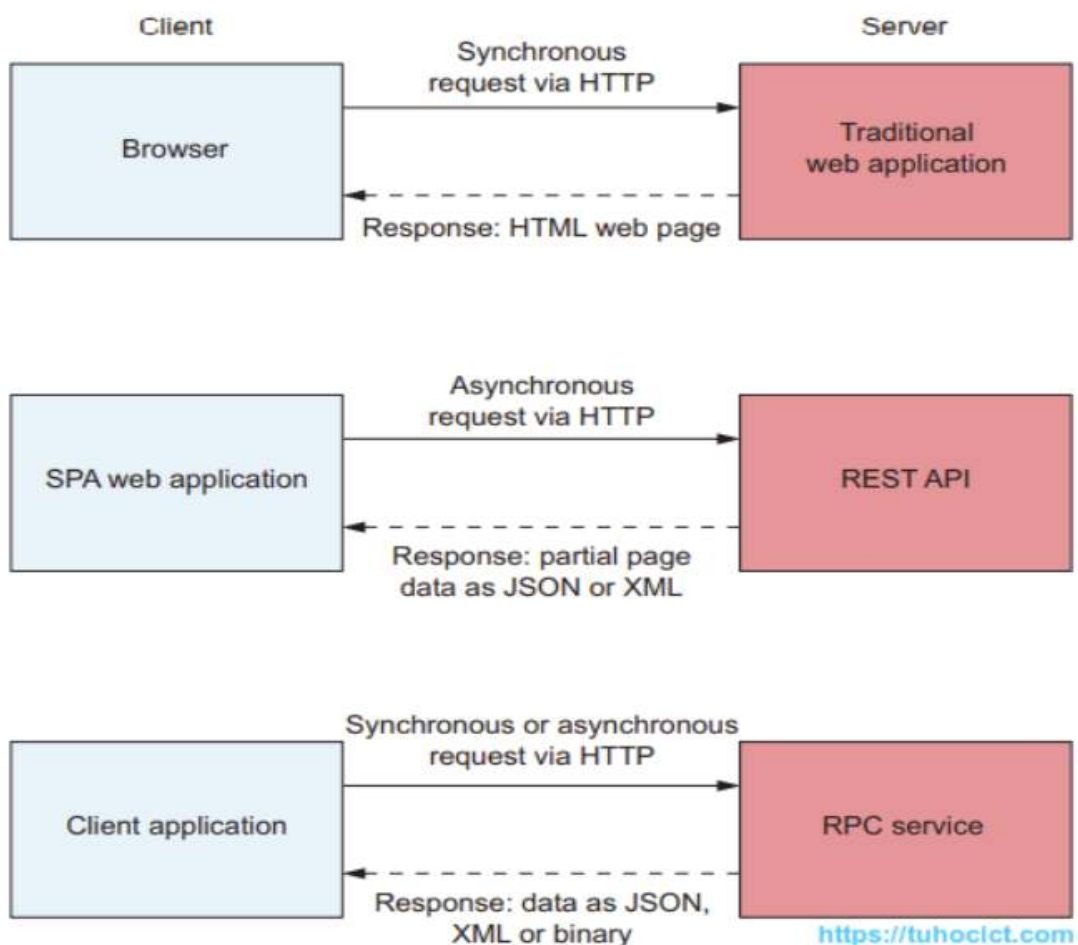
ASP.NET Core đã được xuất hiện từ lâu và được biết đến như là một sản phẩm của .Net Framework. ASP.NET được cộng đồng ủng hộ và phát triển ổn định đến nay. Tuy nhiên đã có sự thay đổi mạnh mẽ khi ASP.NET Core xuất hiện, mang những đặc tính của công nghệ lập trình hiện đại.

ASP.NET Core là kết quả của sự học hỏi từ các Framework Module hóa khác. Nó chạy trên cơ sở là tập hợp của các gói Module được gọi là Nuget Packages. Nuget Packages cho phép bạn tối ưu thiết kế ứng dụng chỉ bao gồm những gì cần thiết. Cho nên những ứng dụng được thiết kế trên nền tảng ASP.NET cho ra kết quả có dung lượng nhỏ hơn, bảo mật tốt hơn, hiệu năng hoạt động tốt và ổn định. Quá trình xây dựng cũng không còn quá phức tạp nên vấn đề phát triển và bảo trì sau này cũng dễ hơn nhiều. Mô hình web API trong ASP.net Core:



Hình 1.1: Mô hình web api

**API** là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của **Application Programming Interface** – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.



Hình 1.2: Mô hình Web API

1. Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.

2. Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

3. Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.

4. Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu...

#### **4. Ngôn ngữ thiết kế web**

##### **4.1. *Html***



HTML (HyperText Markup Language) còn được gọi là ngôn ngữ đánh dấu siêu văn bản, được tạo ra nhằm cấu trúc lên một trang web với các mẫu thông tin được trình bày trên World Wide Web.

HTML được sử dụng để tạo bố cục, cấu trúc trang web. Nó có một số ưu điểm sau:

- 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,

...)

## 4.2. CSS



CSS (Cascading Style Sheet). Website được cấu tạo từ các thẻ html nhưng với những thẻ html thì mới chỉ thể hiện được bộ khung của website. Để căn chỉnh, trình bày cho đẹp mắt thì ta cần sử dụng ngôn ngữ CSS. Đây là ngôn ngữ được dùng rất nhiều trong lập trình web, thường đi cùng với ngôn ngữ Html.

Chúng ta có 3 cách để nhúng thuộc tính CSS vào trong file Html:

**Internal:** mã CSS trong chính văn bản HTML, nằm trong khối thẻ <style>

**External:** mã CSS ở một file riêng biệt sau đó nạp vào HTML bằng phần tử <link>

**Inline:** mã CSS viết tại thuộc tính style của phần tử HTML

Trong thực tế, cách thứ 2 được các lập trình viên sử dụng nhiều nhất do tính tiện dụng và linh hoạt.

## 4.3. C#



**C#** (đọc là "**C thăng**" hay "**C sharp**" ("*xi-sáp*") là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft, là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là **C#**, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

#### 4.4. SQLServer



SQL là ngôn ngữ phi thủ tục, không yêu cầu cách thức truy cập cơ sở dữ liệu như thế nào. Tất cả các thông báo của SQL rất dễ dàng sử dụng và ít mắc lỗi. SQL cung cấp các tập lệnh phong phú cho các công việc hỏi đáp dữ liệu như:

- Chèn, xóa và cập nhật các hàng trong 1 quan hệ
- Tập, thêm, xóa và sửa đổi các đối tượng trong của cơ sở dữ liệu.
- Điều khiển việc truy cập tới cơ sở dữ liệu và các đối tượng của cơ sở dữ liệu để đảm bảo tính bảo mật, tính nhất quán và sự ràng buộc của cơ sở dữ liệu.
- Đối tượng của SQL server là các bảng dữ liệu với các cột và các hàng. Cột được gọi là trường dữ liệu và hàng là bản ghi của bảng. Cột dữ liệu và kiểu dữ liệu xác định tạo nên cấu trúc của bảng. Khi bảng được tổ chức thành một hệ thống cho một mục đích sử dụng cụ thể vào công việc nào đó sẽ trở thành một cơ sở dữ liệu.

## CHƯƠNG II: PHÂN TÍCH HỆ THỐNG

### 2.1. Khảo xác hiện trạng

#### 2.1.1. Giới thiệu chung

Thư viện Trường Đại học Hải Phòng quản lý khoảng 20000 đầu sách và tạp chí, phục vụ cho học sinh, sinh viên, giáo viên (độc giả) của trường. Độc giả có thể mượn sách về nhà hoặc đọc tại chỗ. Việc phân công quản lý trong thư viện được thực hiện như sau:

- Bộ phận quản lý: Gồm 2 người: Chủ nhiệm và phó chủ nhiệm.
- Chủ nhiệm thư viện: điều hành chung toàn bộ các công tác trong thư viện.
- Phó chủ nhiệm: có nhiệm vụ quản lý về cơ sở vật chất (quản lý tài liệu,...)

Bộ phận nghiệp vụ: Gồm 3 tổ chuyên môn :

- Tổ nghiệp vụ: bổ sung tài liệu định kỳ, phân loại tài liệu (phân loại theo chuyên ngành khoa học hoặc theo khu phân chia đã có), mô tả tài liệu để đưa lên mục tra cứu tài liệu (tra theo truyền thống hoặc tra điện tử).

- Tổ thông tin tư liệu: tạo lập CSDL của các đầu sách, xây dựng hoàn thiện cho bộ máy điện tử.

- Tổ phục vụ bạn đọc: Có trách nhiệm cấp thẻ độc giả, lập các phiếu mượn, trả sách, phân công phục vụ các quầy mượn trên kho.

Sách được phân chia thành các kho:

- Kho giáo trình: Kho này sinh viên không được mượn mà chỉ có lớp trưởng của các lớp mượn vào đầu mỗi kỳ học.

- Kho sách tham khảo: gồm có kho vb và vv, ở tầng 3. Kho vv chứa các sách loại vừa (khoảng > 20,5cm). Kho vb chứa các loại sách có kích cỡ bé.(<20,5cm).

- Kho sách ngoại văn: kho này chứa các tài liệu tiếng nước ngoài, nếu là tiếng anh thì có kí hiệu sách là L, tiếng Trung là H...

- Kho tra cứu và tạp chí : chứa các tài liệu như từ điển , sổ tay...

- Kho cao học và tài liệu quý hiếm: chứa tài liệu nghiên cứu khoa học và tài liệu quý, nằm ở tầng 4.

- Đối với tài liệu ở các kho giáo trình, kho sách tham khảo và kho sách ngoại văn, độc giả có thể mượn sách về nhà.

- Các kho còn lại, độc giả chỉ được đọc tại chỗ.

### **2.1.2. Các tác nhân của hệ thống**

**Người quản lí :** là người đăng nhập vào hệ thống sử dụng các chức năng như thêm , sửa, xoá , tìm kiếm độc giả hoặc sách xử lí các vi phạm.

## **2.2. Mô tả các quy trình quản lý tại thư viện**

### **2.2.1. Nhập sách**

(Do bộ phận nghiệp vụ thực hiện) Theo định kỳ khoảng 2 tháng một lần, thư viện có bổ sung sách mới về cho kho, việc đặt mua sách được thực hiện như sau:

- Nhà xuất bản sẽ đăng thông tin sách mới ra trên trang web của mình, thư viện xem xét, chọn những sách cần mua và có đơn đặt hàng, NXB sẽ gửi các danh mục sách kèm theo giá về cho thư viện, thư viện lập danh sách những sách cần mua. Sau khi được hiệu trưởng thông qua, thư viện sẽ tiến hành lập hợp đồng với NXB. Hóa đơn sẽ được gửi cho bộ phận tài vụ của nhà trường thanh toán. Sau đó thư viện nhận sách về.

- Trong trường hợp sách nhận về không đạt yêu cầu, thư viện sẽ gửi trả lại sách cho NXB theo điều khoản đã có trong hợp đồng.

- Ngoài ra nếu nhà sách nào đáp ứng được nhu cầu của thư viện thì thư viện cũng làm việc với họ.

- Sách sau khi mua về sẽ được bộ phận nghiệp vụ tiến hành phân loại. Việc phân các đầu sách vào các kho tùy theo loại sách, kích cỡ sách để đưa vào các kho khác nhau và lập thẻ quản lý cho sách.

- Mỗi khi có bổ sung sách mới vào thì tổ nghiệp vụ có trách nhiệm rà xét xem số sách đó đã có hay chưa, nếu chưa có thì tiến hành tạo lập thẻ quản lý và cho mã số mới. Còn đã có rồi thì ta chỉ việc cập nhật số lượng thêm.

### **2.2.2. Cấp thẻ độc giả**

Hàng năm thư viện tiến hành làm thẻ thư viện cho học sinh, sinh viên mới trong trường. Thư viện dựa vào danh sách yêu cầu làm thẻ của các lớp để tiến hành làm thẻ cho học sinh, sinh viên. Sau khi hoàn thành thủ tục đăng ký, học sinh, sinh viên sẽ được tổ phục vụ bạn đọc cấp thẻ thư viện (thẻ bạn đọc).

- Thẻ thư viện gồm các thuộc tính: Số thẻ, họ tên, ngày sinh, lớp, khoa, địa chỉ.

- Đối với những độc giả mất thẻ, muốn làm lại thì phải có đơn yêu cầu. Thẻ được cấp với mã mới.

- Đối với sinh viên ở lại lớp, khi hết thời hạn sử dụng thẻ. Sinh viên phải làm lại thẻ nếu muốn mượn sách.



- Đối với giáo viên, nhân viên trong trường và cán bộ thư viện muốn mượn sách họ cũng phải làm thủ tục như sinh viên.

### **2.2.3. Mượn trả sách**

Để mượn được sách, độc giả tiến hành tra cứu (tra truyền thống hay tra điện tử) ở danh mục có sẵn của thư viện. Sau đó, sẽ điền các thông tin cần thiết vào phiếu yêu cầu mượn sách. Nhân viên phục vụ bạn đọc căn cứ vào phiếu này để tìm, đồng thời kiểm tra sách có còn trong kho hay không để cho độc giả mượn.

Phiếu yêu cầu bao gồm các thông tin: Họ tên, tên sách1, tên sách2, ký hiệu, ngày mượn, ký tên của người giao sách và của người mượn.

Khi mượn sách, mỗi độc giả chỉ được mượn 1 cuốn, thời hạn mượn là 7 ngày. Độc giả có thể gia hạn mượn sách thêm 1 lần nữa.

Độc giả là giáo viên, nhân viên, cán bộ thư viện thì có thể mượn được nhiều sách và thời hạn mượn có thể lâu hơn độc giả là học sinh, sinh viên.

Các hình thức xử phạt của thư viện:

- Khi độc giả trả sách, nhân viên phục vụ bạn đọc sẽ xem trên phiếu trả sách, nếu quá 7 ngày thì độc giả bị xử phạt 500 đồng /1 ngày. Thư viện cũng có hình thức xử phạt thích đáng cho những độc giả trả sách không còn nguyên vẹn.

- Trường hợp làm mất sách, độc giả phải mua đền đúng sách đó, nếu không có sách thì độc giả phải đền bằng tiền theo giá sách, đồng thời chịu một mức phạt theo quy định.

- Đối với những học sinh, sinh viên không trả sách cho thư viện thì cuối khóa thư viện sẽ gửi danh sách cho phòng giáo vụ và đối tượng đó sẽ không được nhận bằng tốt nghiệp. Thư viện cũng xử phạt tương tự đối với lớp trưởng của những đối tượng này.

- Thư viện luôn cập nhật tin tức với phòng giáo vụ để tránh nhầm lẫn trong việc không cấp bằng tốt nghiệp cho lớp trưởng của những học sinh, sinh viên bị đuổi học.

### **2.2.4. Hủy đầu sách**

Hàng năm, có kiểm tra định kỳ các kho sách. Các sách bị hư hỏng (không dùng được nữa) hoặc sách không có độc giả mượn được lập thành danh sách.

Quyết định hủy sách do hội đồng (có chủ nhiệm thư viện, phó chủ nhiệm và các thành viên nhà trường) đưa ra. Sách có quyết định hủy được lấy ra khỏi kho và giao cho bộ phận bảo quản sách xử lý. Bộ phận xử lý thông tin loại các đầu sách này ra khỏi CSDL.

### **2.2.5. Hủy độc giả**

Đối với độc giả là học sinh, sinh viên, thẻ có giá trị sử dụng trong suốt khóa học. Hết thời hạn trên, thẻ sẽ bị hủy.

Về việc quản lý độc giả của thư viện vẫn còn hạn chế, đó là: chưa quản lý được trường hợp sinh viên bị thôi học mà vẫn tiếp tục mượn được sách.

### **2.2.6 . Báo cáo thống kê**

Đối với công tác phòng đọc, ngoài công việc phục vụ bạn đọc, định kỳ hàng tháng hay theo từng quý nhân viên còn phải thống kê, lập báo cáo về số sách đã mượn, hiện trạng của sách, độc giả, danh sách các sách cần mua bổ sung (căn cứ vào phiếu yêu cầu của độc ) gửi lên ban chủ nhiệm thư viện.

### **2.2.7. Các vấn đề đặt ra cho hệ thống mới**

Nhận xét tình hình thực tế: Qua thực tế nghiên cứu, nhận thấy rằng số lượng công việc trong thư viện thì nhiều mà hầu hết các công việc được làm thủ công nên có nhiều bất tiện như:

- Tốn nhiều thời gian trong việc lập báo cáo và thống kê
- Gây nhiều bất tiện trong việc quản lý số sách và quản lý độc giả

Xuất phát từ những nhược điểm của hệ thống quản lý cũ của thư viện nên việc tin học vào công tác quản lý là việc làm hợp lý

Từ những công việc thực tế, đưa ra hệ thống quản lý thư viện với các chức năng sau:

- Quản lý kho sách
- Quản lý độc giả
- Quản lý mượn trả sách
- Báo cáo thống kê

### **2.2.8. Phạm vi nghiên cứu**

Nghiên cứu quá trình mượn trả sách của sinh viên, và quá trình quản lý sách trong thư viện.

### **2.2.9. Giới hạn hệ thống**

Hệ thống không phân chia đối tượng độc giả, không thực hiện quản lý nhân viên, các vấn đề liên quan đến tài chính, và các cơ sở vật chất khác.

### **2.2.10 . Các chức năng của hệ thống mới**

Hệ thống xây dựng trên cơ sở là một ứng dụng theo mô hình client – server.

Với ứng dụng server:

- Để sử dụng chương trình người sử dụng phải đăng nhập vào hệ thống .

Đối tượng sử dụng :

- Admin, người thủ kho, người quản lý thông tin độc giả, người thực hiện cho mượn trả sách

Với ứng dụng client :

- Không cần thiết phải đăng nhập để sử dụng chương trình

Đối tượng sử dụng:

- Độc giả

Chức năng đăng nhập:

- Tuỳ vào tài khoản đăng nhập mà hệ thống sẽ cho phép người dùng sử dụng những chức năng nào.

- Do database đặt tại máy chứa chương trình dùng cho server nên thông tin đăng nhập ngoài tài khoản người dùng còn thêm tên server và mật khẩu server

Chức năng mượn sách:

- Độc giả sẽ sử dụng bản client để tìm kiếm sách nếu tìm thấy sách cần mượn sẽ gửi thông tin sách cần mượn đến người quản lý mượn trả.

- Do có nhiều người thực hiện thao tác cho mượn trả sách nên muốn gửi đi độc giả phải xác định người nhận tin gửi (Thông qua tên máy nhận đã được cài đặt sẵn).

- Thông tin gửi đi bao gồm tên sách mã sách tài khoản của độc giả và tên máy đã gửi thông tin.

- Nếu hệ thống kiểm tra tài khoản người mượn là đúng sẽ cho phép gửi thông tin.

- Sau khi gửi thông tin độc giả ngồi đợi thông báo phản hồi từ người quản lý mượn trả (có thể cho mượn sách đó hay không hoặc thông báo đến nhận sách)

Chức năng tìm kiếm:

- Người dùng chương trình có thể thực hiện các thao tác tìm kiếm như: tìm kiếm sách trong kho, thông tin người mượn, sách đã mượn, sách đang mượn v.v...

Chức năng cho mượn sách :

- Khi một độc giả gửi yêu cầu mượn sách đến người cho mượn trả sách người này phải thực hiện thao tác kiểm tra xem sách có thể cho mượn được không nếu được sẽ lấy sách và gửi thông báo nhận sách cho độc giả yêu cầu

- Nếu độc giả không gửi yêu cầu mượn sách mà trực tiếp đến liên hệ với nhân viên mượn trả sách thì nhân viên cần kiểm tra tài khoản của độc giả trước khi thực hiện thao tác cho mượn sách.

- Trong khi thực hiện thao tác cho mượn sách hệ thống sẽ kiểm tra tư cách mượn của độc giả (đã mượn sách chưa trả), và tư cách sách ( sách là giáo trình hay là các loại khác) mà kiểm tra xem có thể cho độc giả này mượn quyền sách này hay không.

Chức năng trả sách :

- Để thực hiện thao tác này độc giả phải đem sách đến người quản lý mượn trả. Người này phải kiểm tra tư cách mượn của độc giả (có quá hạn mượn sách hay không) sau đó mới thực hiện thao tác cho độc giả trả sách

Chức năng quản lý tài khoản độc giả :

- Người mượn muốn được tạo thẻ độc phải gửi cho thư viện các thông tin số thẻ, họ tên, ngày sinh, lớp, khoa, địa chỉ.

- Hệ thống hỗ trợ các thao tác tạo mới thẻ làm lại thẻ.

- Ngoài ra hệ thống hỗ trợ trong việc in thẻ cho độc giả ( hình của độc giả đã scan vào máy)

Chức năng quản lý sách :

Phân loại sách :

- Phân loại cho sách mới nhập về kho hay thay đổi lại loại sách

- Cập nhật sách khi sách mới nhập về người thủ kho phải nhập thông tin cần thiết cho sách như tên sách, tác giả, năm xuất bản v.v...

- Hệ thống sẽ tự động tạo mã cho sách

Kiểm kê sách :

- Kiểm tra số lượng sách trong kho sách đã mượn v.v...

Chức năng quản lý tài khoản :

Tạo tài khoản :

- Người Admin mới có quyền tạo tài khoản mới, khi tạo tài khoản sẽ không tạo mật khẩu mà để mật khẩu trống cho người truy cập lần sau vào đặt mật khẩu

Thay đổi thông tin :

- Người dùng có thể thay đổi thông tin cá nhân của mình khi đăng nhập vào hệ thống, thay đổi mật khẩu đăng nhập . Tuy nhiên không thể thay đổi quyền truy cập

Phân quyền :

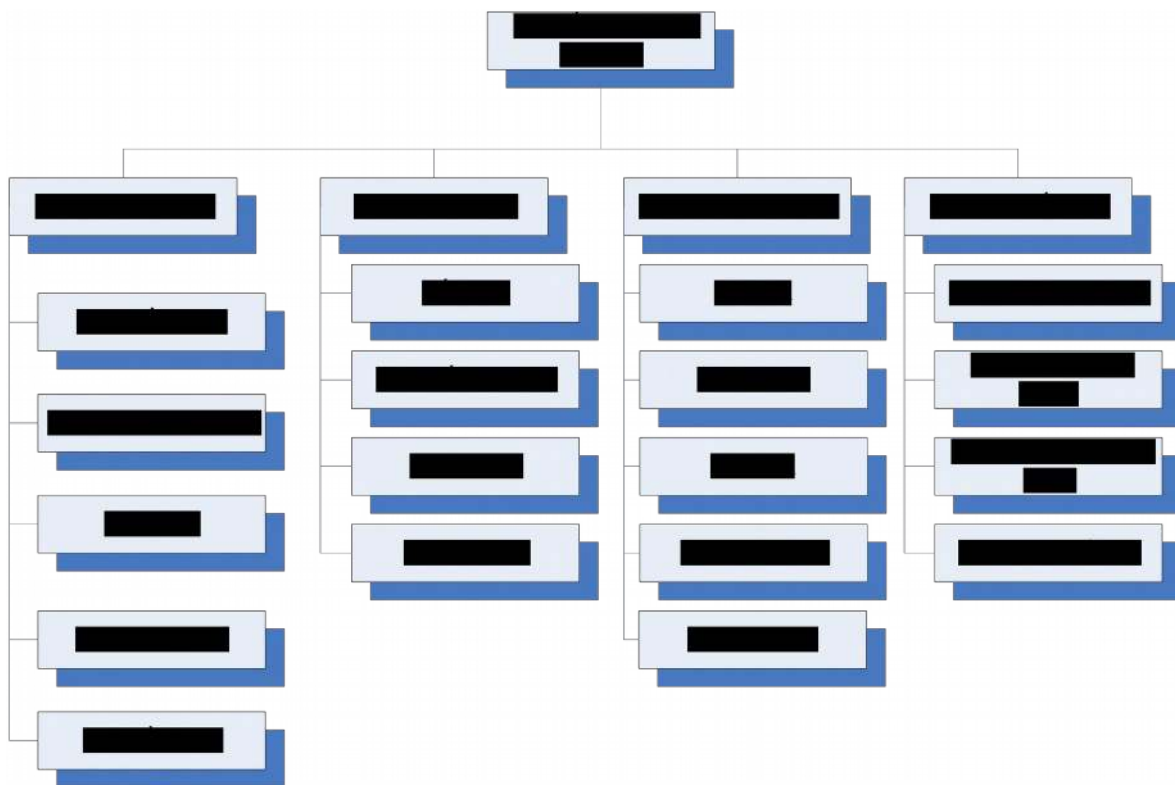
- Chức năng này chỉ có ở Admin, do yêu cầu công việc tại thư viện một người có thể đảm nhiệm nhiều công việc nên mục phân quyền này một tài khoản cũng có thể có nhiều quyền hạn tùy vào việc Admin phân cho

Chức năng báo cáo thống kê:

- Thống kê sách cho mượn trong ngày sách quá hạn, sách đang mượn v.v....

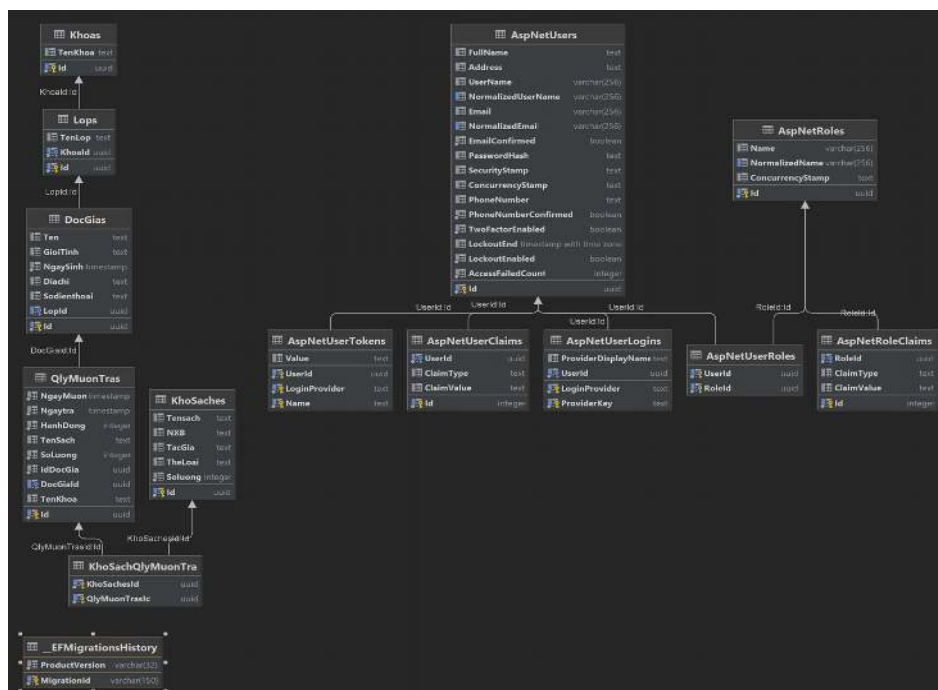
## 2.3. Thiết kế sơ đồ

### 2.3.1. Biểu đồ chức năng nghiệp vụ



Hình 2.1: Biểu đồ chức năng nghiệp vụ

### 2.3.2. Biểu đồ Lớp



Hình 2.2: Biểu đồ lớp

### 2.3.3 . Biểu đồ phân rã với chức năng đăng nhập

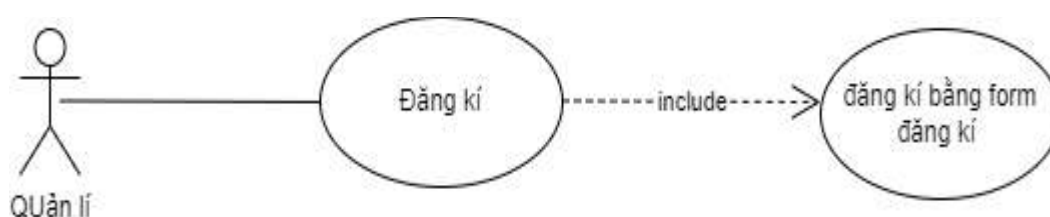


Hình 2.3: Biểu đồ phân rã chức năng đăng nhập

**Bảng 2.1: Bảng đặc tả UseCase đăng nhập**

Tên UseCase	Đăng nhập
Mô tả	Cho phép người quản lý đăng nhập tài khoản.
Tác nhân	Người quản lý chưa đăng nhập.
Điều kiện đầu vào	Người quản lý đã có tài khoản và xác thực tài khoản thành công.
Điều kiện đầu ra	Đăng nhập thành công hoặc không thành công.
Luồng sự kiện chính	Phần mềm yêu cầu người quản lý nhập tên và mật khẩu. Người quản lý nhập đầy đủ thông tin tên và mật khẩu.
Luồng sự kiện phụ	Người quản lý nhấn nút “Đăng nhập”. Hệ thống kiểm tra thông tin tài khoản người dùng. Thông báo người quản lý đã đăng nhập thành công.

### 2.3.4. Biểu đồ phân rã với chức năng đăng kí

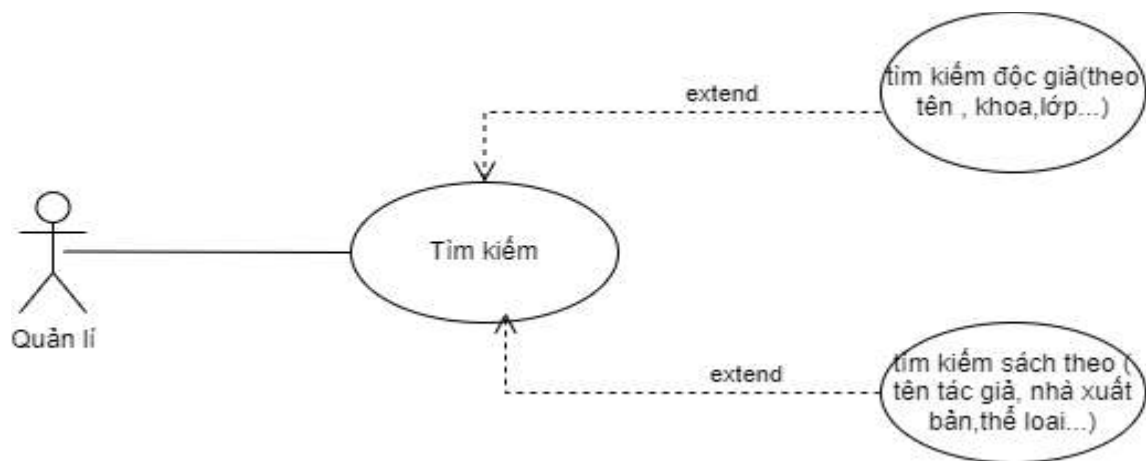


Hình 2.4: Biểu đồ phân rã chức năng đăng kí

**Bảng 2.2: Bảng đặc tả Use Case đăng kí**

Tên UseCase	Đăng kí
Mô tả	Cho phép người quản lí đăng kí tài khoản
Tác nhân	Người quản lí chưa đăng nhập
Điều kiện đầu vào	Người quản lí chưa có tài khoản chọn chức năng đăng kí
Điều kiện đầu ra	Người quản lí đăng kí thành công hoặc huỷ đăng kí
Luồng sự kiện chính	Từ giao diện, người quản lí nhấn nút “Đăng xuất”. Hệ thống hiển thị thông báo “Bạn có muốn đăng xuất khỏi hệ thống” và 2 nút chức năng: Yes, No.
Luồng sự kiện phụ	Người quản lí chọn No. Người quản lí quyết định huỷ bỏ việc đăng xuất. Tiếp tục sử dụng các use case khác.

### 2.3.5. Biểu đồ phân rã với chức năng tìm kiếm

**Hình 2.5: Biểu đồ phân rã chức năng tìm kiếm****Bảng 2.3: Bảng đặc tả UseCase tìm kiếm**

Tên UseCase	Tìm kiếm
Mô tả	Cho phép người quản lí tìm kiếm độc giả hoặc các đầu sách
Tác nhân	Cho người quản lí
Điều kiện đầu vào	Khi người quản lí dùng Keyword tìm kiếm trên khung tìm kiếm
Điều kiện đầu ra	Kết quả tìm kiếm là độc giả hoặc đầu sách

### Mô tả chi tiết quản lý độc giả và sách

*Mô tả usecase quản độc giả (tìm kiếm)*

Tên: Quản lí độc giả

Tác nhân: người quản lí

Mô tả: Khi người quản lý muốn thực hiện một thay đổi với độc giả (thêm, sửa, tìm kiếm).

Tiền điều kiện: người quản lý đã đăng nhập vào hệ thống.

Luồng sự kiện chính:

Sau khi đăng nhập thành công, quản lý chọn Quản lý độc giả.

Người quản lý chọn hành động muốn thực hiện đối với độc giả.

*Tìm kiếm độc giả*

- Người quản lý nhập thông tin cần tìm kiếm
- Người quản lý ấn nút Tìm kiếm

*Thay đổi thông tin độc giả .*

- Sau khi chọn *Sửa*, hệ thống chuyển sang giao diện thay đổi chi tiết
- Nhập các thông tin về độc giả.
- Chọn *Lưu*

*Mô tả usecase quản lý sách (tìm kiếm)*

Tên : quản lý sách.

Tác nhân: người quản lý

Mô tả: Khi người quản lý muốn tìm kiếm sách và xem thông tin của đơn sách đó.

Luồng sự kiện:

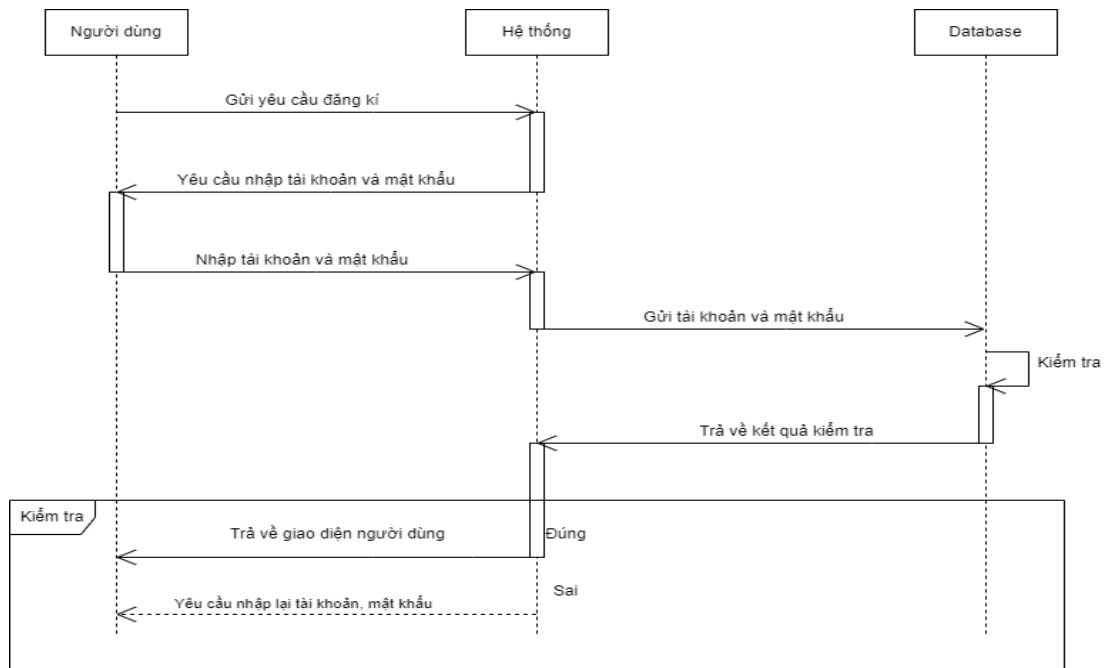
- Sau khi đăng nhập thành công, người quản lý chọn quản lý sách.
- Nhập thông tin sách cần tìm kiếm vào ô input
- Nhấn nút tìm kiếm.

Hậu điều kiện: hiển thị ra đơn hàng cần tìm hoặc thông báo không tìm thấy sách.



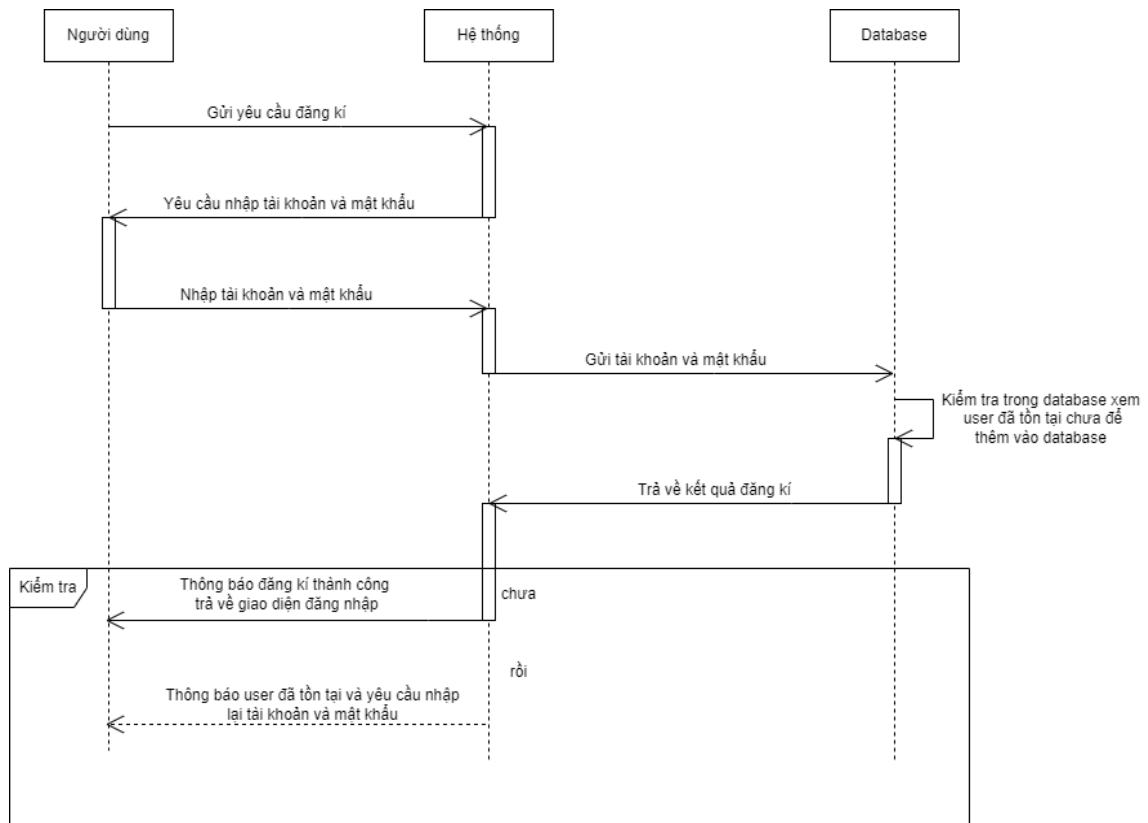
## 2.4. Sơ đồ tuần tự

### 2.4.1. Sơ đồ tuần tự chức năng đăng nhập



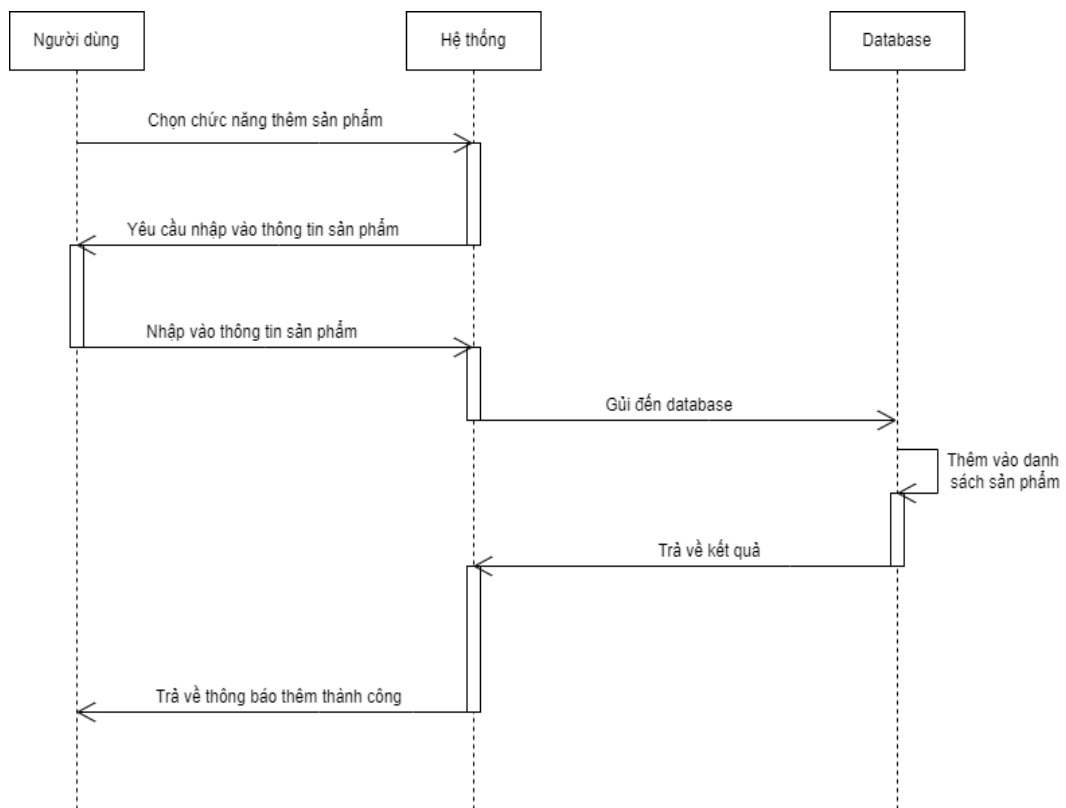
Hình 2.6: Biểu đồ tuần tự chức năng đăng nhập

### 2.4.2. Sơ đồ tuần tự chức năng đăng kí



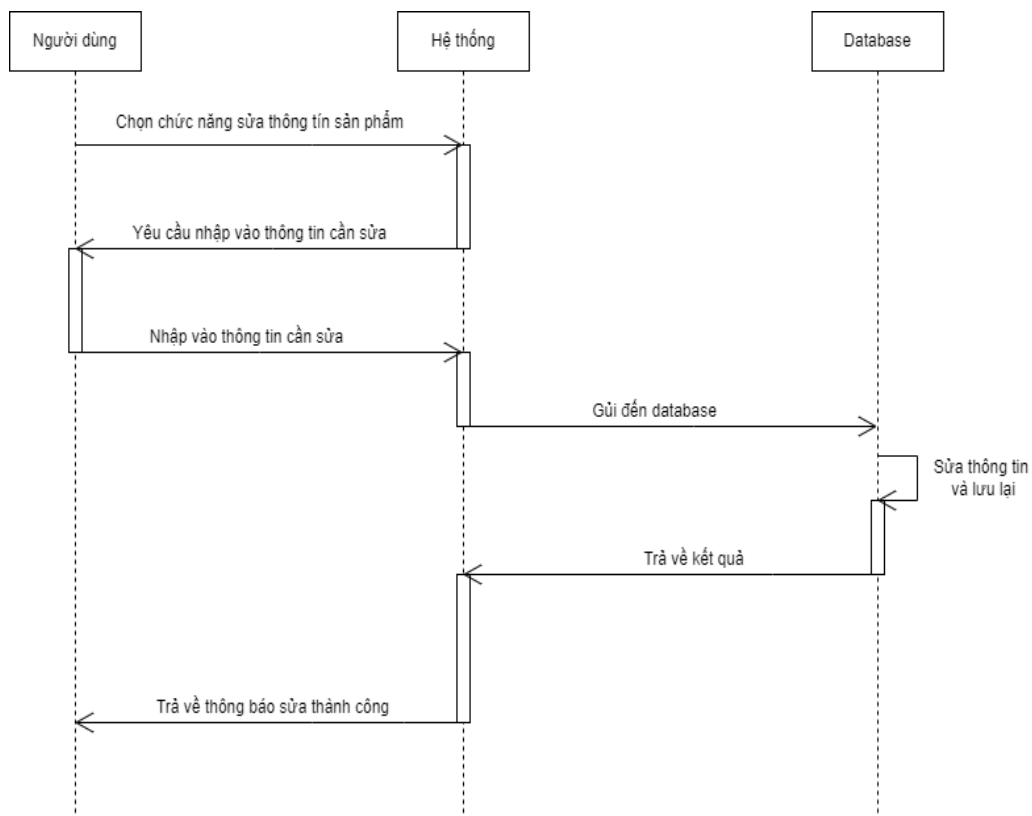
Hình 2.7: Biểu đồ tuần tự chức năng đăng kí

### 2.4.3. Sơ đồ tuần tự chức năng thêm sản phẩm(độc giả hoặc sách)



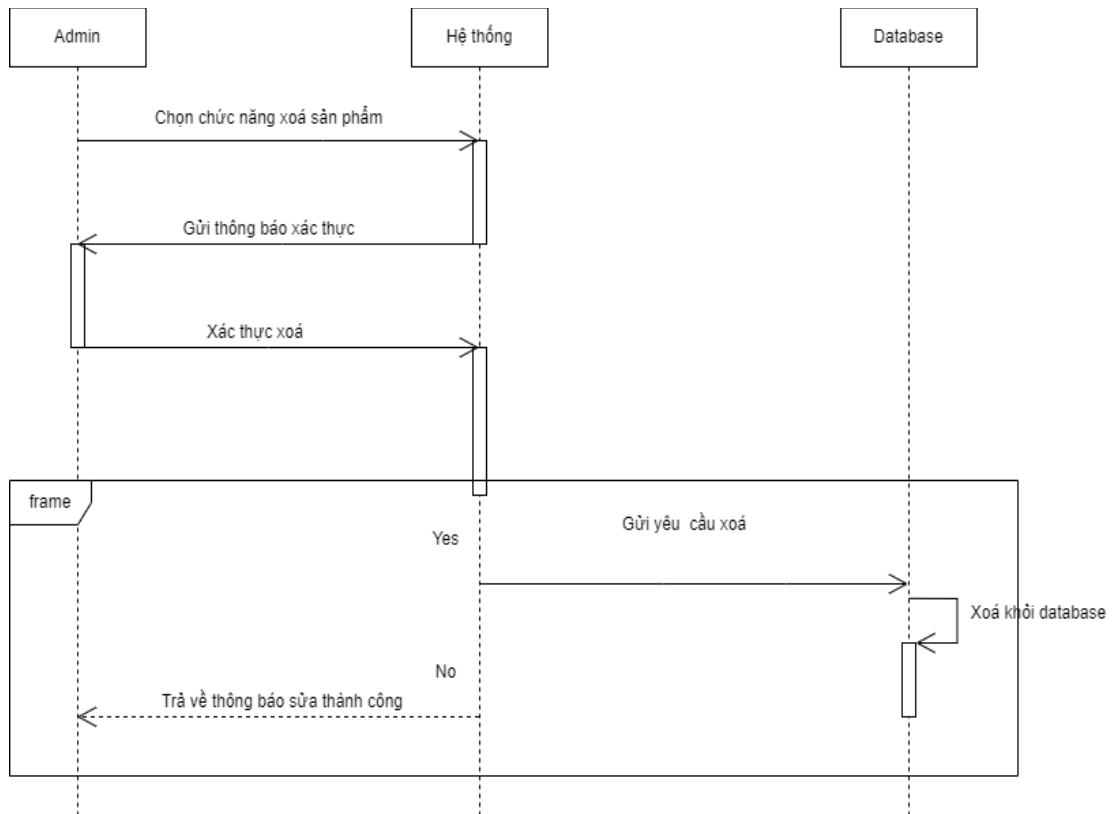
Hình 2.8: Biểu đồ tuần tự chức năng thêm sản phẩm(độc giả hoặc sách)

### 2.4.4. Sơ đồ tuần tự chức năng sửa thông tin sản phẩm (độc giả hoặc sách)



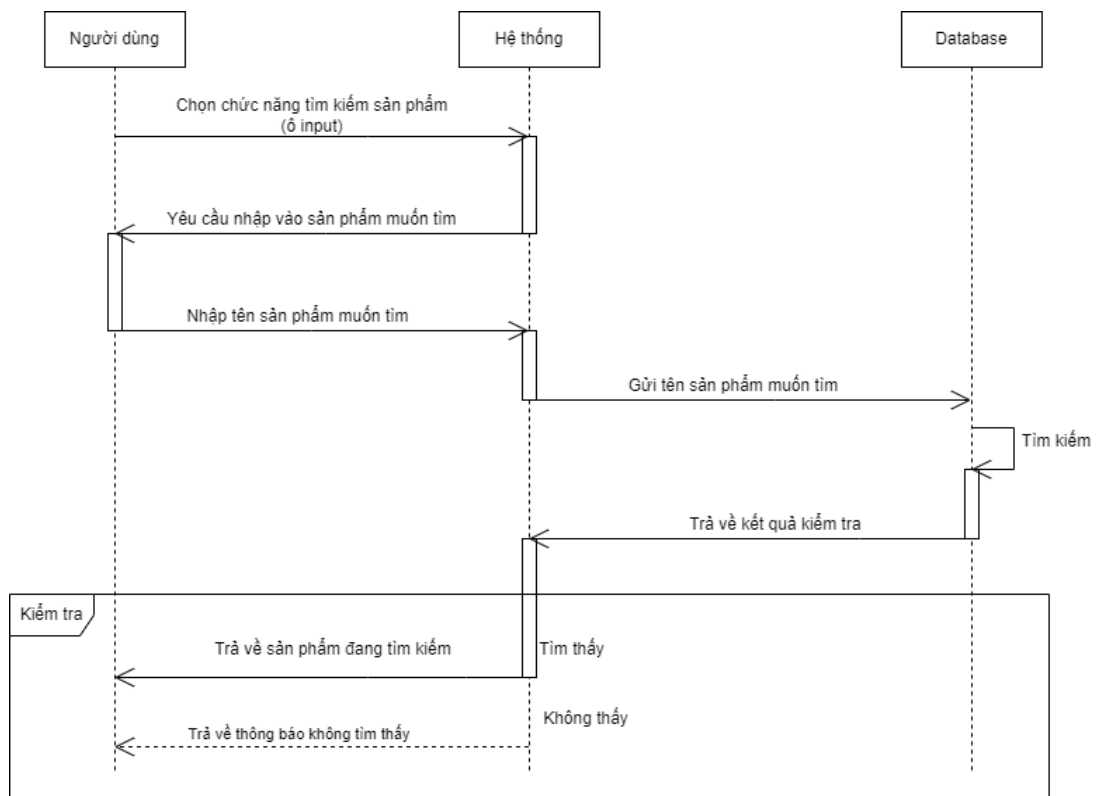
Hình 2.9: Biểu đồ tuần tự chức năng sửa thông tin sản phẩm(độc giả hoặc sách)

#### 2.4.5. Sơ đồ tuần tự chức năng xoá sản phẩm (độc giả hoặc sách)



Hình 2.10: Biểu đồ tuần tự chức năng xoá sản phẩm(độc giả hoặc sách)

#### 2.4.6. Sơ đồ tuần tự chức năng tìm kiếm sản phẩm (độc giả hoặc sách)

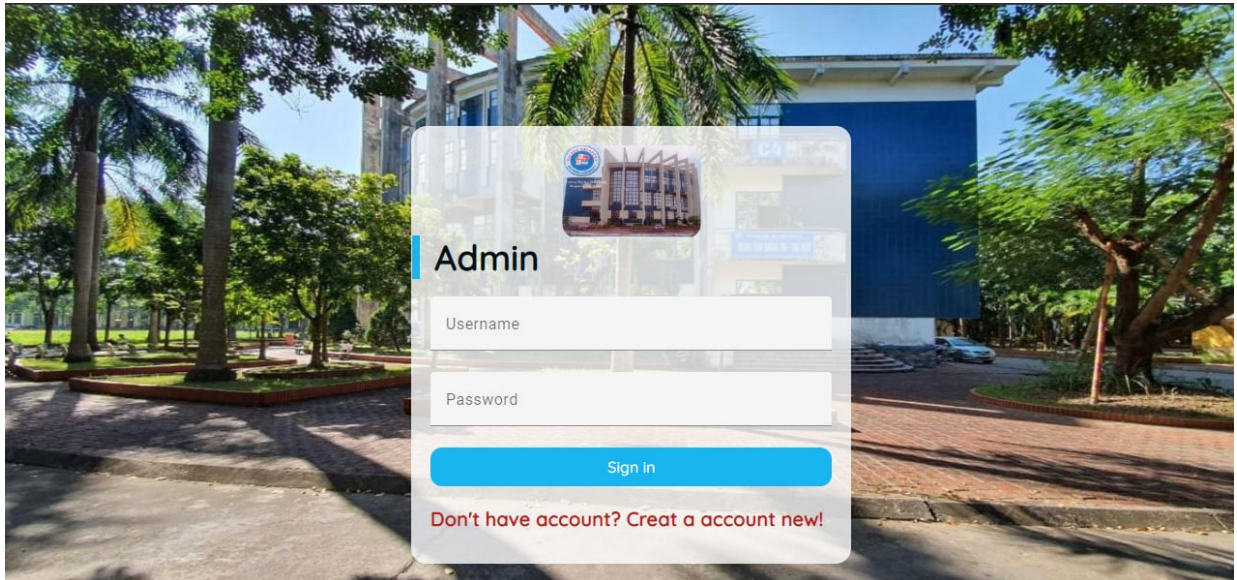


Hình 2.11: Biểu đồ tuần tự chức năng tìm kiếm sản phẩm(độc giả hoặc sách)

## CHƯƠNG III: THỰC NGHIỆM XÂY DỰNG HỆ THỐNG

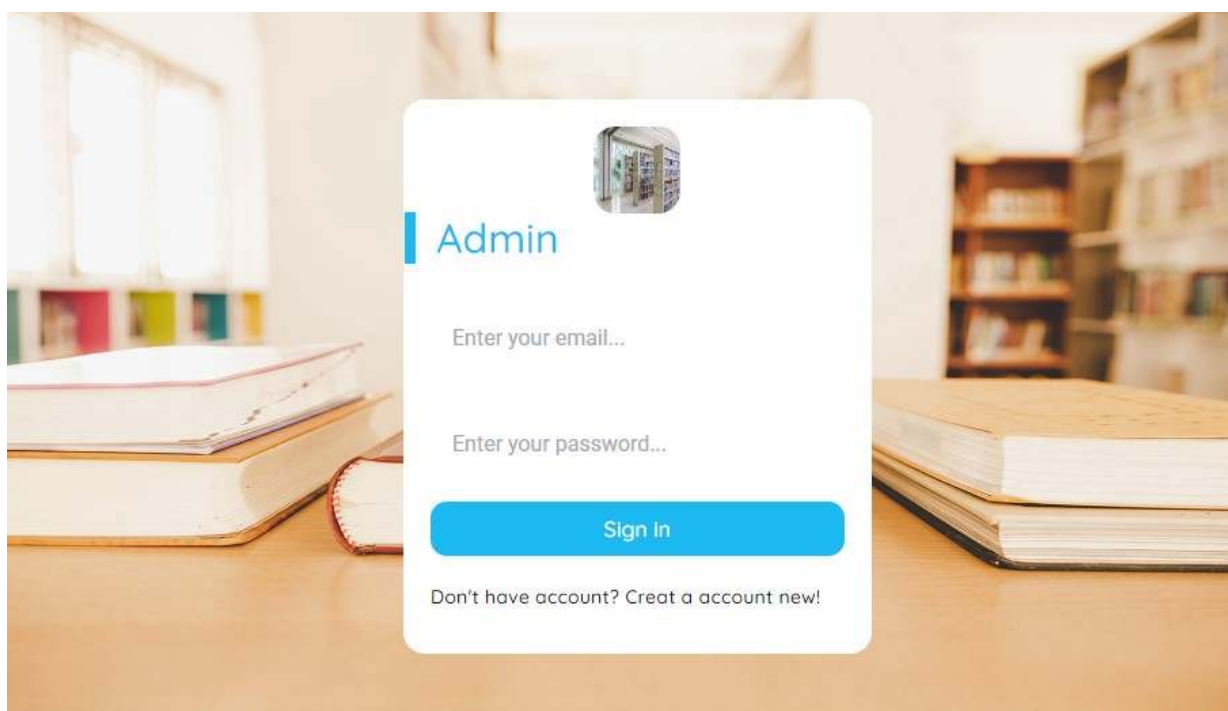
### 3. Một số giao diện của website

#### 3.1. Form đăng nhập



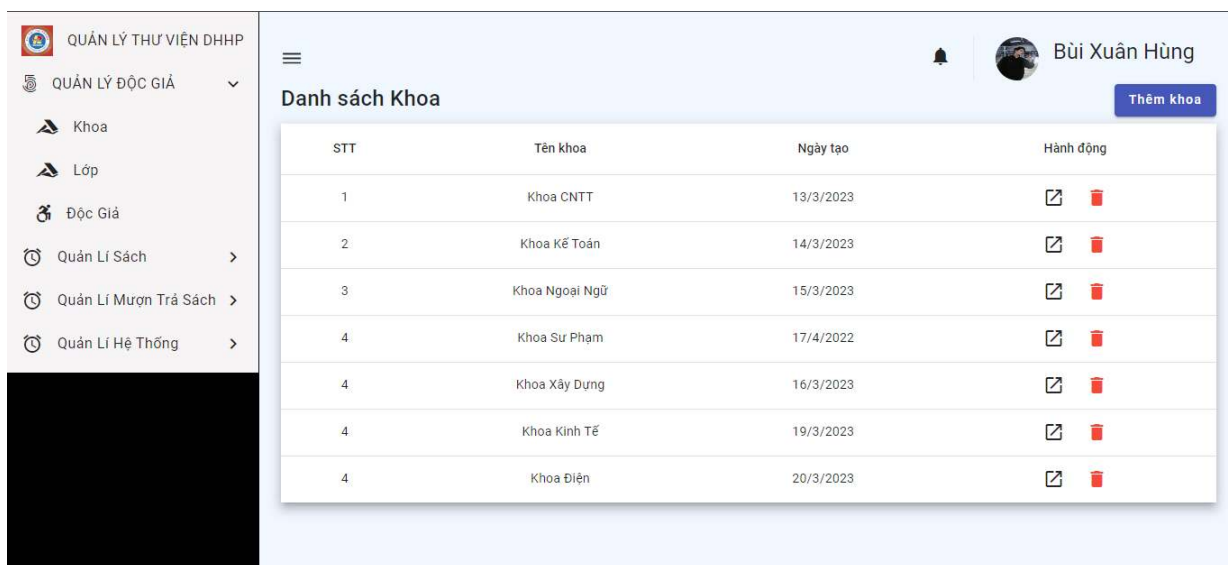
*Hình 3.1: Giao diện đăng nhập*

### 3.2. Form đăng kí



Hình 3.2: Giao diện đăng nhập

### 3.3. Form main



Hình 3.3: Giao diện quản lý độc giả (danh sách khoa)

QUẢN LÝ THƯ VIỆN DHHP

QUẢN LÝ ĐỘC GIẢ >

Quản Lý Sách >

Quản Lý Mượn Trả Sách >

Quản Lý Hệ Thống >

Thêm Khoa

Tên khoa

VD: Khoa CNTT, Khoa Toán, ...

Lưu Huỷ bỏ

Bùi Xuân Hùng

Hình 3.4: Giao diện khi thêm khoa mới

QUẢN LÝ THƯ VIỆN DHHP

QUẢN LÝ ĐỘC GIẢ >

Khoa

Lớp

Độc Giả

Quản Lý Sách >

Quản Lý Mượn Trả Sách >

Quản Lý Hệ Thống >

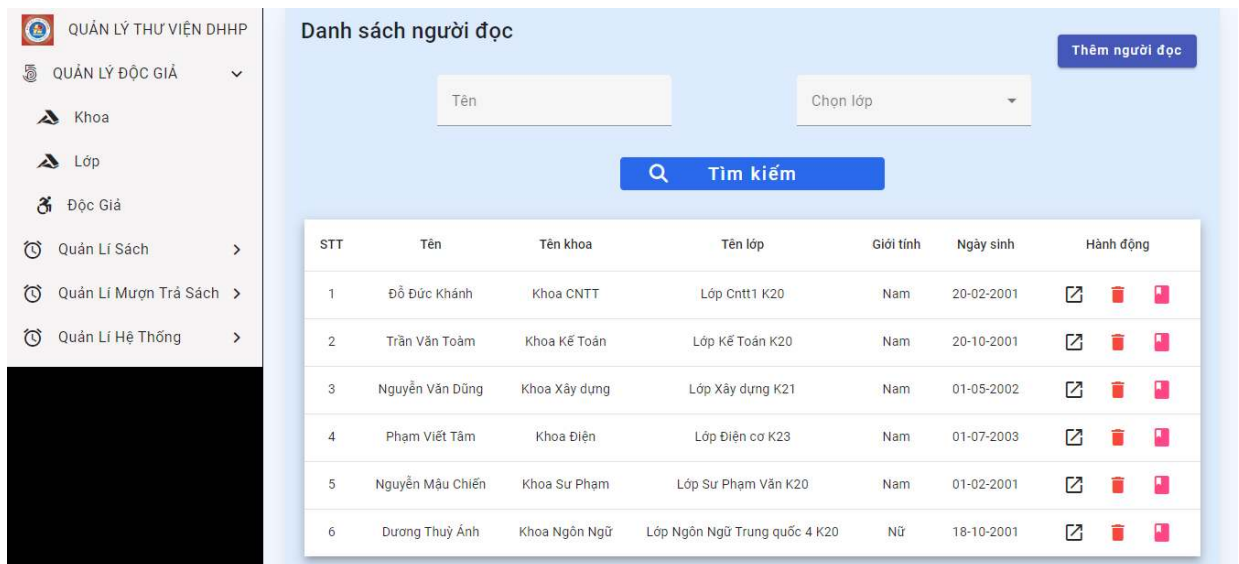
Danh sách Lớp

Thêm lớp

STT	Tên Lớp	Ngày tạo	Hành động
1	Lớp CNTT1 K20	12/3/2023	
2	Lớp Sư phạm Văn K20	12/3/2023	
3	Lớp Giáo dục tiểu học mầm non K21	14/3/2023	
5	Lớp Ngôn ngữ Anh 4	15/3/2023	
6	Lớp Ngôn ngữ Trung quốc 4	16/3/2023	
7	Lớp Lý luận chính trị	17/3/2023	
8	Lớp Kinh tế ngoại thương	18/3/2023	
9	Lớp Quản trị kinh doanh	19/3/2023	

Bùi Xuân Hùng

Hình 3.5: Giao diện quản lý độc giả (danh sách lớp)



Hình 3.6: Giao diện quản lý độc giả (danh sách người đọc)

**Thêm người đọc**

Tên\*: Nguyễn Thị Linh

Chọn giới tính: Nữ

Chọn ngày sinh: 2/14/2001

MM/DD/YYYY

Chọn Khoa: Khoa CNTT

Chọn lớp: Lớp Cntt1

Địa chỉ\*: Xã Nam Sơn - Huyện An Dương - Tp Hải Phòng

Số Điện Thoại\*: 0939226789

Choose File: anh-chan-dung.jpg

Hình 3.7: Giao diện khi thêm người đọc



QUẢN LÝ THƯ VIỆN DHHP

QUẢN LÝ ĐỘC GIẢ

Quản Lý Sách

Tác Giả

Phân Loại

Nhà Xuất Bản

Sách

Quản Lý Mượn Trả Sách

Quản Lý Hệ Thống

Bùi Xuân Hùng

Thêm tác giả

Danh sách Tác giả

STT	Tên	Nghề nghiệp	Địa chỉ	Giới tính	Năm sinh	Hành động
1	Nguyễn Nhật Ánh	Nhà văn,nhà thơ	Thăng Bình - Quảng Nam	Nam	07-05-1955	
2	Trang Hạ	Nhà văn	Hà Nội	Nữ	30-11-1975	
3	Nguyễn Phong Việt	Nhà thơ	Tuy Hoà-Phú Yên	Nam	không rõ	
4	Anh Khang	Nhà văn	Sài Gòn	Nam	11-08-1987	
6	Nguyễn Ngọc Thạch	Nhà văn	Sài Gòn	Nam	1987	

Hình 3.8: Giao diện quản lý sách (danh sách tác giả)

QUẢN LÝ THƯ VIỆN DHHP

QUẢN LÝ ĐỘC GIẢ

Quản Lý Sách

Quản Lý Mượn Trả Sách

Quản Lý Hệ Thống

Bùi Xuân Hùng

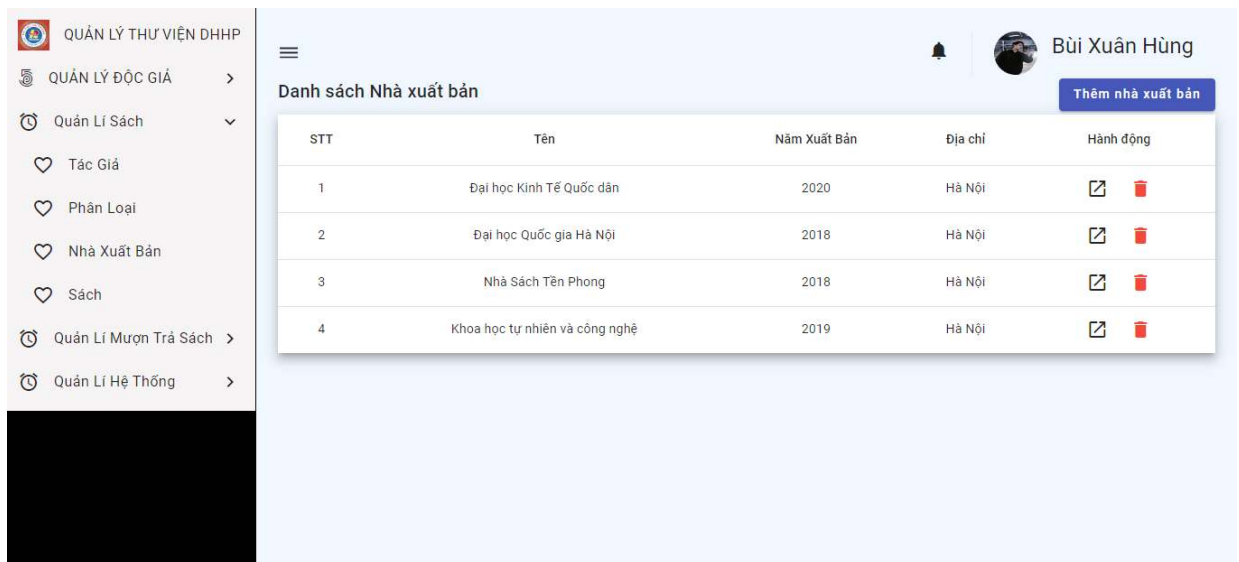
Thêm mới

Danh sách Phân loại

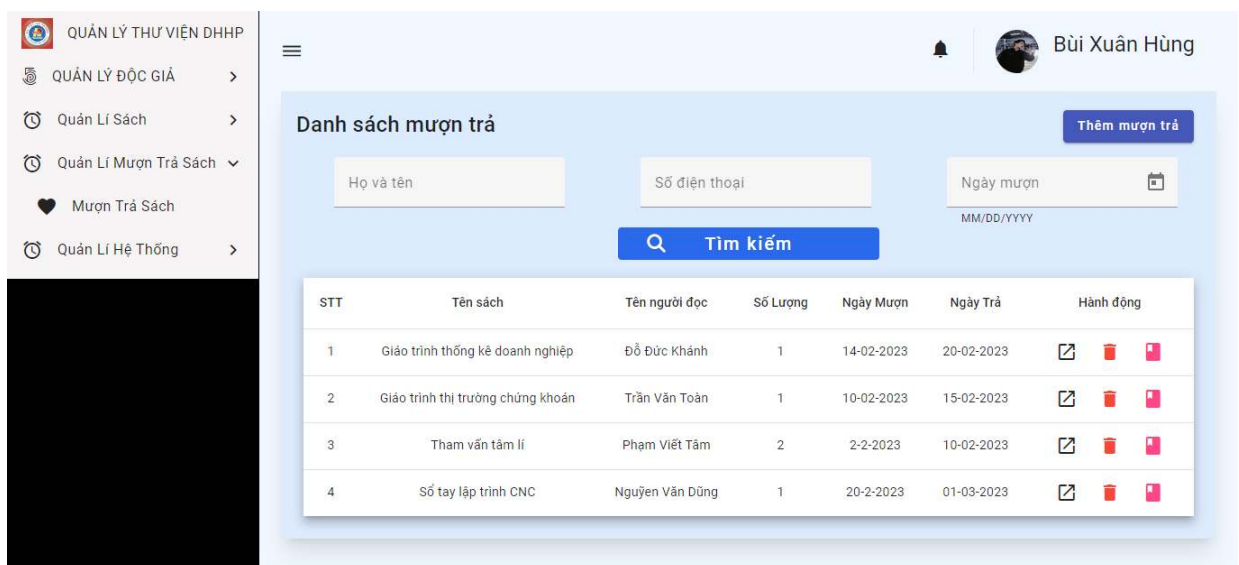
STT	Thể loại	Ngày tạo	Hành động
1	Truyện tranh	01-02-2023	
2	Sách tham khảo	01-03-2022	
3	Giáo trình	01-04-2023	
4	Từ điển	01-05-021	
5	Tài liệu	06-04-2022	
6	Trình thám	11-07-2020	
7	Bách khoa toàn thư	11-05-2019	

Hình 3.9: Giao diện quản lý sách (danh sách phân loại)





Hình 3.10: Giao diện quản lý sách (danh sách nhà xuất bản)



Hình 3.11: Giao diện quản lý mượn trả sách

## KẾT LUẬN

### \* Kết quả đạt được:

Sau thời gian tìm hiểu nghiệp vụ quản lý và thu thập các hồ sơ dữ liệu liên quan đến đề tài, em đã tương đối hoàn chỉnh những mục tiêu đặt ra cả đề tài “ Xây dựng quản lý thư viện”. Đề tài đã đưa ra một cách tổng quan việc xây dựng Website quản lý thư viện phục vụ cho việc quản lý một cách dễ dàng và công nghệ hoá hơn.

Trong đề tài này, vận dụng các kiến thức căn bản đã tìm hiểu về công nghệ ASP.NET CORE, Angular, PostgreSQL cùng với các nghiệp vụ cơ bản trong quản lý thư viện, em đã đạt được những kết quả:

- Hệ thống cung cấp giao diện trực quan, thân thiện, mang tính hiện đại và tiện dụng cho người dùng.
- Đăng nhập cho người quản lý.
- Hiện thị theo mục.
- Hoàn tất chức năng thêm, sửa, xoá , tìm kiếm độc giả và đầu sách.

### \* Kết quả chưa đạt được:

- Về chức năng: Hiện tại chỉ gồm các chức năng cơ bản của quản lý thư viện chưa có các chức năng nâng cao như là: thông báo, xử lý vi phạm, chưa phát triển được chức năng phân quyền.

- Về giao diện: Tuy thân thiện với người sử dụng nhưng kéo theo đó là hệ thống phải dành một phần hiệu năng tương đối mỗi lần tải giao diện.

- Về hiệu năng: Hiệu năng tuy phù hợp ở thời điểm hiện tại nhưng trong tương lai, quy mô dự án thay đổi chắc chắn hiệu năng sẽ không đủ đáp ứng được số lượng người truy cập lớn.

- Về bảo mật: Hệ thống mới chỉ phát triển mức độ bảo mật cơ bản của ASP.NET Core.

Mặc dù rất cố gắng, nhưng do kiến thức còn hạn hẹp nên đề tài không thể tránh khỏi thiếu sót và hạn chế. Em rất mong có được những đóng góp của thầy cô và các bạn để bài báo cáo đồ án của em được hoàn thiện và đầy đủ hơn.

### \* Định hướng phát triển:

Trong thời gian tới, em sẽ cố gắng khắc phục những hạn chế trên để tiếp tục hoàn thiện đề tài mà mở rộng một số tính năng. Định hướng phát triển trong thời gian tới là bổ sung thêm các chức năng:

- Hoàn thiện tính năng phân quyền.
- Cho phép độc giả có thể mượn cùng lúc nhiều cuốn sách.
- Việc cải thiện hiệu năng cũng vô cùng quan trọng trong tương lai khi mà lượng người dùng và dữ liệu ngày một nhiều hơn. Giải pháp trong tương lai là tối ưu lại toàn bộ cấu trúc truy vấn, thiết kế lại toàn bộ cơ sở dữ liệu, giao diện loại bỏ bớt hiệu ứng không cần thiết...
- Thực hiện chiến dịch SEO(Tối ưu hóa công cụ tìm kiếm)

## TÀI LIỆU THAM KHẢO

- [1] a. S. P. G. Krasner, Journal of Object Oriented Programming 1, Michigan University 1988.
- [2] Lê Hoàng Dũng, Tổng quan về ASP.NET MVC, Nhà xuất bản Khoa học và kỹ thuật, 2005.
- [3] <http://msdn.microsoft.com/>
- [4] <http://dot.net.vn/>
- [5] <http://asp.net.vn/>
- [6] <http://itprotraining.vn/>
- [7] <https://viblo.asia/>
- [8] Lưu Đan Thọ và Tôn Thất Hoàng Hải, Thương mại điện tử hiện đại, Nhà xuất bản Tài chính, 2015.
- [9] Jon Galloway, Brad Wilson K. Scott Allen, David Matson-Profesional ASP.NET

## NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Hải Phòng, ngày ... tháng ... năm 2023*

**Giảng viên hướng dẫn**