

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



BÀI TẬP LỚN KẾT THÚC HỌC PHẦN MÔN

ĐỒ ÁN CƠ SỞ

ĐỀ TÀI:

**“THIẾT KẾ WEBSITE QUẢN LÝ VÀ TRA CỨU NỘI DUNG CỦA NGHỊ ĐỊNH QUY
ĐỊNH XỬ PHẠT HÀNH CHÍNH TRONG LĨNH VỰC LÂM NGHIỆP”**

Nhóm số: 1

Giảng viên hướng dẫn : ThS.Phạm Văn Hà

Nhóm sinh viên thực hiện:

- | | |
|----------------------|----------------------|
| - Hoàng Hải Long | MSV: 20010909 |
| - Cao Thạch Đức Mạnh | MSV: 20010911 |
| - Nguyễn Việt Tuấn | MSV: 20010932 |
| - Bùi Khánh Huyền | MSV: 20010903 |

Lớp: Đồ án cơ sở-1-1-22(N02)

Source Code : <https://github.com/Longdevnotpro/dacs>

LỜI CẢM ƠN

Trong suốt quãng thời gian học tập và thực hiện Đồ án cơ sở, chúng em đã luôn nhận được sự quan tâm và hướng dẫn rất nhiệt tình của thầy cô trong khoa Công nghệ thông tin đã trang bị cho chúng em kiến thức cần thiết để hoàn thiện đề tài sáng tạo này.

Lời đầu tiên nhóm chúng em xin được bày tỏ lòng biết ơn chân thành đến Ban giám hiệu Trường Đại học Phenikaa đã giảng dạy những kiến thức, những kỹ năng cơ bản để hoàn thành đồ án.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, thầy Phạm Văn Hà, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Đề tài và bài báo cáo được chúng em thực hiện với những hạn chế về mặt kiến thức, kỹ thuật và cũng như kinh nghiệm thực tế. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của thầy để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau.

Chúng em xin chân thành cảm ơn.

Hà Nội, tháng 12 năm 2022

MỞ ĐẦU

Hiện nay, công nghệ dành cho thiết bị laptop, máy tính bảng, di động ngày càng phát triển. Nếu như trước đây, người dùng phải dùng chiếc máy tính để bàn cùng kèm cùng trình duyệt web IE (Internet Explorer) để lướt web, thì giờ đây với máy tính xách tay (laptop), điện thoại thông minh (smartphone) cùng rất nhiều trình duyệt khác (Firefox, Opera, Google Chrome...) người dùng có thể dễ dàng lướt “net” ở bất kì nơi đâu. Tuy nhiên với các thiết bị, trình duyệt khác nhau, nội dung sẽ hiển thị trên màn hình khác nhau. Chẳng hạn trên máy tính có thể xem trang web này rất tốt, nhưng điện thoại thông minh thì giao diện và cấu trúc trang bị xáo trộn. Hay có thể xem phim rất tốt với Google Chrome nhưng với opera, IE, Firefox thì không. Vậy giải pháp nào để người dùng có thể sử dụng bất kỳ trình duyệt trên bất kỳ thiết bị nào cũng có thể xem được đầy đủ, trọn vẹn nội dung, thông tin trên Internet. Đây cũng chính là lý do để nhóm chúng em tìm hiểu về HTML, CSS và xây dựng đề tài “Xây dựng website quản lý các điều luật”

CHƯƠNG 1: GIỚI THIỆU CHUNG

I. TỔNG QUAN VỀ CÔNG NGHỆ SỬ DỤNG

1.1 HTML

HTML là từ viết tắt của Hypertext Markup Language, là sự kết hợp của Hypertext và Markup, hay còn gọi là ngôn ngữ siêu văn bản. HTML có chức năng giúp người dùng xây dựng cấu trúc các phần trong trang web hoặc ứng dụng, thường được sử dụng trong phân chia các đoạn văn, heading, link, block quotes...

Những trang web này sau đó có thể được xem bởi bất kỳ ai khác kết nối với Internet. Nó tương đối dễ học, với những điều cơ bản có thể tiếp cận được với hầu hết mọi người trong một lần ngồi; và khá mạnh mẽ trong những gì nó cho phép bạn tạo ra. Nó liên tục trải qua sửa đổi và tiến hóa để đáp ứng nhu cầu và yêu cầu của khán giả Internet đang phát triển dưới sự chỉ đạo của W3C, tổ chức chịu trách nhiệm thiết kế và duy trì ngôn ngữ.

Ưu điểm của HTML

- HTML được ra đời từ rất lâu, do đó HTML có nguồn tài nguyên khổng lồ, hỗ trợ một cộng đồng người dùng lớn. Bên cạnh đó, cộng đồng HTML ngày càng phát triển trên thế giới.
- Mã nguồn của HTML là mã nguồn mở, do đó người dùng có thể sử dụng hoàn toàn miễn phí.
- HTML được sử dụng và được sử dụng trên nhiều trình duyệt được nhiều người dùng ưa chuộng hiện nay như Internet Explorer, Chrome, FireFox, Cốc cốc,...
- Học và tìm hiểu HTML đơn giản nên người học dễ dàng nắm được kiến thức và vận dụng trong xây dựng trang web của bạn.
- HTML được quy định theo một tiêu chuẩn nhất định nên việc markup sẽ trở nên gọn gàng, đồng nhất bởi HTML được vận hành bởi World Wide Web Consortium (W3C).

- HTML được thực hiện dễ dàng bởi HTML được tích hợp nhiều ngôn ngữ khác nhau như PHP, Java, NodeJS, Ruby,...Điều này sẽ giúp tạo thành một website hoàn chỉnh với nhiều tính năng.

Nhược điểm HTML

- Nhược điểm lớn nhất của HTML đó chính là chỉ có thể web tĩnh, web tĩnh có thể hiểu là những trang web chỉ hiện thông tin mà không có sự tương tác cho người dùng. Do đó, khi xây dựng tính năng động hoặc xây dựng hệ thống website có sự tương tác với người dùng, lập trình viên cần phải dùng thêm JavaScript hoặc ngôn ngữ backend của bên thứ ba.
- HTML thường chỉ có thể thực thi những thứ logic và cấu trúc nhất định, HTML không có khả năng tạo sự khác biệt và mới mẻ.
- Một số trình duyệt vẫn còn chậm trong việc hỗ trợ các phiên bản mới của HTML, đặc biệt là HTML5.
- Một số trình duyệt không thể render những tag mới trong HTML5.

1.2 CSS

CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,... thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc... CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì HTML không được thiết kế để gắn tag giúp định dạng trang web.

Ưu điểm CSS

- Giải quyết một vấn đề lớn: Nhờ CSS mà source code của trang Web sẽ được tổ chức gọn gàng hơn, trật tự hơn. Nội dung trang web sẽ được tách

bạch hơn trong việc định dạng hiển thị. Từ đó, quá trình cập nhật nội dung sẽ dễ dàng hơn và có thể hạn chế tối thiểu làm rối cho mã HTML.

- Tiết kiệm rất nhiều thời gian: sử dụng CSS giúp bạn không cần thực hiện lặp lại các mô tả cho từng thành phần. Từ đó, bạn có thể tiết kiệm được tối đa thời gian làm việc với nó, làm code ngắn lại giúp kiểm soát dễ dàng hơn các lỗi không đáng có.
- Cung cấp thêm các thuộc tính: CSS cung cấp các thuộc tính chi tiết hơn HTML để định nghĩa giao diện của trang web. CSS giúp người dùng nhiều styles trên một trang web HTML nên khả năng điều chỉnh trang của bạn trở nên vô hạn.

Nhược điểm CSS

- CSS hoạt động khác biệt cho từng trình duyệt, vì vậy khi thay đổi đã được thực hiện đòi hỏi chúng ta phải xác nhận được tính tương thích khi CSS hiển thị hiệu ứng thay đổi tương tự cho từng trình duyệt.
- Khá khó khăn cho người mới: với nhiều cấp độ của CSS thì việc tìm hiểu, có thể nắm bắt để sử dụng càng trở nên khó khăn hơn
- Định dạng của web có khả năng gặp rủi ro: có thể chịu tổn thương, gặp gián đoạn khi có hành động, hoặc tai nạn xảy ra với tệp. Lúc này nó sẽ yêu cầu truy cập đọc hoặc ghi vào web dự định để có thể ghi đè lên được các thay đổi.

1.3 Javascript

JavaScript là ngôn ngữ lập trình hướng đối tượng được tạo ra với mục đích xây dựng website. Đây được phân loại là ngôn ngữ cao cấp mang tính kịch bản. JavaScript giúp các website tương thích với người dùng. Nếu như trước đây với HTML và CSS, các website chỉ có thể hiển thị nội dung tĩnh thì giờ đây, với JavaScript, bạn có thể tương tác và truy xuất thông tin từ website.

Ưu điểm của JavaScript

- Chương trình JavaScript rất dễ học.

- Lỗi JavaScript dễ phát hiện và sẽ giúp bạn sửa lỗi nhanh hơn.
- Các trình duyệt web có thể dịch nó bằng HTML mà không cần một compiler.
- JS hoạt động trên rất nhiều nền tảng và trình duyệt khác nhau.
- Được đánh giá là ngôn ngữ lập trình nhẹ, nhanh so với các ngôn ngữ khác.
- JS có thể được gắn trên một số element hoặc các events của trang web.
- Khi website có sử dụng JS thì sẽ giúp cho trang web đó tương tác và tăng trải nghiệm người dùng khi truy cập.
- Bạn có thể tận dụng JavaScript để kiểm tra các input thay vì kiểm tra thủ công thông qua việc truy xuất database.

Mặt khác, ngoài những ưu điểm nhưng mọi ngôn ngữ lập trình khác đều có những nhược điểm riêng của nó như:

- Dễ bị khai thác từ những hacker và scammer.
- Có thể được dụng để thực thi mã độc trên máy tính của người dùng.
- JS code snippet lớn.
- Các thiết bị khác nhau có thể thực hiện JS khác nhau dẫn đến không đồng nhất.
- Vì tính bảo mật nên client-side JavaScript không cho phép đọc và ghi các file.
- JS không được hỗ trợ khi sử dụng trong kết nối mạng.
- JavaScript không có khả năng đa luồng hoặc đa xử lý.

1.4 NodeJs

- NodeJS là một nền tảng được tạo ra để phát triển các ứng dụng web có tốc độ thực thi và hiệu suất cao. Nó cho phép web developer triển khai ngôn ngữ javascript ở phía server, tạo nội dung động cho trang web.

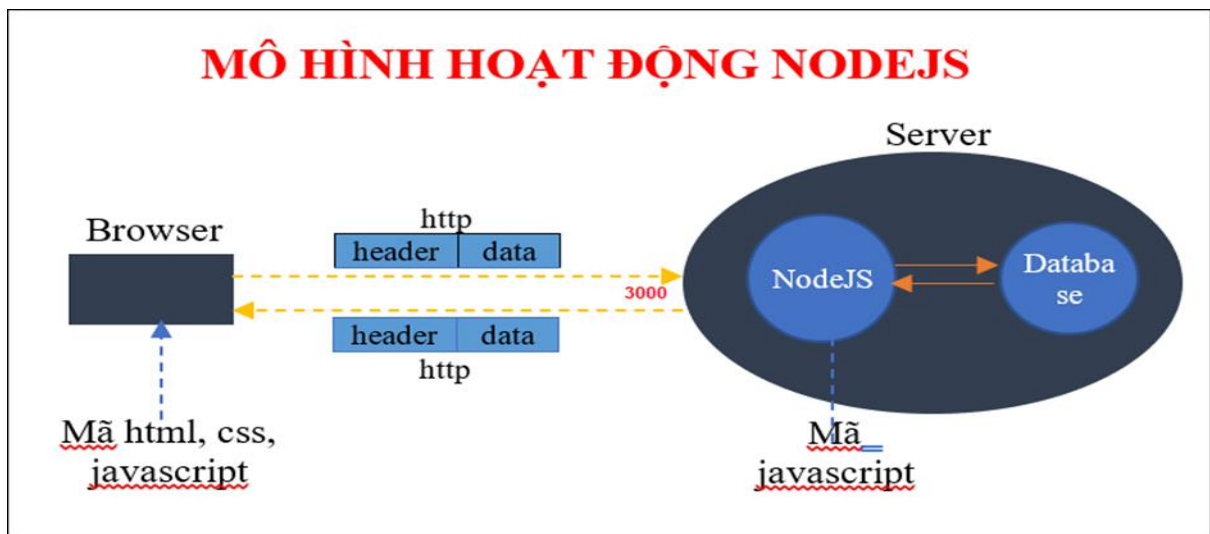
- Nodejs tạo ra bởi Ryan Dahl vào 2009, ban đầu được phát triển trên Linux, Mac OS. Bản chạy trên Windows phát hành vào 2011.

- NodeJS sử dụng kỹ thuật điều khiển theo sự kiện và bất đồng bộ giúp code chạy nhanh và hiệu quả.

- NodeJS được sử dụng nhiều trong các lĩnh vực I/O bound Applications, Data Streaming Applications, Data Intensive Real-time Applications (DIRT), JSON APIs based Applications, Single Page Applications.

Mô hình hoạt động của NodeJS

NodeJS hoạt động ở phía server, giúp tạo ra các trang web động. Viết bằng ngôn ngữ Javascript. Ưu điểm của NodeJS là hướng sự kiện, bất đồng bộ, chạy rất nhanh.



Ví dụ demo:

- Tạo file index.js sử dụng module http:

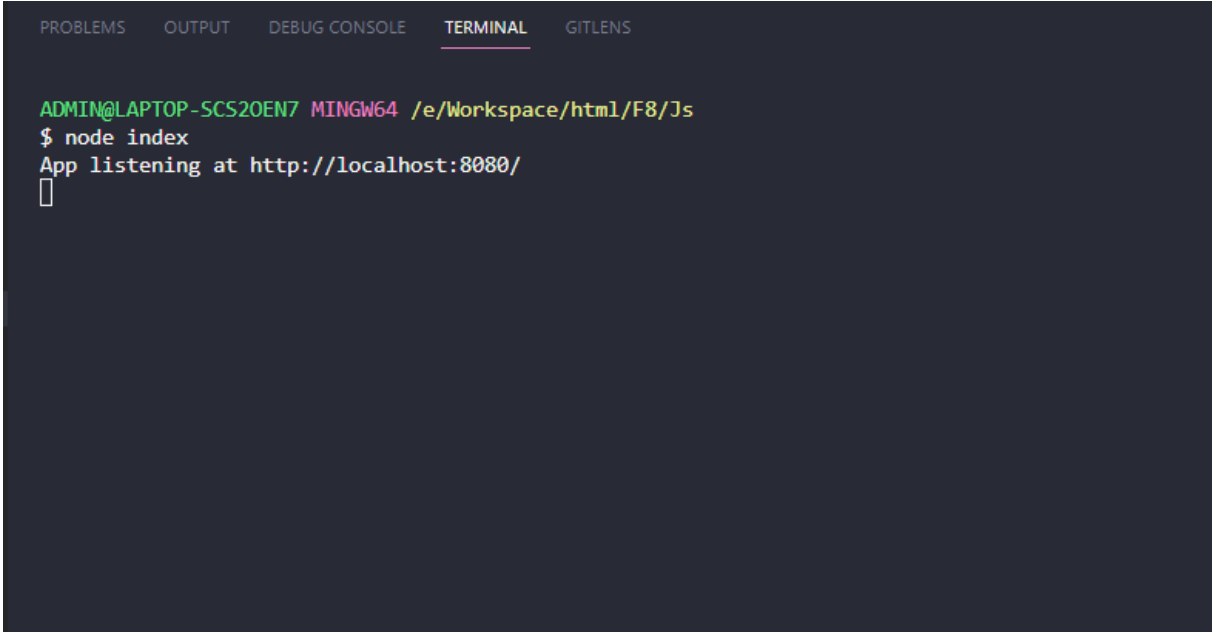
```
var http = require('http');
var port = 8080;

var app = http.createServer(function (req, res) {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.end('Hello World!');
});

app.listen(port, () => {
    console.log(`App listening at http://localhost:${port}/`);
});
```

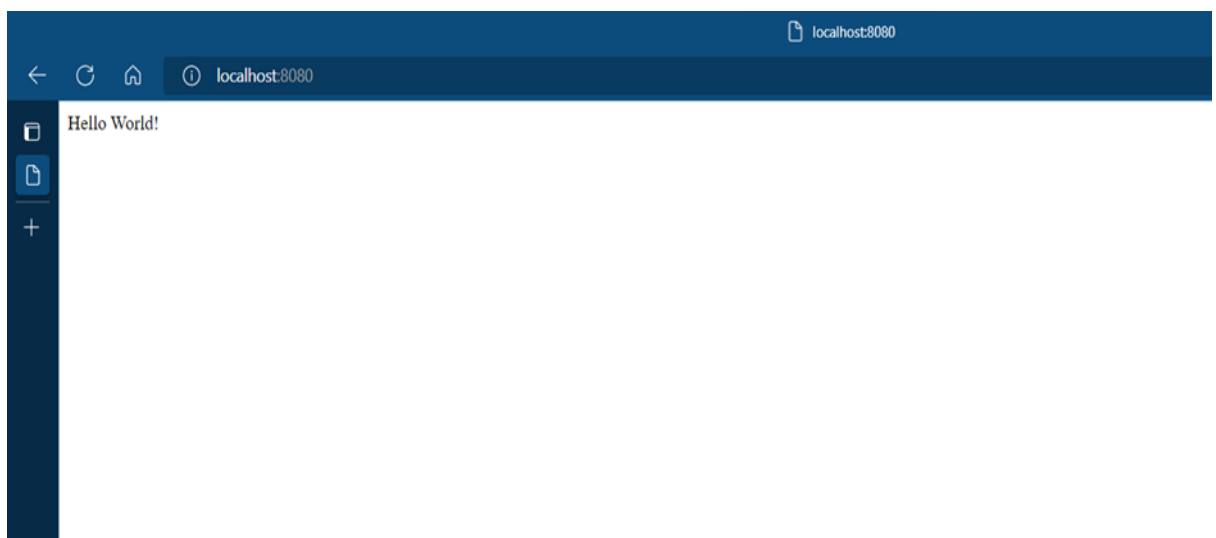
```
});
```

– Chạy server:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS  
  
ADMIN@LAPTOP-SCS20EN7 MINGW64 /e/Workspace/html/F8/Js  
$ node index  
App listening at http://localhost:8080/  
█
```

– Test: <http://localhost:8080>



1.5 MongoDB

Khái niệm:

- MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL(*) hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++.
- Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

Cách hoạt động của MongoDB:

- MongoDB hoạt động dưới một tiến trình ngầm service, luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.
- Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu “_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
- Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
- Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.

Ưu điểm của MongoDB:

- Ít schema hơn:
- Cấu trúc của một đối tượng rõ ràng.
- Không có các Join phức tạp.

- Khả năng mở rộng cực lớn: việc mở rộng dữ liệu mà không phải lo đến các vấn đề như khóa ngoại, khóa chính, kiểm tra ràng buộc, ...
- Sử dụng bộ nhớ trong để lưu giữ cửa sổ làm việc cho phép truy cập dữ liệu nhanh hơn. Việc cập nhật được thực hiện nhanh gọn nhờ update tại chỗ.

Nhược điểm của MongoDB:

- Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.
- Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

1.6 Phiên bản của công nghệ và cách cài đặt

-HTML5

-CSS

-Javascript ES6

-NODE JS Version 18.12.1

+Cách cài đặt: chúng ta sẽ truy cập vào trang web

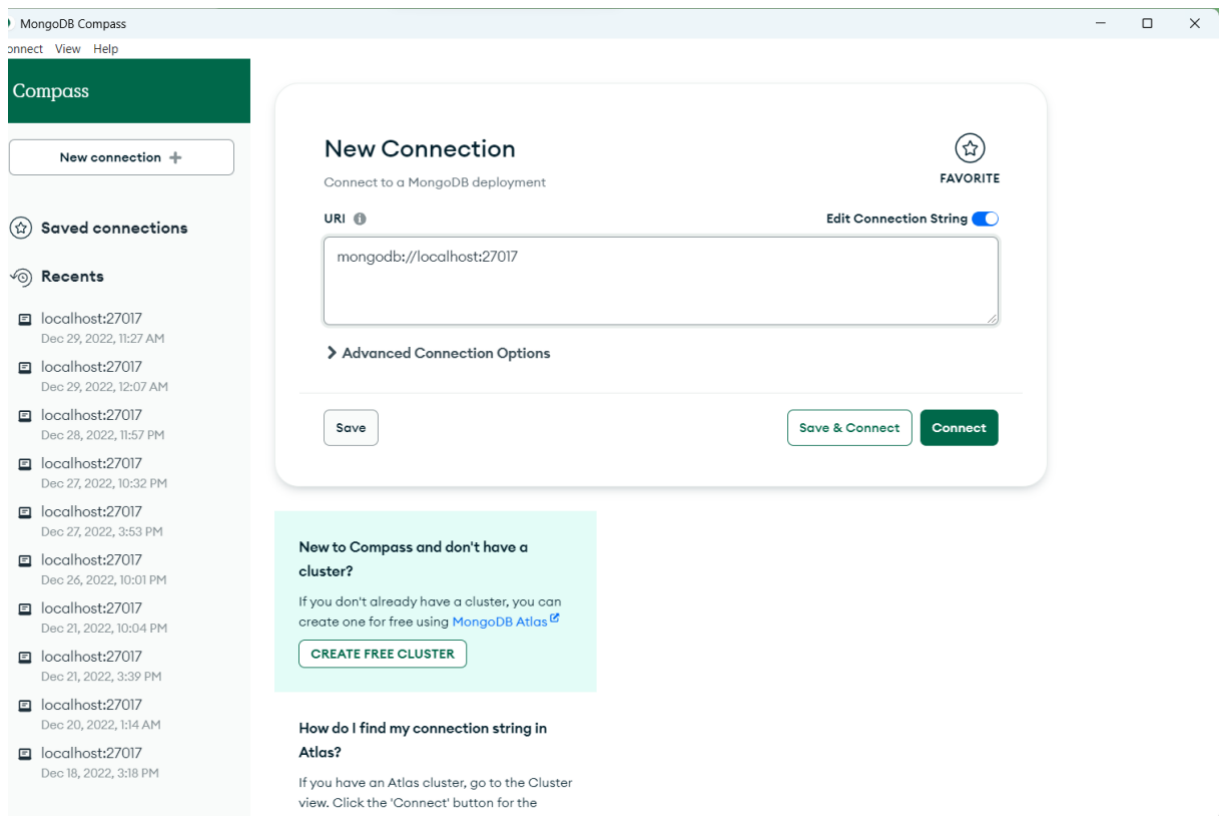
<https://nodejs.org/en/download/> và tải phiên bản phù hợp với máy của bạn

-Mongodb version 1.34.2

+Cách cài đặt: chúng ta truy cập dưới link sau để tải mongodb

<https://www.mongodb.com/try/download/compass>

Sau khi tải xong chúng ta giải nén và truy cập vào tên miền hoặc là localhost



Tiếp đến chúng ta connect và setup connect ở trong sourcecode của chúng ta

-Thư viện NODE js : expressjs version 4.18.2

+Cách cài đặt thư viện expressjs của nodejs: Do là thư viện của nodejs nên chúng ta sẽ sử dụng câu lệnh **npm install expressjs** trong terminal khi đó nó sẽ được thêm vào package.json thư viện express.js

-Thư viện Node js : express handlebars version: 6.0.6

+Cách cài đặt thư viện handlebars của nodejs: Tương tự express chúng ta sẽ sử dụng câu lệnh **npm install express handlebars** để thực thêm thư viện vào trong package.json

```

"dependencies": {
  "express": "^4.18.2",
  "express-handlebars": "^6.0.6",
  "express-session": "^1.17.3",
  "method-override": "^3.0.0",
  "mongoose-aggregate-paginate-v2": "^1.0.6",
  "popups": "^1.1.3"
},
"devDependencies": {
  "cookie-parser": "^1.4.6",
  "mongoose": "^6.8.1",
  "morgan": "^1.10.0",
  "node-sass": "^8.0.0",
  "nodemon": "^2.0.20"
}

```

-Mọi thư viện sẽ được lưu ở đây trong file package.json

II. ĐỀ TÀI

2.1 Đặt vấn đề

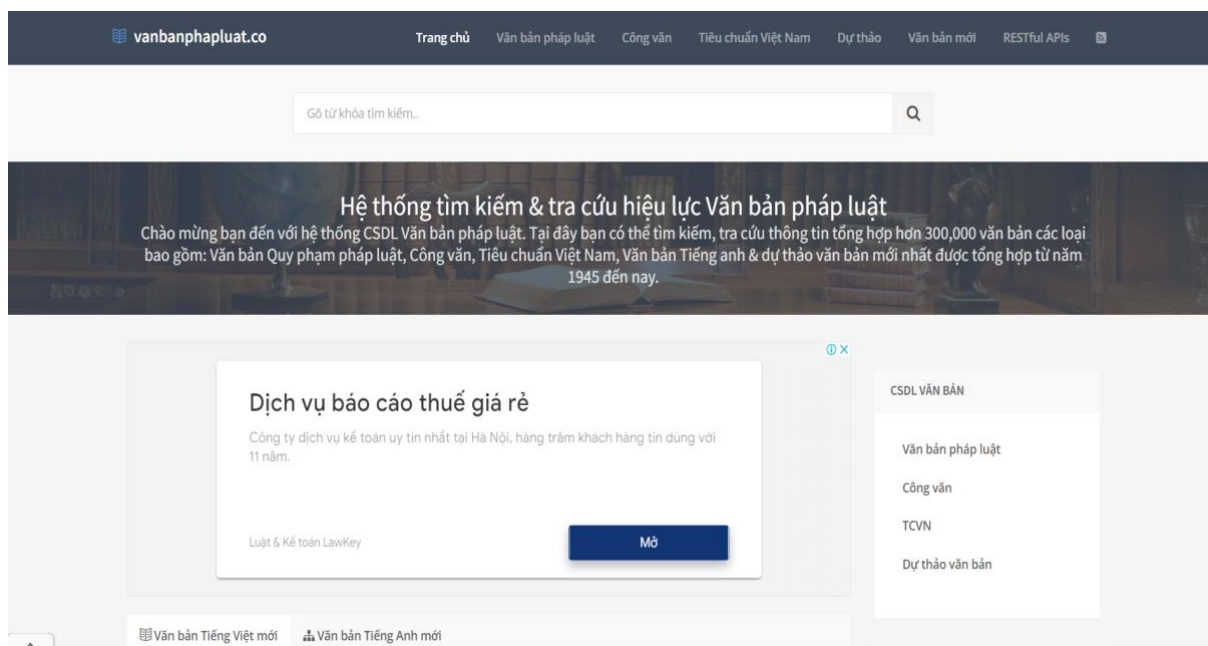
Theo báo cáo tổng kết công tác quản lý bảo vệ và phát triển rừng 6 tháng đầu năm năm 2022 của Chi cục Kiểm lâm cho thấy, tổng số vụ vi phạm Luật Lâm nghiệp phát hiện và xử lý là 3.696 vụ, trong đó: Xử lý hành chính 3.502 vụ, xử lý hình sự 194 vụ. Tổng số tiền phải thu nộp vào ngân sách theo quyết định xử phạt là 35,1 tỷ.

(<http://www.kiemlam.org.vn/Desktop.aspx/List/Tin-hoat-dong/Cuc-Kiem-lam-so-ket-cong-tac-6-thang-dau-nam-2022-va-trien-khai-nhiem-vu-6-thang-cuoi-nam/>)

□ Số liệu cụ thể trên đã cho thấy nhận thức của người dân còn kém, thiếu hiểu biết trong việc chấp hành các quy định của luật Lâm nghiệp và các văn bản dưới luật có liên quan.

Sau đây là một số website tra cứu luật mà chúng em đã tìm hiểu:

- Đây là 1 website có tên miền là vanbanphapluat.co/



Ưu điểm:

- Hình thức đơn giản, dễ nhìn, không gây rối mắt người tiếp cận
- Nội dung cần quan tâm luôn được để ở vị trí dễ thấy nhất.
- Cập nhật văn bản thường xuyên, đầy đủ từ Bộ Luật, Luật, Nghị định, Thông tư cho đến Công văn, Quyết định và Chỉ thị của các cơ quan ban ngành.
- Không cần tài khoản, đăng nhập: không cần cung cấp bất kỳ thông tin cá nhân nào.

Hạn chế:

- Chức năng tìm kiếm văn bản chưa được tối ưu: văn bản cần tìm kiếm không hiện lên ở những kết quả tìm kiếm đầu tiên.
- Các văn bản mới chưa được chứng thực và chi tiết.

- luatvietnam.vn/

Ưu điểm:

- Dễ sử dụng, dễ nhìn.
- Cung cấp tính năng sửa đổi bổ sung
- Có thể tải về văn bản mà không cần phải đăng nhập hay đăng kí tài khoản.

Hạn chế:

- Phải trả phí cho các tính năng quan trọng: nếu muốn xem thì đều phải đăng ký tài khoản trả phí dịch vụ với các mức phí khác nhau tùy theo nhu cầu.
- Tích hợp nhiều nội dung khác có liên quan đến pháp luật với nhiều người đó là một ưu điểm, nhưng cũng có thể là hạn chế gây mất tập trung nếu người dùng chỉ muốn tìm kiếm nội dung mà mình muốn tra cứu.

- thuvienphapluat.vn/



Ưu điểm:

- Cập nhật thường xuyên, đầy đủ.
- Cung cấp tính năng sửa đổi, bổ sung.
- Chức năng tìm kiếm trên trang dễ dàng sử dụng.

Hạn chế:

- Hình thức hơi rối mắt với nhiều banner, ảnh động khác nhau, cỡ chữ nhỏ, phối màu sắc gây khó chịu.
- Phân tán nhiều chủ đề: gây mất tập trung.
- Trả phí cho các tính năng như: hiệu lực văn bản, sửa đổi bổ sung.
- Tải văn bản, xem bản gốc cần đăng ký và đăng nhập.

2.2 Yêu cầu của đề tài

2.2.1 Yêu cầu chung

- Tìm hiểu các giải pháp xây dựng một website “Quản lý và tra cứu nội dung Nghị định” như thế nào.
- Xây dựng một trang web với đầy đủ tính năng phục vụ cho việc tra cứu nội dung Điều/Khoản.

- Xây dựng các chức năng giúp người quản trị dễ dàng quản lý nội dung trang web.

2.2.2 Xây dựng website "Quản lý và tra cứu nội dung Nghị định luật Lâm nghiệp"

- Giúp cho sinh viên hiểu biết về các phương pháp cũng như cách thức xây dựng một website tra cứu luật pháp Việt Nam quy mô nhỏ.

- Xây dựng Website mang lại thuận tiện cho người dùng trong việc tra cứu thông tin một cách trực quan bởi sự tiện lợi, nhanh chóng, hữu ích. Giúp người dùng trong một thời gian ngắn có thể tìm được toàn văn bản pháp luật bất kì, trích các điều, các khoản bất kì của nhiều văn bản khác nhau trong lĩnh vực Lâm Nghiệp

- Có một số chức năng cơ bản như sau:

- Đối với User: Xem thông tin và Tra cứu, Download văn bản, thông tin về Điều/Khoản của Nghị định.
- Đối với Admin: Quản trị nội dung trang web bao gồm các thao tác: Thêm, Sửa, Xóa, Tìm kiếm các nội dung liên quan đến Nghị định.

2.3 Đối tượng và phạm vi nghiên cứu

- Đối tượng:

- Cá nhân người sử dụng.
- Các cơ quan luật pháp muốn tra cứu thông tin về các Điều/Khoản của một Nghị định do Chính phủ ban hành.

III. PHÂN TÍCH VÀ THIẾT KẾ

3.1 Phân tích chức năng

Hệ thống bao gồm các chức năng chính:

STT	Chức năng	Mô tả
1	Đăng Nhập	Người Dùng tiến hành đăng nhập vào hệ thống.
2	Đăng Xuất	Nếu không còn nhu cầu sử dụng hệ thống hoặc cần đăng nhập bằng tài khoản khác có thể tiến hành đăng xuất

3	Thêm	Người Quản Trị tiến hành thao tác Thêm nội dung Điều/Khoản.
4	Sửa	Người Quản Trị tiến hành thao tác Sửa nội dung Điều/Khoản.
5	Xóa	Người Quản Trị tiến hành thao tác Xóa nội dung Điều/Khoản.
6	Tìm Kiếm/Xem thông tin	Đăng nhập bất kể dưới quyền nào: Người Quản Trị và Người Dùng đều có thể Tìm Kiếm/Xem thông tin nội dung Điều/Khoản.
7	Thiết lập profile	Sau khi đăng nhập Người dùng có thể tùy chỉnh các thông tin cá nhân của mình trong profile
8	Download văn bản luật	Người dùng có thể lưu điều luật mà mình cần

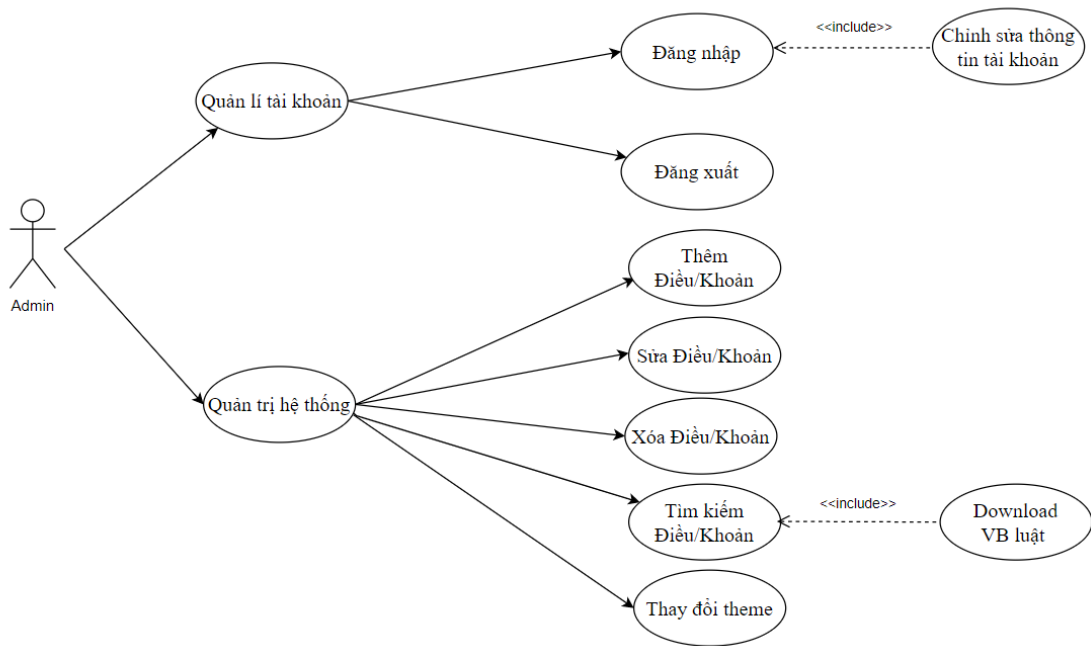
3.1.1 Xác định các Actor

Danh sách các Actor

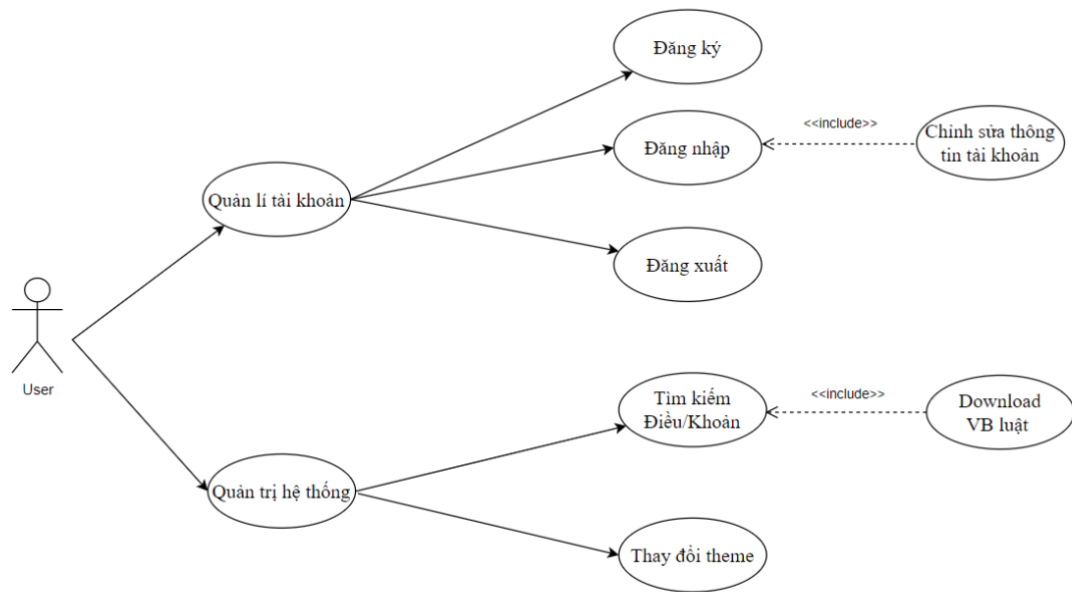
STT	Actor	Mô tả
1	Admin	Là người có toàn quyền tương tác với hệ thống, thực hiện các thao tác: <ul style="list-style-type: none"> - Thêm các Điều/Khoản. - Sửa các Điều/Khoản. - Xóa các Điều/Khoản.

		- Tìm kiếm các Điều/Khoản.
2	User	Là người có quyền tương tác với hệ thống, thực hiện các thao tác: - Xem thông tin luật. - Tìm kiếm thông tin luật. - Download văn bản

3.1.2 Biểu đồ Use-case



Admin



User

3.2 Đặc tả Usecase

3.2.1. Đặc tả Usecase Đăng nhập

Tên	Đăng Nhập
Mô tả	Cho phép người dùng đăng nhập vào hệ thống.

Tiền điều kiện	Người dùng chưa đăng nhập.
Hậu điều kiện	<ul style="list-style-type: none"> - Nếu đăng nhập thành công: người dùng được chuyển đến trang chủ phù hợp. - Nếu đăng nhập thất bại: người dùng được chuyển đến trang chủ của giao diện cho khách.
Luồng sự kiện chính	<ul style="list-style-type: none"> - Người dùng vào trang đăng nhập. - Người dùng nhập username, password. - Nhấn nút đăng nhập. - Hệ thống kiểm tra thông tin đăng nhập. - Nếu thông tin đăng nhập hợp lệ, hệ thống chuyển người dùng đến trang chủ phù hợp.
Luồng sự kiện phụ	<ol style="list-style-type: none"> 1. Đăng nhập thất bại <ul style="list-style-type: none"> - Hệ thống sẽ chuyển người dùng về lại trang đăng nhập - Hệ thống sẽ hiện thông báo lỗi "Thông tin đăng nhập không hợp lệ" - Trở lại bước 2 của luồng sự kiện chính 2. Người dùng thoát khỏi trang đăng nhập <ul style="list-style-type: none"> - Người dùng nhấn thoát khỏi giao diện đăng nhập - Hệ thống chuyển người dùng về lại trang chủ của giao diện khách.

3.2.2 Đặc tả Usecase Đăng ký

Tên	Đăng Kí
Mô tả	Cho phép người dùng đăng kí vào hệ thống.
Tiền điều kiện	Người dùng chưa có tài khoản

Hậu điều kiện	<ul style="list-style-type: none"> - Nếu đăng kí thành công: người dùng được chuyển đến trang chủ phù hợp. - Nếu đăng kí thất bại: người dùng được yêu cầu nhập lại thông tin.
Luồng sự kiện chính	<ul style="list-style-type: none"> - Người dùng vào trang chọn đăng kí. - Người dùng nhập các thông tin theo form mẫu - Nhấn “Đăng kí ngay”. - Hệ thống kiểm tra sự hợp lệ thông tin. - Nếu thông tin đăng kí trùng khớp với sự kiện đã lưu hệ thống sẽ hiện thông báo thành công và thêm tài khoản vào cơ sở dữ liệu. - Kết thúc use case.
Luồng sự kiện phụ	<p>1. Đăng kí thất bại</p> <ul style="list-style-type: none"> - Hệ thống sẽ chuyển người dùng về lại trang đăng kí - Hệ thống sẽ hiện thông báo lỗi "Thông tin đăng kí không hợp lệ" - Kết thúc use case. <p>2. Người đăng kí hủy bỏ việc đăng kí.</p> <ul style="list-style-type: none"> - Người dùng nhấn thoát khỏi giao diện đăng kí. - Hệ thống bỏ qua form đăng kí và trở về form chính. - Kết thúc use case.

3.2.3 Đặc tả Usecase Thêm Điều/Khoản

Tên	Thêm Điều/Khoản
Mô tả	Cho phép Người Quản trị thêm Điều/Khoản

Tiền điều kiện	Người Quản trị phải truy cập vào ứng dụng và đăng nhập thành công.
Hậu điều kiện	Người quản trị chọn chức năng Thêm.
Luồng sự kiện chính	<ul style="list-style-type: none"> - Người Quản trị chọn chức năng Thêm các điều luật - Hệ thống hiển thị trang Thêm. - Người Quản trị nhập các thông tin liên quan đến Điều/Khoản. - Người Quản Trị chọn Thêm. - Hệ thống kiểm tra sự hợp lệ của thông tin và thêm nội dung Điều/Khoản vào hệ thống. - Kết thúc use case.
Luồng sự kiện phụ	<p>Dòng thứ nhất:</p> <ul style="list-style-type: none"> - Người Quản Trị hủy bỏ việc Thêm các điều luật - Hệ thống sẽ xóa các điều khiển nhập liệu về giá trị mặc định. - Kết thúc use case. <p>Dòng thứ hai:</p> <ul style="list-style-type: none"> - Thông tin Người Quản Trị nhập vào không hợp lệ. - Hệ thống không tiến hành lưu thông tin vào database. - Kết thúc use case.

3.2.4 Đặc tả Usecase Sửa Điều/Khoản

Mô tả	Cho phép Người Quản Trị Sửa Điều/Khoản
Tiền điều kiện	Người Quản Trị phải truy cập vào hệ thống và đăng nhập thành công.

Hậu điều kiện	Người Quản Trị chọn chức năng Sửa và chọn nội dung cần sửa.
Luồng sự kiện chính	<ul style="list-style-type: none"> - Người Quản Trị chọn nội dung cần sửa và sửa lại thông tin cho đúng. - Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào. - Người dùng chọn Sửa. - Hệ thống xác nhận và cập nhật thông tin. - Kết thúc use case.
Luồng sự kiện phụ	<p>Dòng thứ nhất:</p> <ul style="list-style-type: none"> - Người Quản Trị hủy bỏ việc sửa nội dung Điều/Khoản. - Hệ thống sẽ xóa các điều khiển nhập liệu về giá trị mặc định. - Kết thúc use case. <p>Dòng thứ hai:</p> <ul style="list-style-type: none"> - Thông tin Người Quản Trị nhập vào không hợp lệ. - Hệ thống không tiến hành lưu thông tin vào database. - Kết thúc use case.

3.2.5 Đặc tả Usecase Xóa Điều/Khoản

Mô tả	Cho phép Người Quản Trị xóa Điều/Khoản.
Tiền điều kiện	Người Quản Trị phải truy cập vào ứng dụng và đăng nhập thành công.
Hậu điều kiện	Người Quản Trị chọn chức năng Xóa và chọn Điều/Khoản cần xóa.

Luồng sự kiện chính	<ul style="list-style-type: none"> - Người Quản Trị chọn nội dung cần xóa. - Người Quản Trị chọn xóa. - Hệ thống xác nhận và cập nhật thông tin. - Kết thúc use case.
Luồng sự kiện phụ	<p>Dòng thứ nhất:</p> <ul style="list-style-type: none"> - Người Quản Trị hủy bỏ việc xóa Điều/Khoản. - Hệ thống sẽ xóa các điều khiển nhập liệu về giá trị mặc định. - Kết thúc use case. <p>Dòng thứ hai:</p> <ul style="list-style-type: none"> - Thông tin Người Quản Trị nhập vào không hợp lệ. - Hệ thống không tiến hành lưu thông tin vào database. - Kết thúc use case.

3.2.6 Đặc tả Usecase Chỉnh sửa thông tin cá nhân

Mô tả	Cho phép Người Quản Trị xóa Điều/Khoản.
Tiền điều kiện	Người Quản Trị phải truy cập vào ứng dụng và đăng nhập thành công.
Hậu điều kiện	Người Quản Trị chọn chức năng Xóa và chọn Điều/Khoản cần xóa.
Luồng sự kiện chính	<ul style="list-style-type: none"> - Người Quản Trị chọn nội dung cần xóa. - Người Quản Trị chọn xóa. - Hệ thống xác nhận và cập nhật thông tin.

	- Kết thúc use case.
Luồng sự kiện phụ	<p>Dòng thứ nhất:</p> <ul style="list-style-type: none"> - Người Quản Trị hủy bỏ việc xóa Điều/Khoản. - Hệ thống sẽ xóa các điều khiển nhập liệu về giá trị mặc định. - Kết thúc use case. <p>Dòng thứ hai:</p> <ul style="list-style-type: none"> - Thông tin Người Quản Trị nhập vào không hợp lệ. - Hệ thống không tiến hành lưu thông tin vào database. - Kết thúc use case.

3.2.7 Đặc tả Usecase download văn bản luật

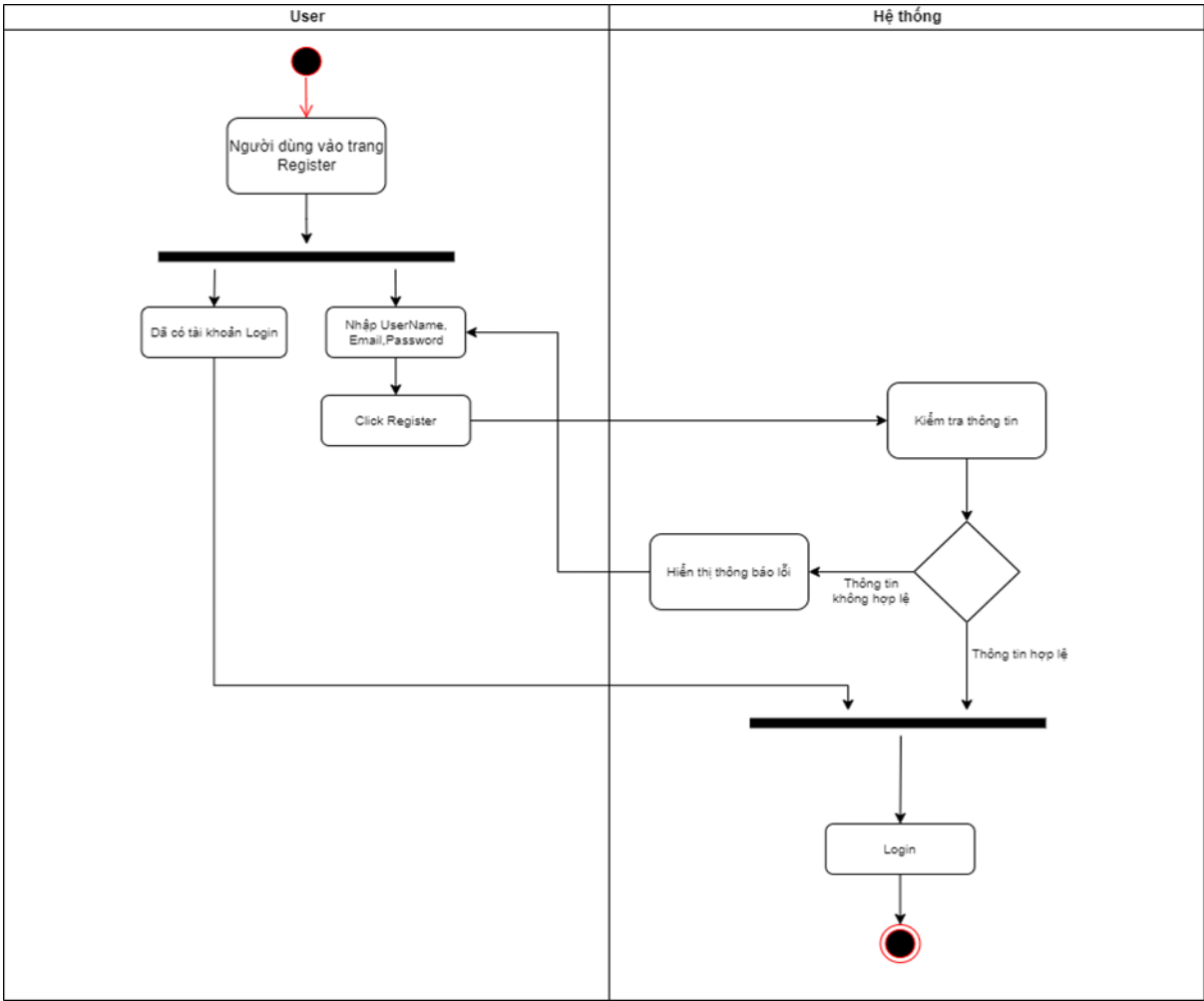
Mô tả	Cho phép Người dùng Chỉnh sửa thông tin cá nhân
Tiền điều kiện	Người dùng phải truy cập vào ứng dụng và đăng nhập thành công.
Hậu điều kiện	Người dùng chọn chức năng thiết lập profile.
Luồng sự kiện chính	<ul style="list-style-type: none"> - Người dùng chọn nội dung cần chỉnh sửa - Hệ thống xác nhận và cập nhật thông tin. - Kết thúc use case.

Luồng sự kiện phụ	<p>Dòng thứ nhất:</p> <ul style="list-style-type: none">- Người dùng hủy bỏ việc chỉnh sửa.- Hệ thống sẽ xóa các điều khiển nhập liệu về giá trị mặc định.- Kết thúc use case. <p>Dòng thứ hai:</p> <ul style="list-style-type: none">- Thông tin Người dùng nhập vào không hợp lệ.- Hệ thống không tiến hành lưu thông tin vào database.- Kết thúc use case.
-------------------	---

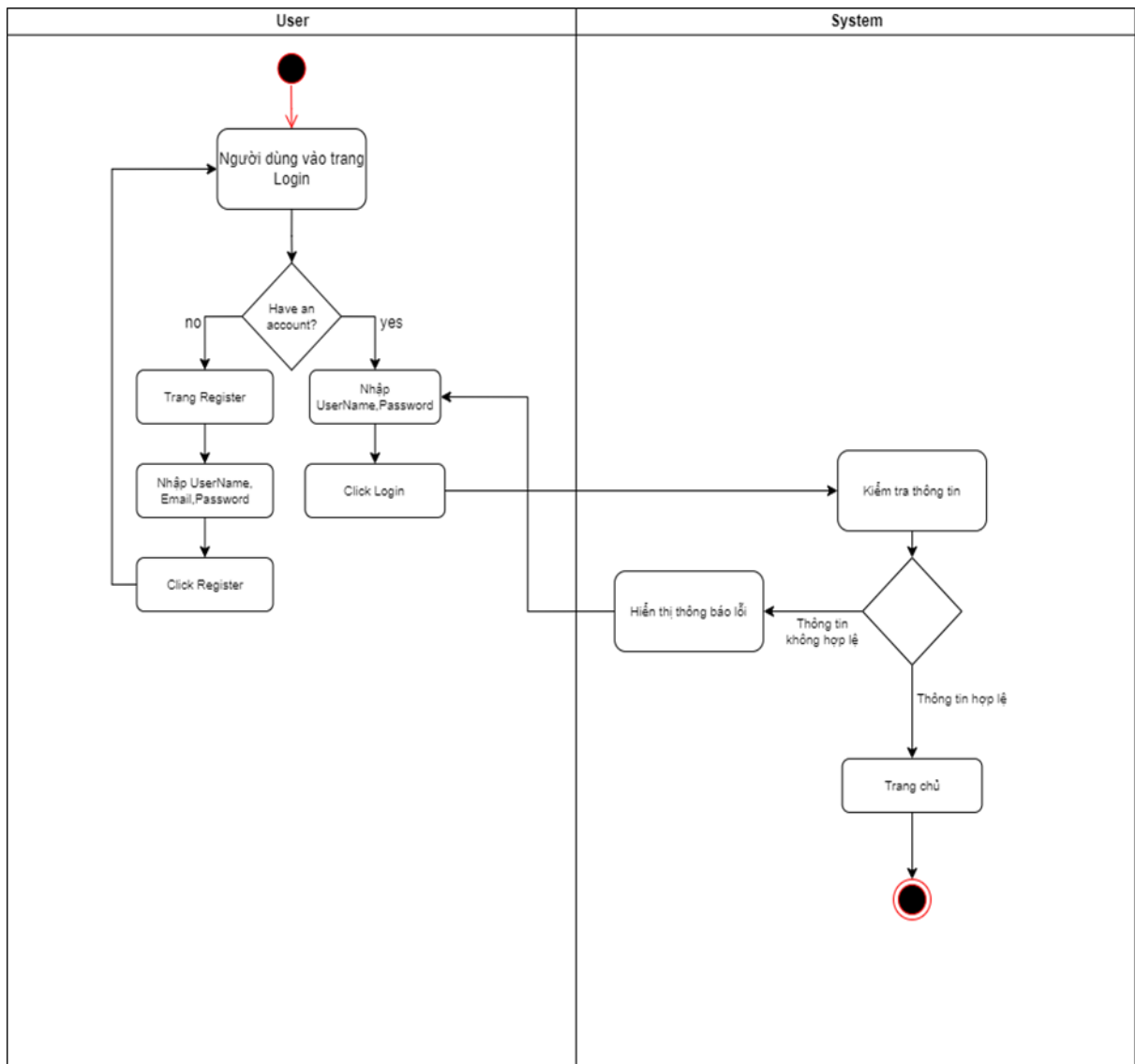
3.3 Activity Diagrams

3.3.1 User

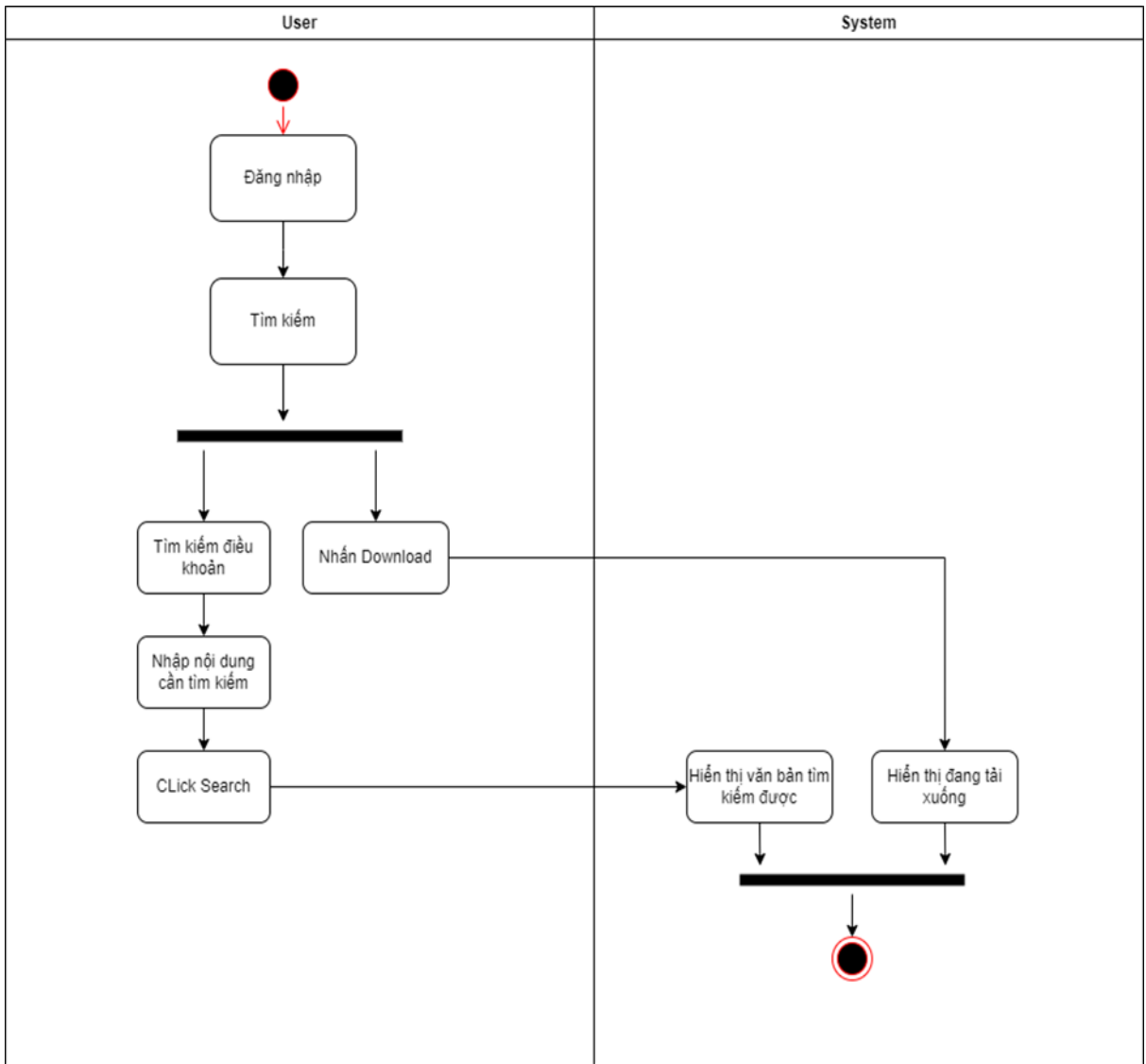
Register



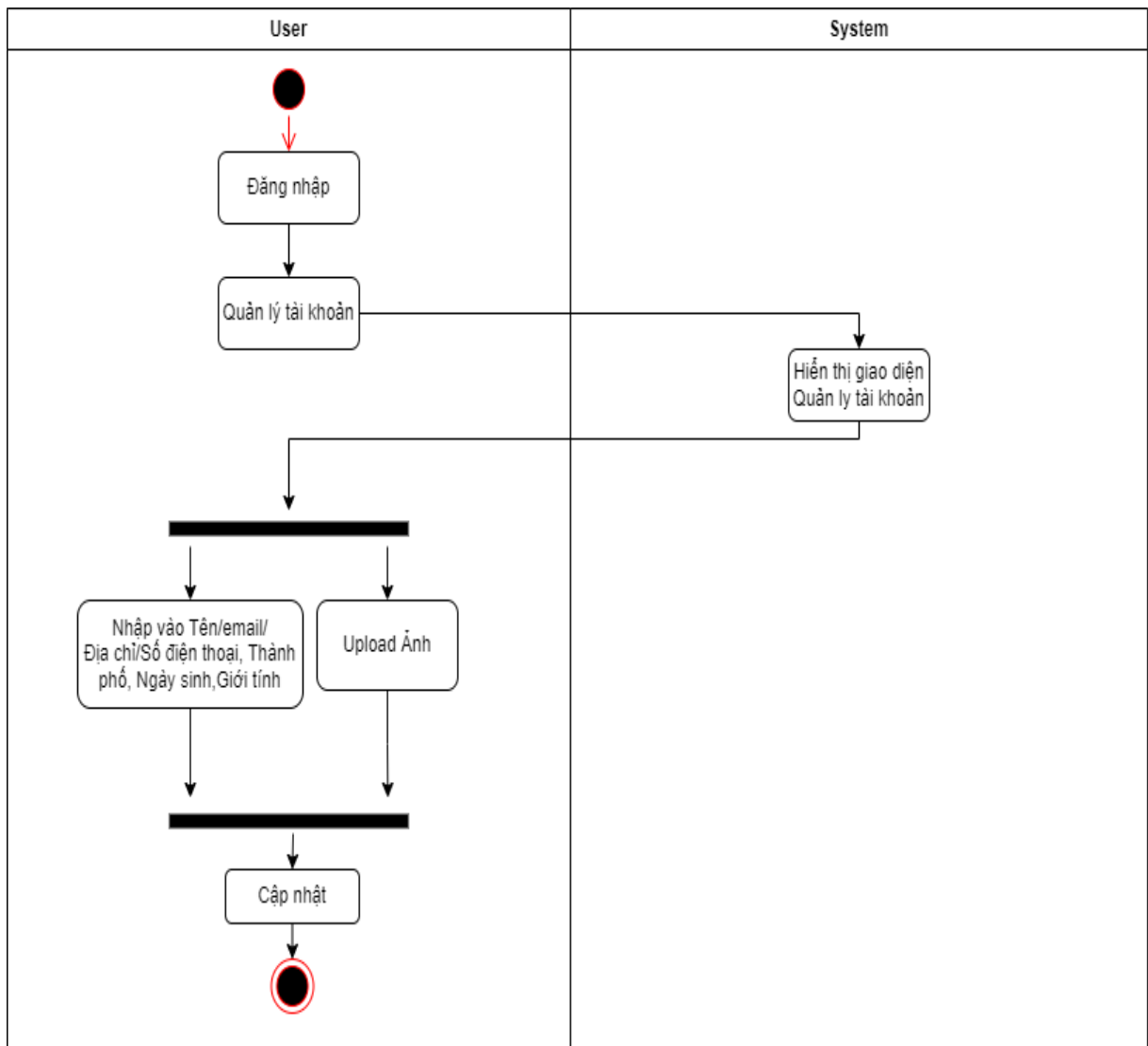
Login



Tìm kiếm điều khoản

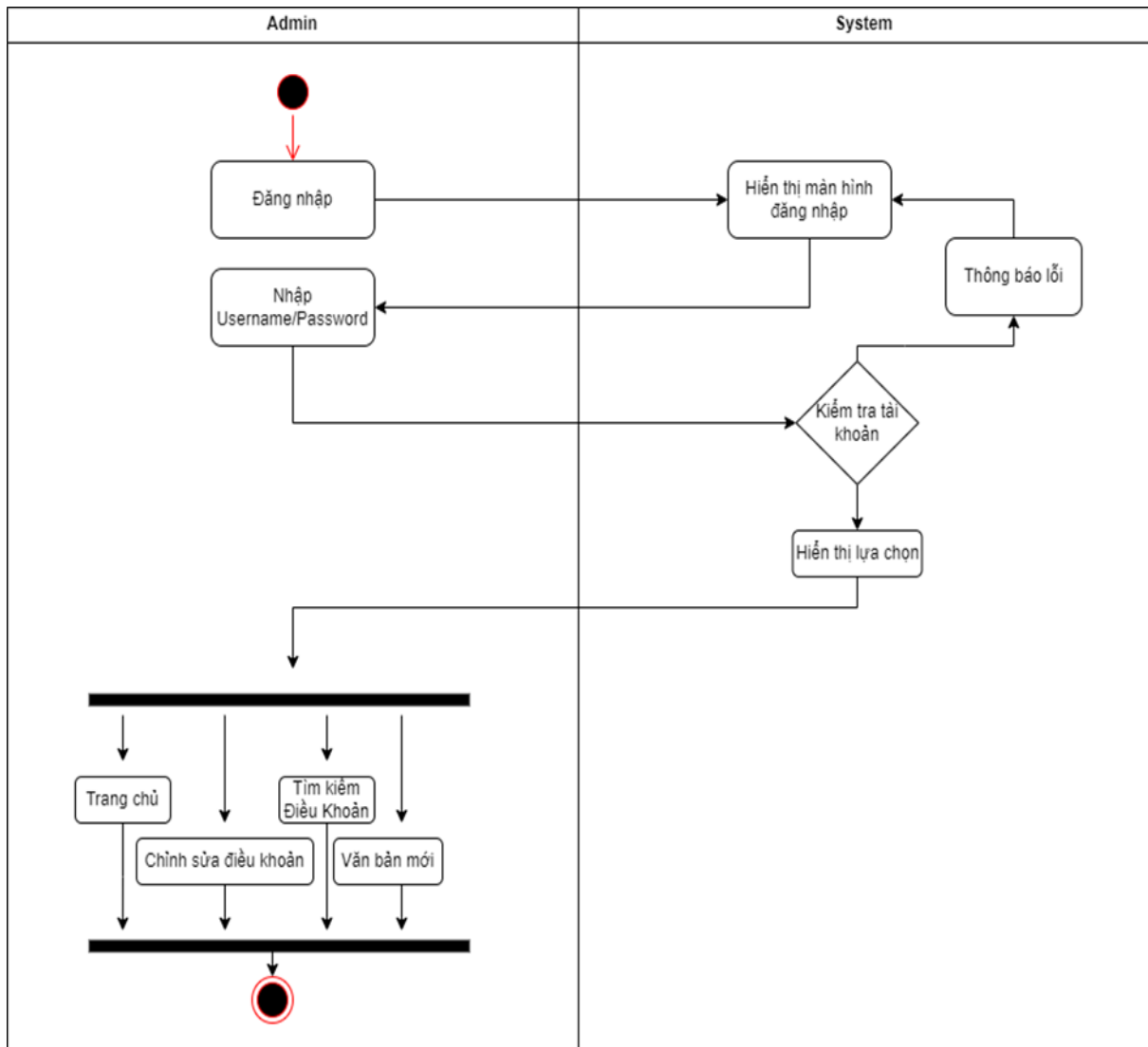


Quản lý tài khoản

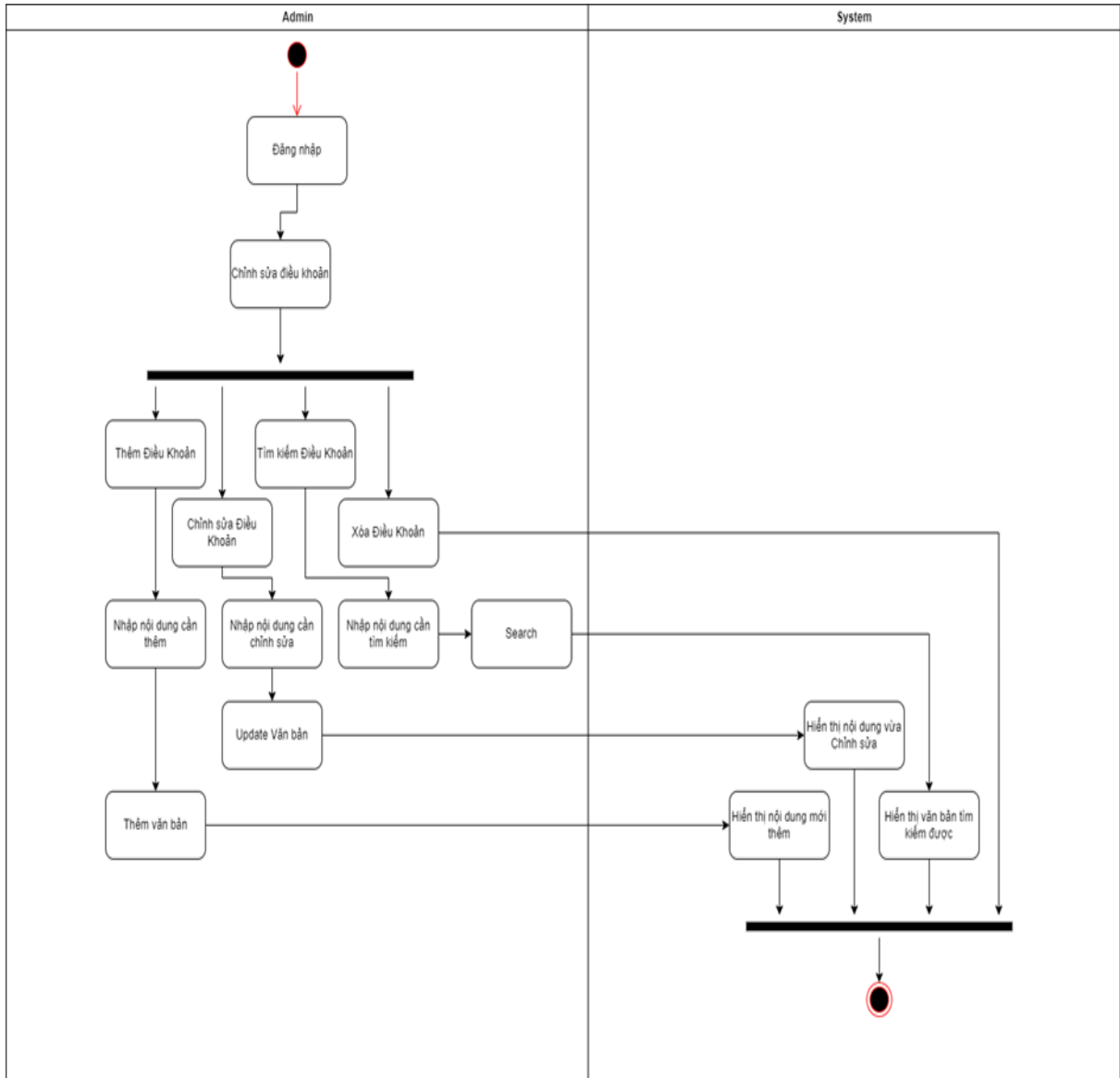


3.3.2 Admin

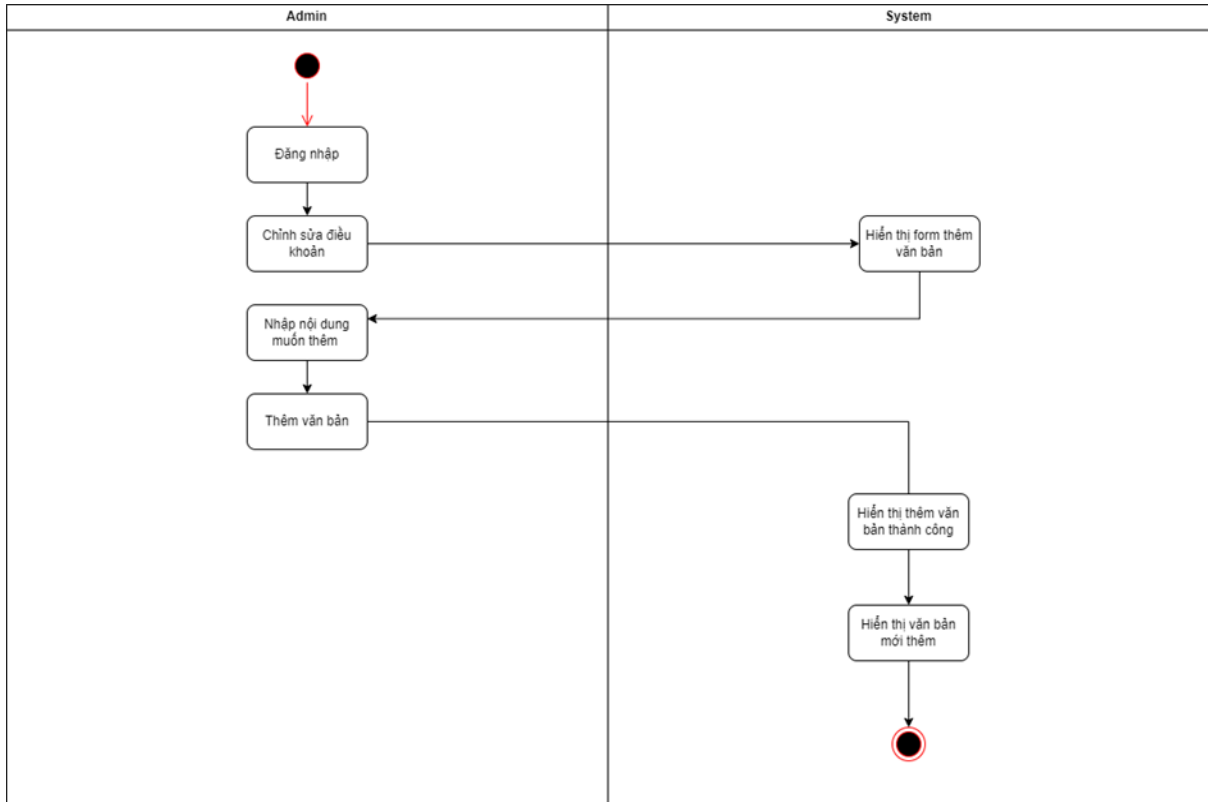
Tổng quan



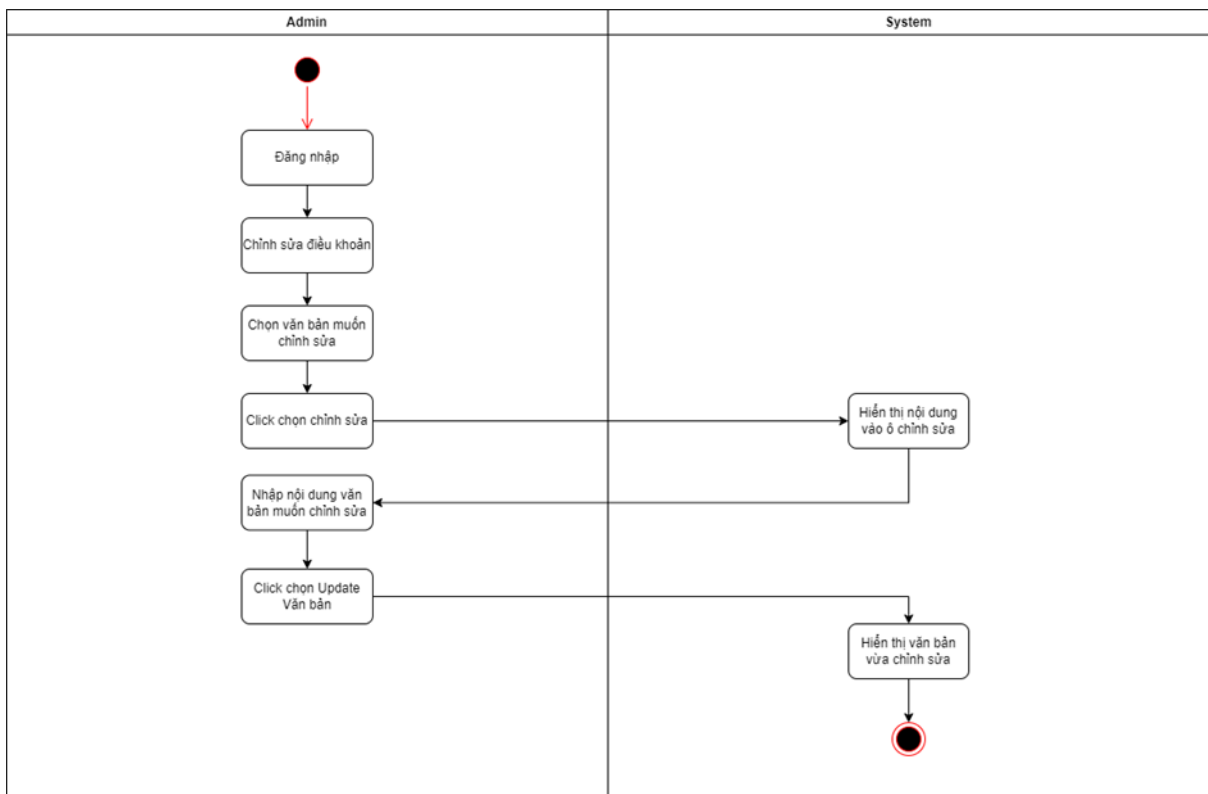
Admin



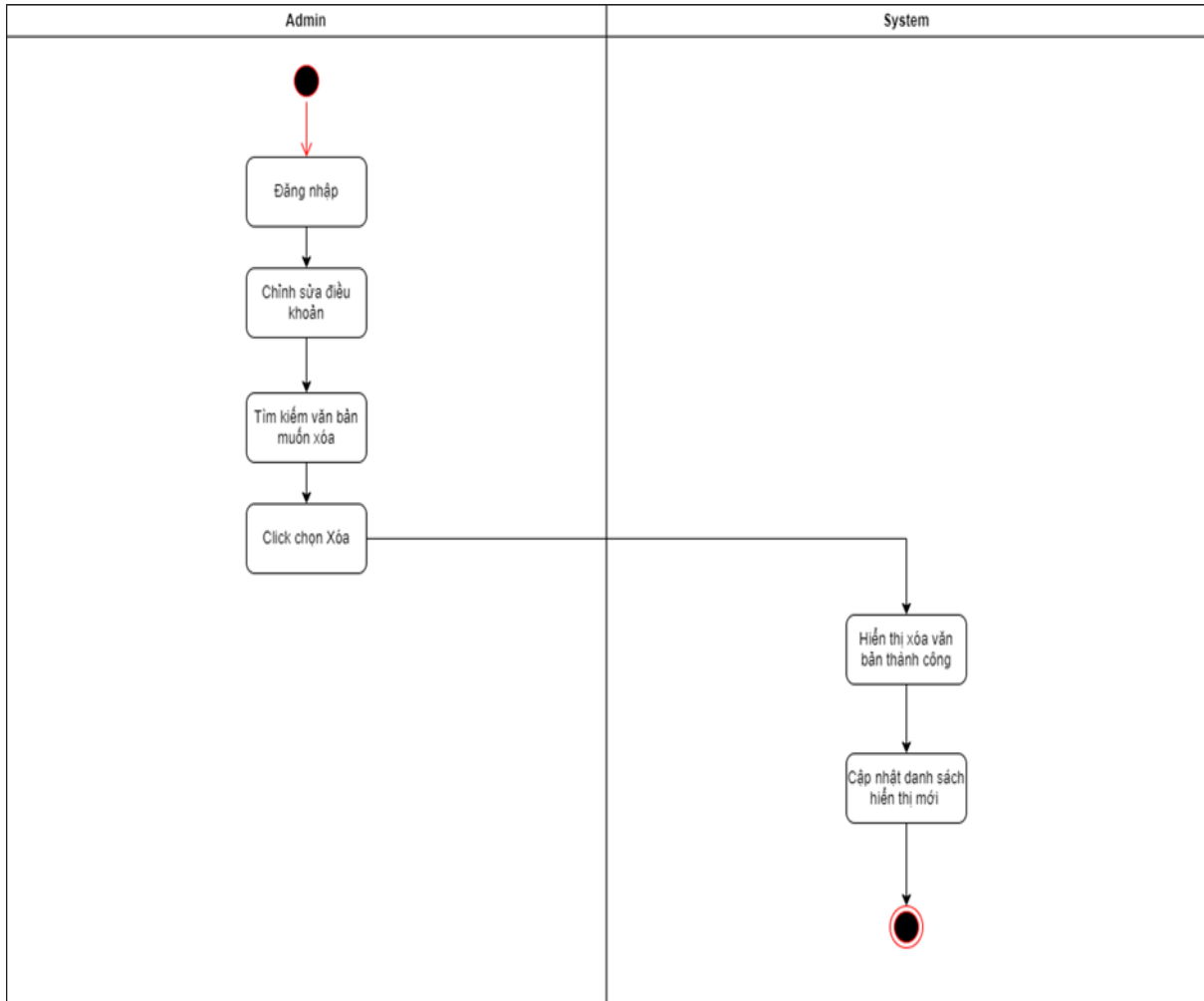
Thêm điều khoản



Sửa điều khoản

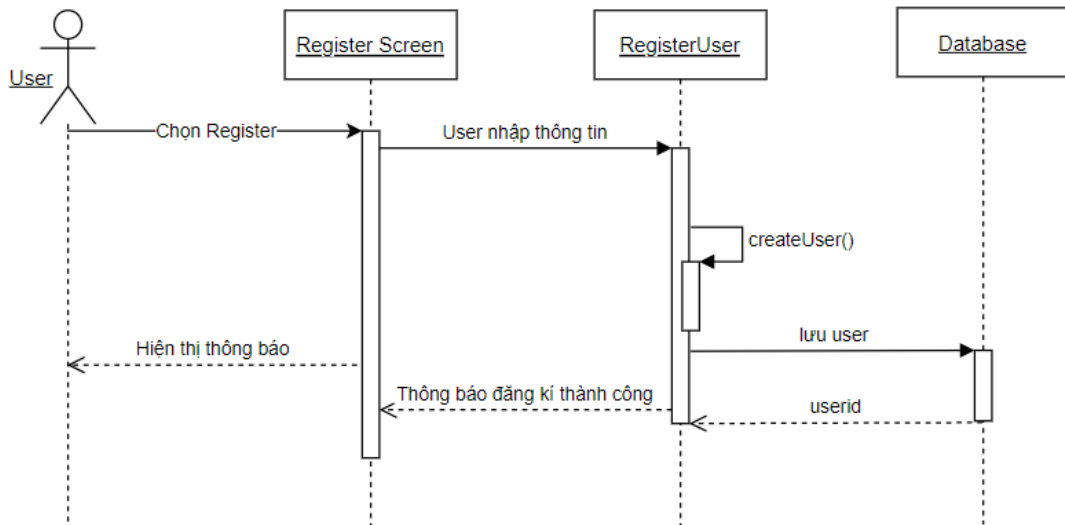


Xóa điều khoản

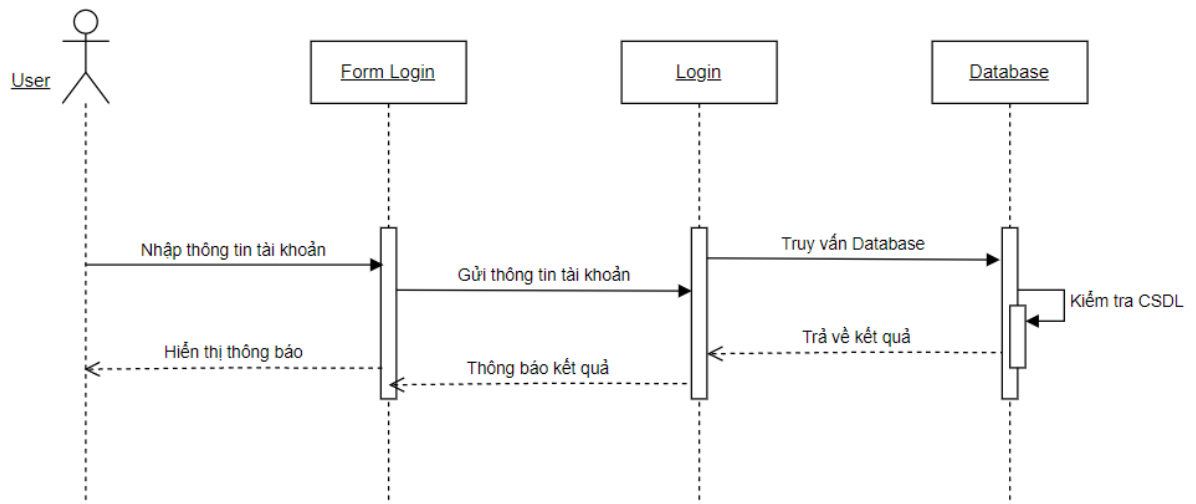


3.4 Sequence Diagrams

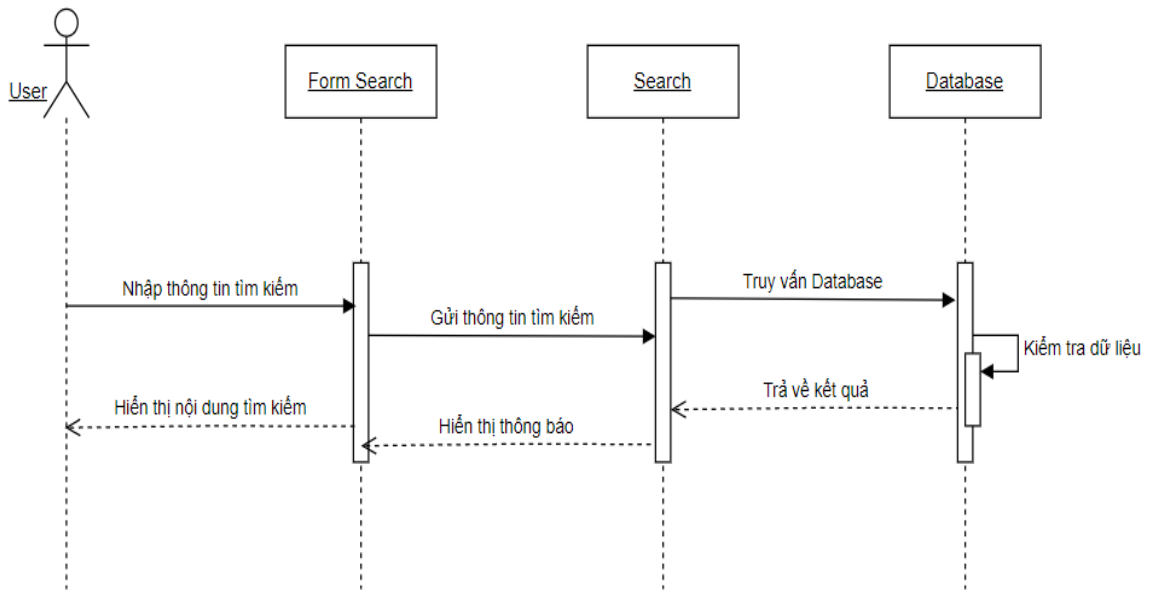
3.4.1 Đăng ký tài khoản



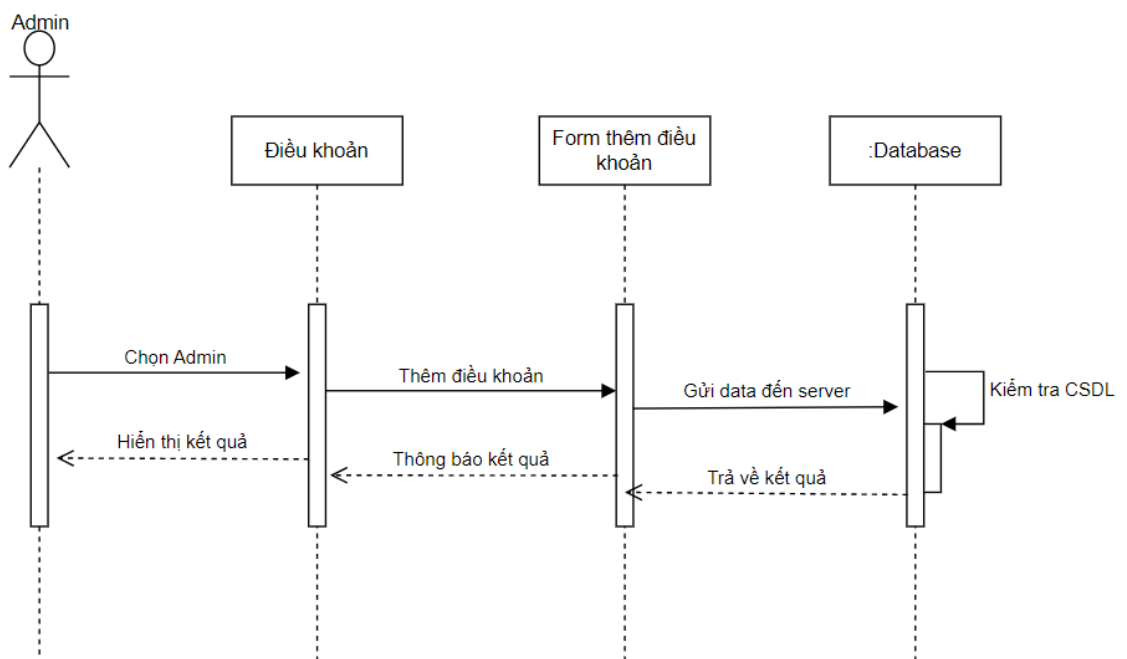
3.4.2 Đăng nhập tài khoản



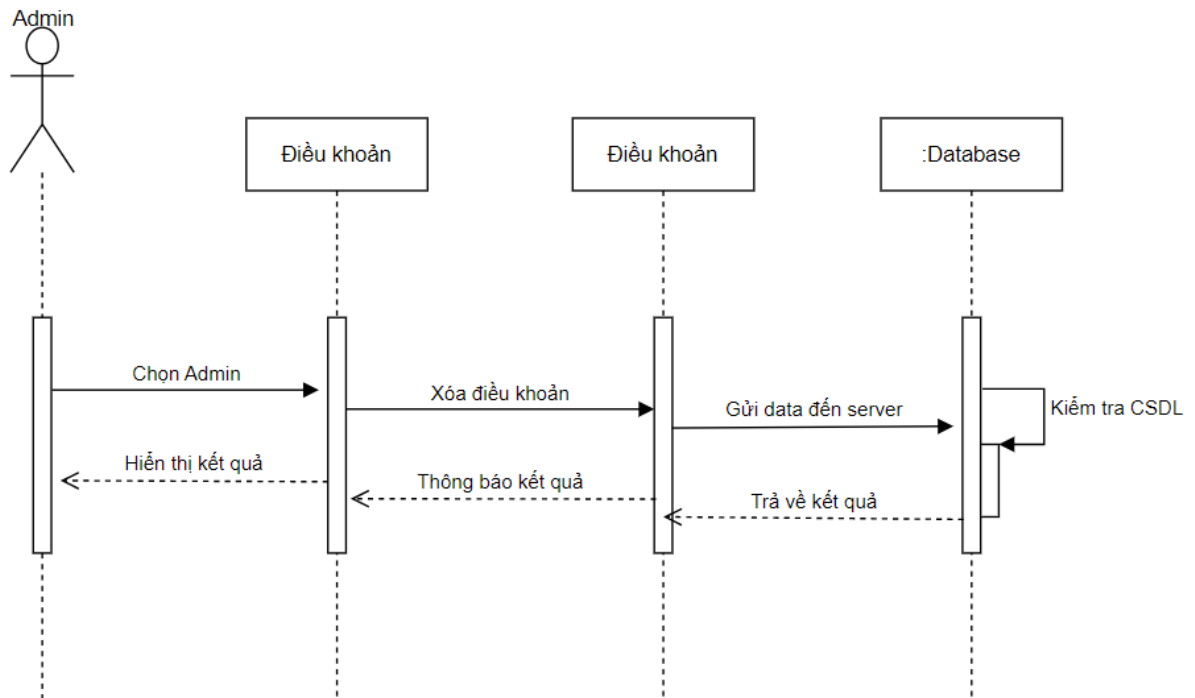
3.4.3 Tìm kiếm điều/khoản



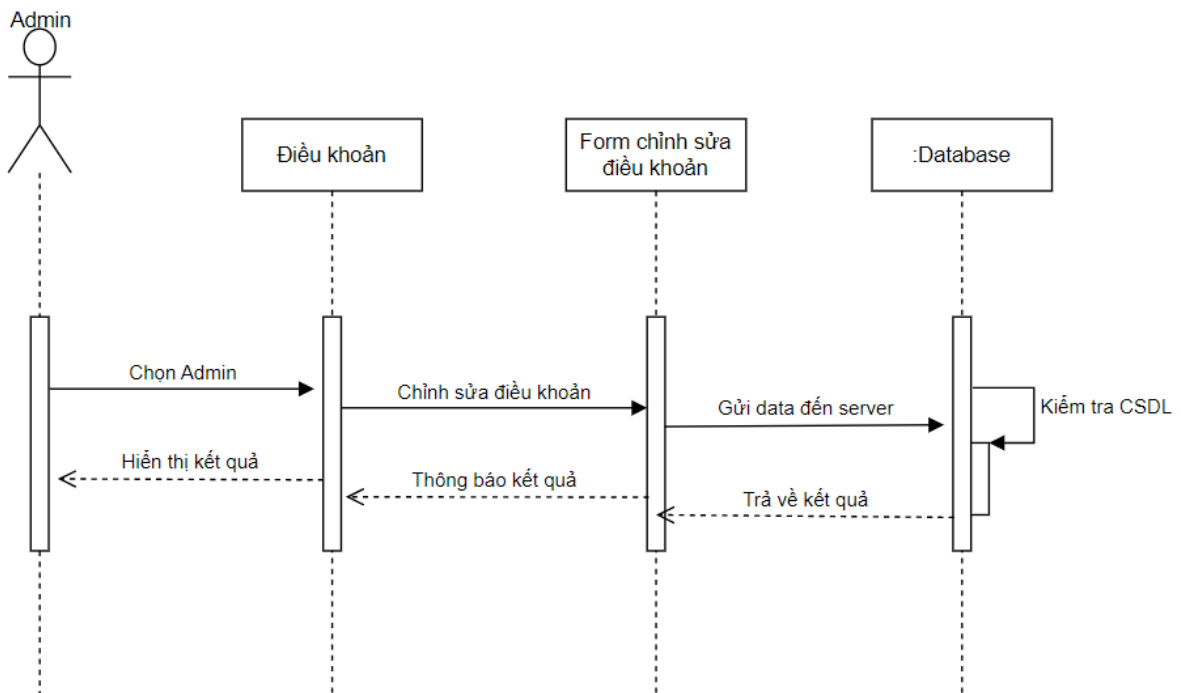
3.4.4 Thêm điều/khoản



3.4.5 Xóa điều/khoản



3.4.6 Chỉnh sửa điều/khoản



3.5 Database

Mô hình Law

```

const Law = new Schema(
  {
    content: { type: String },
    chuong: { type: String, default: 'Đang cập nhật..' },
    khoan: { type: String, default: 'Đang cập nhật..' },
    dieu: { type: String, default: 'Đang cập nhật..' },
    muc: { type: String, default: 'Đang cập nhật..' },
    priceabove: { type: String, default: 'Đang cập nhật..' },
    pricebelow: { type: String, default: 'Đang cập nhật..' },
    hiuluc: { type: String, default: 'Đang cập nhật..' },
    apdung: { type: String, default: 'Đang cập nhật..' },
  },
  { timestamps: true }
);

```

Mô hình User

```

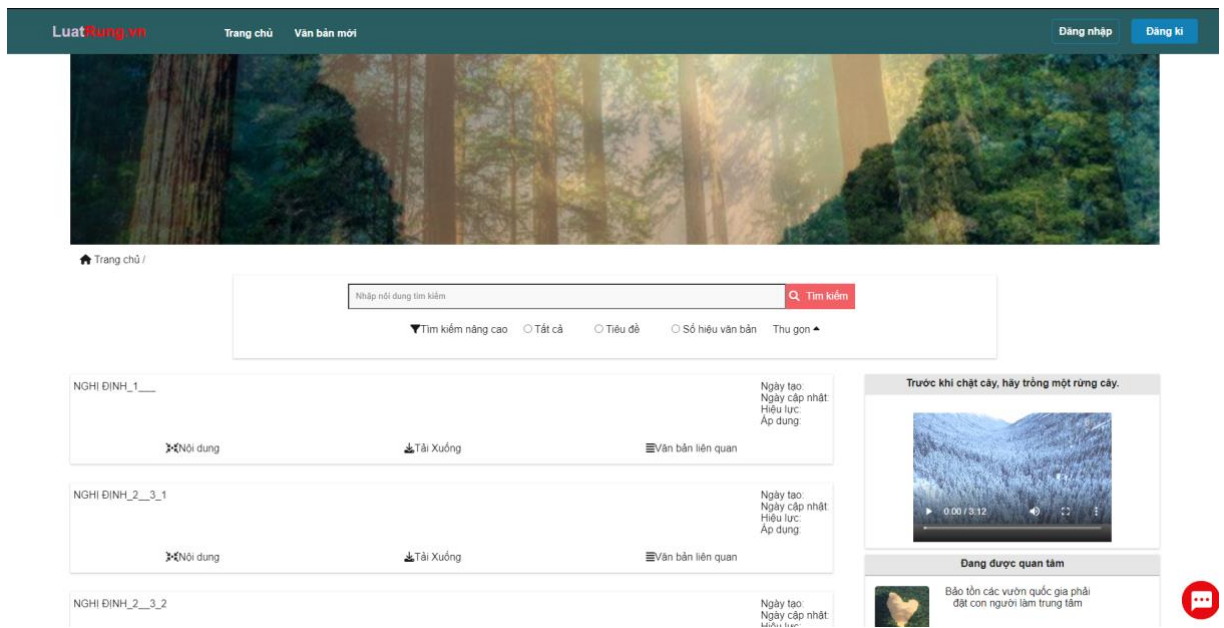
const User = new Schema({
  name: { type: String },
  email: { type: String, },
  password: { type: String,},
  repassword: { type: String },
  address: { type: String },
  phone: { type: String },
});

```

3.6 Thiết kế giao diện

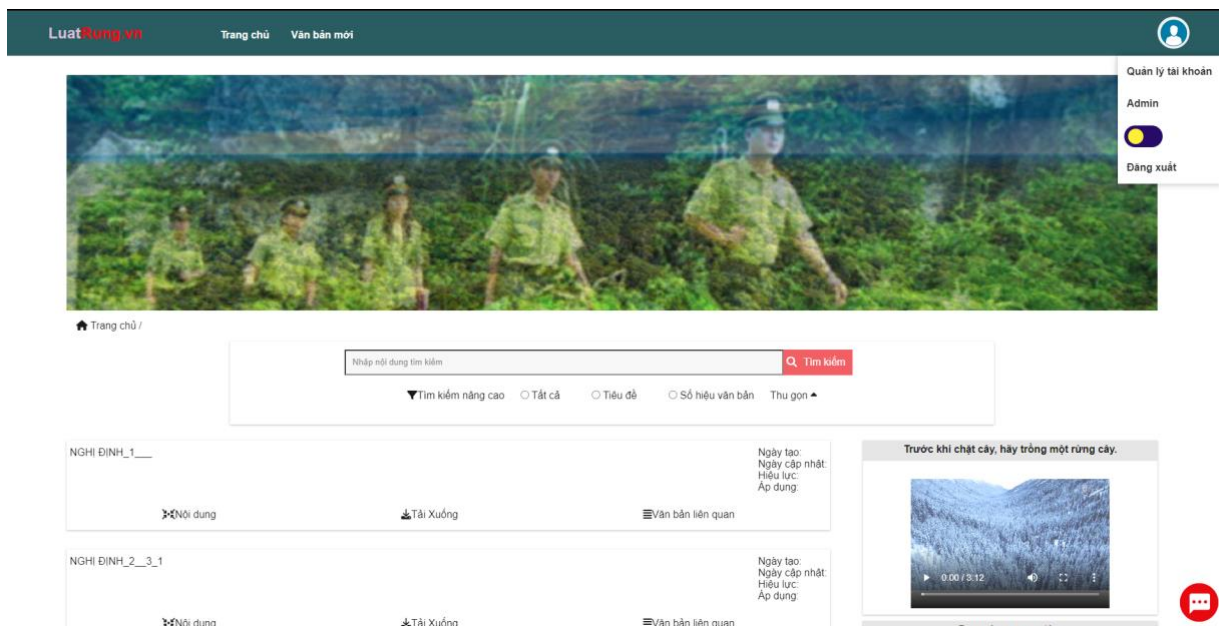
3.6.1 Giao diện trang chủ

+ Đối với người sử dụng (không cần đăng nhập)



Mô tả giao diện:

- Đối với người dùng sau khi vào trang web thì người dùng có thể xem điều luật và tìm kiếm điều luật mà không cần phải đăng nhập
- + Đối với người sử dụng (đã đăng nhập)



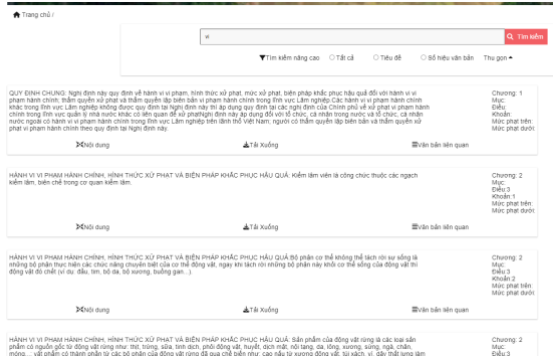
- Mô tả giao diện:

- Sự khác nhau giữa đăng nhập và không đăng nhập là người dùng có thêm các chức năng như là quản lý tài khoản, thay đổi màu nền để phù hợp với mắt

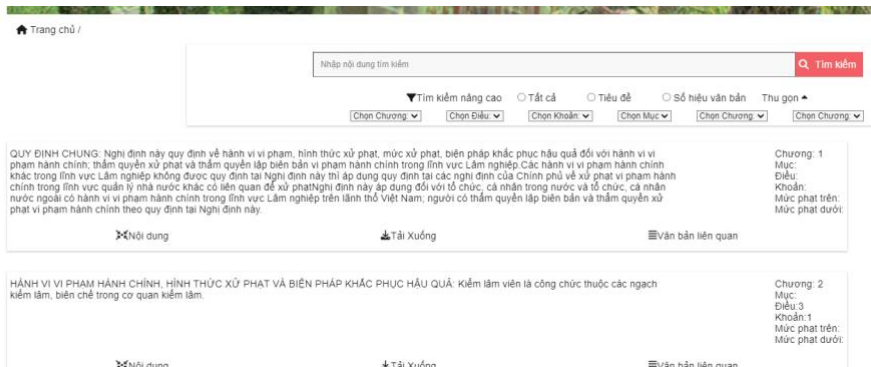
-Chức năng tìm kiếm: Sau khi người dùng nhập từ khóa website sẽ xử lí yêu cầu và đưa ra kết quả phù hợp

+ ví dụ khi nhập từ “ vi ” trang web sẽ đưa ra kết quả chi tiết như

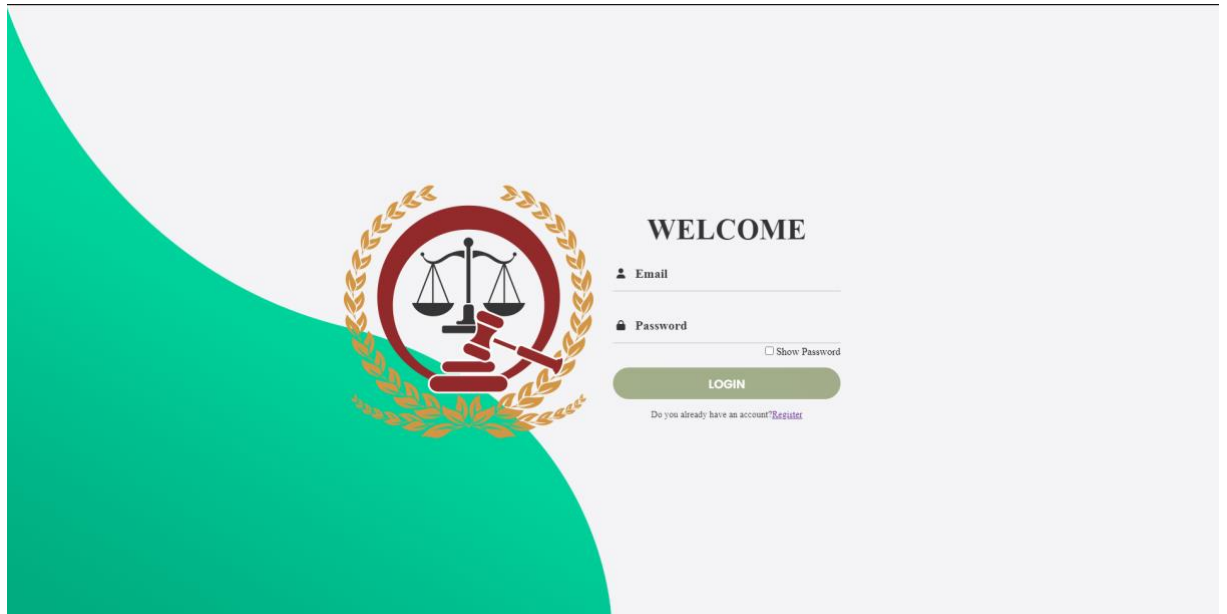
sau:



+ Người dùng có thể tùy chọn chức năng nâng cao nhằm tìm được điều luật như ý muốn:



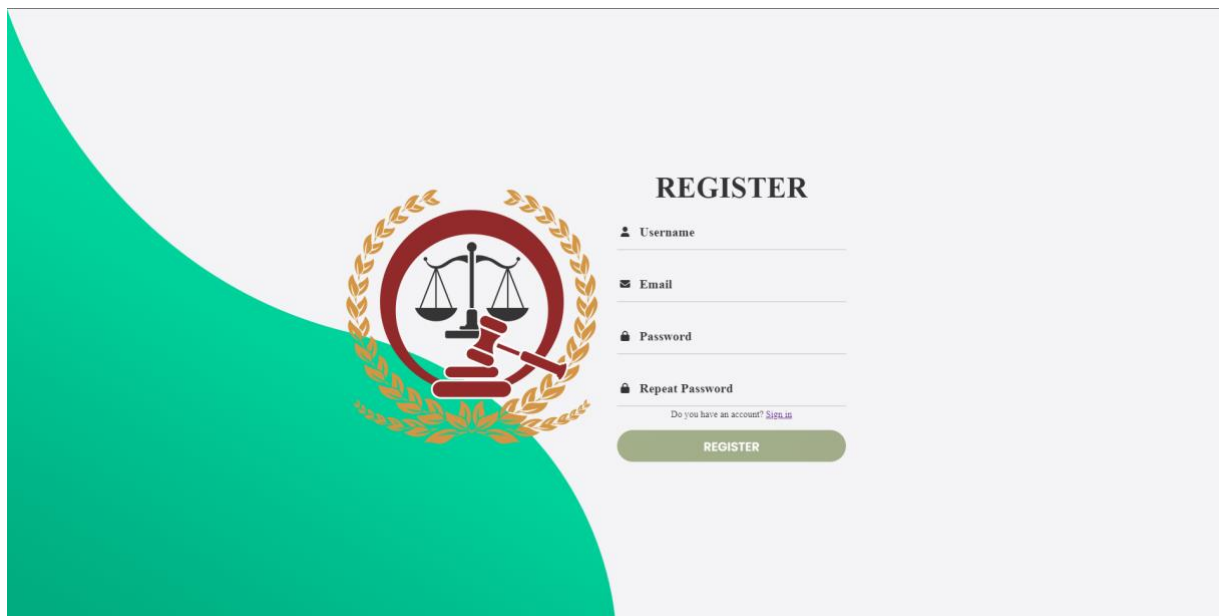
3.6.2 Giao diện đăng nhập



-Mô tả giao diện:

-Người dùng sẽ nhập email và password như đã đăng kí nếu đúng thì sẽ đưa người dùng đến trang chủ

3.6.3 Giao diện đăng ký



-Mô tả giao diện:

+ Người dùng khi đăng kí phải nhập username mà mình muốn, email và password. Nếu phù hợp sẽ thông báo đăng kí thành công và đưa người dùng về trang Đăng nhập

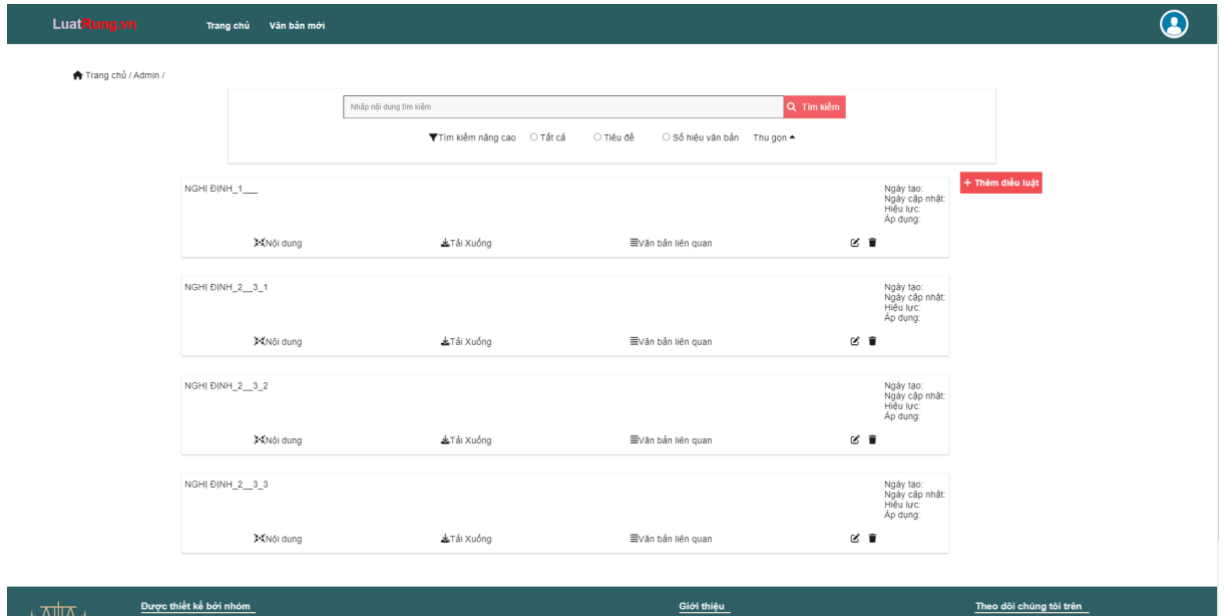
3.6.4 Giao diện quản lý tài khoản

The screenshot shows the user account management page on the website LuatRung.vn. The page has a dark green header with the logo and navigation links. Below the header is a banner image of a gavel and scales of justice. A user profile card is visible, showing a placeholder for a profile picture and the name 'User'. To the right of the profile card is a 'Quản lý tài khoản' (Manage Account) link. The main content area is divided into two columns. The left column contains account details: 'Tài khoản: info@gmail.com', 'Số người sử dụng: 1 người sử dụng', 'Ngày đăng ký: 17/12/2022', and 'Thời hạn sử dụng: 17/12/2023'. Below this is a form with input fields for 'Họ và tên', 'Email', 'Số điện thoại', 'Địa chỉ', 'Thành Phố', 'Ngày sinh', and 'Giới Tính'. A red 'Cập nhật' (Update) button is at the bottom of the form. The right column features a promotional banner for 'CẢN TIỀN GỌI LIÊN 88' and a social media post with the text 'ARE YOU OKAY?' and several hashtags.

Mô tả giao diện:

+ Người dùng sau khi đăng nhập có thể sử dụng chức năng này nhằm phục vụ cho các mục đích liên quan

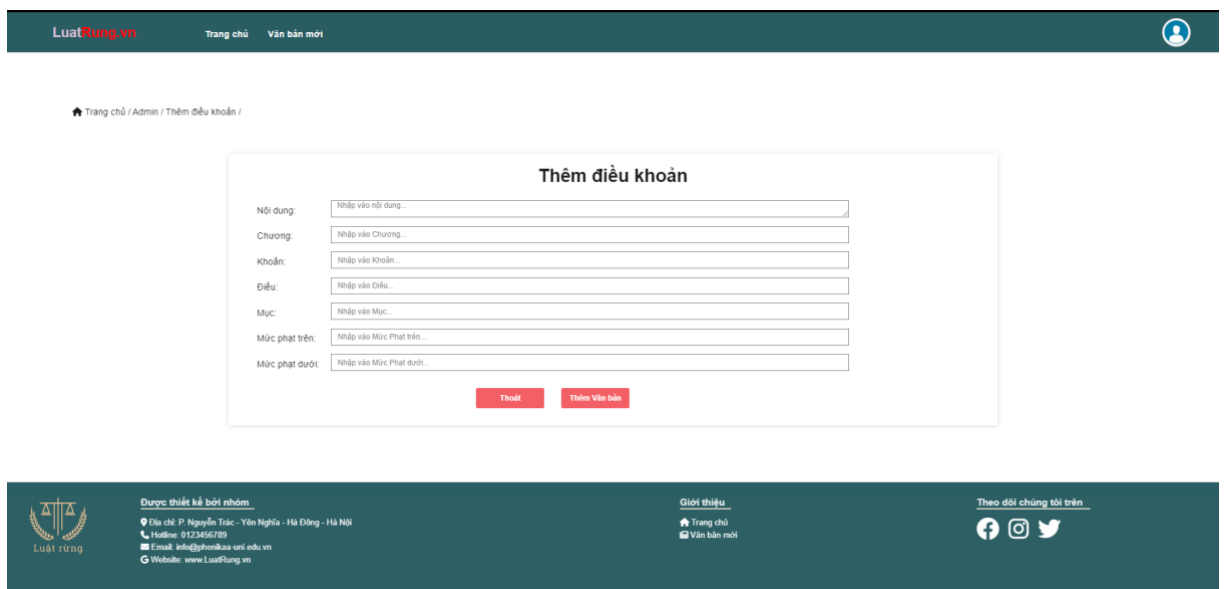
3.6.5 Giao diện Admin



-Mô tả giao diện

+ Trang Admin chỉ có người dùng được cấp phép mới có thể vào trang admin này. Trong đây Admin có thể thêm sửa và xóa các điều luật mà mình muốn

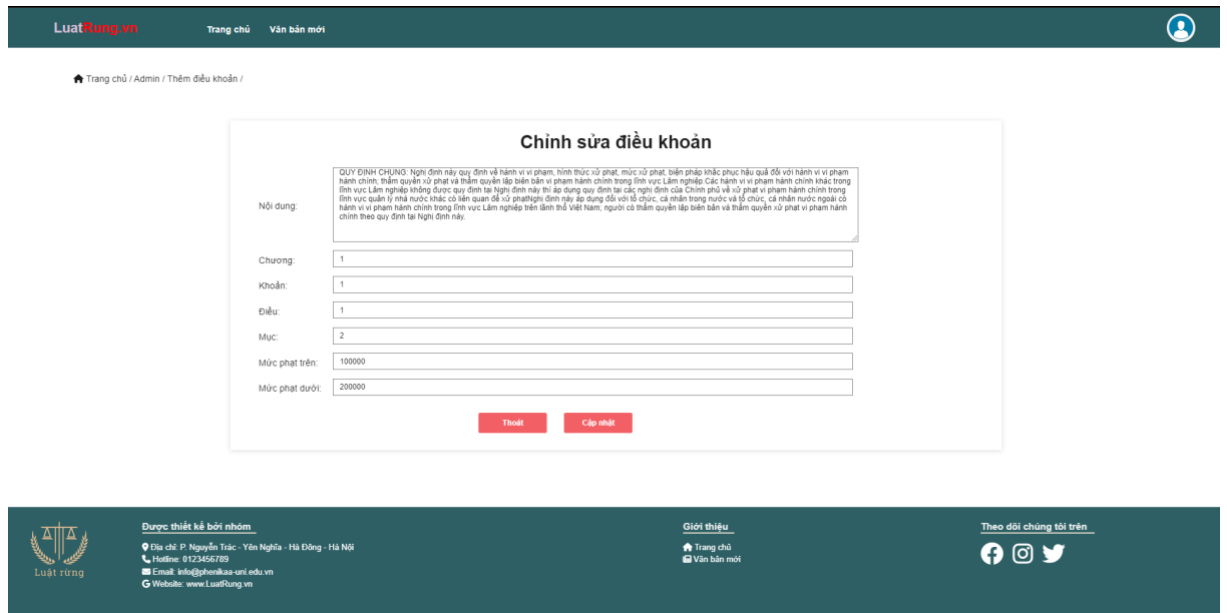
3.6.6 Giao diện thêm điều luật



-Mô tả giao diện:

+Khi Admin bấm thêm điều luật thì website sẽ hướng admin đến 1 trang khác và Admin có thể thêm điều luật, điều luật được thêm sẽ được add vào database

3.6.7 Giao diện sửa điều luật



-Mô tả giao diện

+ Khi bấm vào chỉnh sửa website sẽ lấy Id từ database tương ứng với điều luật và sẽ đưa các thông tin vào ô nhằm mục đích tiện cho việc chỉnh sửa. Sau khi chỉnh sửa bấm update và thông tin sẽ được update

3.7. Mô tả code và chức năng

- Trang web Luật được phân ra thành hai luồng ứng với các Router

+ Luồng người dùng: Ở đây sẽ có các chức năng chính như là: Xem điều luật , Tải điều luật, Tìm kiếm điều luật, Xem văn bản mới,..

+Luồng Admin: Ở đây sẽ có các chức năng giống như người dùng nhưng sẽ có thêm chức năng: Thêm, Sửa , Xóa,.

- Các bước thực hiện: Tạo ra 1 file index.js như sau:

```

const express = require('express');
const app = express();
const port = 3000;
const morgan = require('morgan')
const route = require('./route')
const methodOverride = require('method-override')
const db = require('./config/db');
const path = require('path');
// CONNECT DB

db.connect();

// use middleware to call function put patch delete,...

app.use(methodOverride('_method'))

// parse body

app.use(express.urlencoded({ extended: true }))

// SET VIEW HANDLEBAR
const hbs = require('express-handlebars')
app.engine('.hbs', hbs.engine({
  extname: '.hbs',

}))
app.set('view engine', '.hbs');
app.set('views', './src/resources/views')

app.use(morgan('combined'))

// set static route (if path coincide public/... it redirect path u put)
app.use(express.static('./src/public'))

```

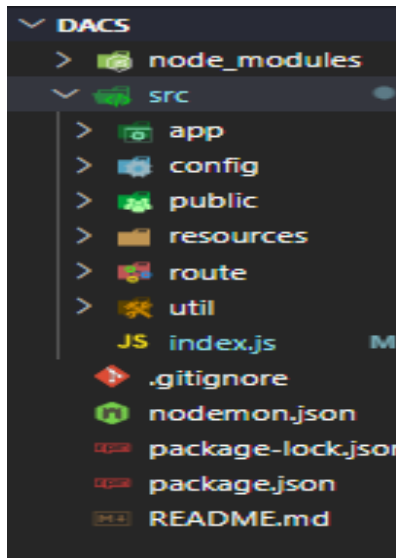
```

route(app);

app.listen(port, () => {
  console.log(`App listening at http://localhost:${port}/`);
})

```

- Ở đây chúng ta sẽ setup cho project các cái thư viện cần thiết để dễ dàng sử dụng trong quá trình Code (Connect database Express Library), ...



- Thứ hai Em sẽ setup các folder cho từng nhiệm vụ của nó:

- +App chứa các thư mục Controller(đây là nơi tạo các chức năng để Render), models(đây là nơi tạo ra DataBase phù hợp)
- +Config chứa các thư mục Data(ở đây chứa dữ liệu import vào MongoDB), db(trong đây chứa file connectdb)
- +Public chứa các thư mục tĩnh như Css, file, img.
- +Resources chứa thư mục views: chịu trách nhiệm tạo ra các giao diện cho người dùng trải nghiệm(trong đây sẽ chứa các thư mục body: phần thân của website), patials(chứa phần header và footer), layouts(chứa các bố cục mà Dev muốn đưa đến người dùng OK)

- Thứ ba là cách tạo ra các chức năng chính của trang Web

- +Render dữ liệu: Ở đây Footer ,Header sẽ được nạp vào layouts như bên dưới. Body sẽ được tùy chỉnh ở trong controller.

```

<div id="toast"></div>
<div class="wrap" id="wrap">

```



```
    {{> header-afterLog}}

    {{{body}}}

    {{> footer}}
</div>
```

Ở file controller sẽ đưa nhiệm vụ cho layouts là Render ra tất cả các điều khoản.

```
show(req, res, next) {
  Law.find({}).then((Law) => {
    res.render('body/default', {
      layout: 'home.hbs',
      law: multipleMongooseToObject(Law),
    });
  });
};
```

Và câu lệnh sẽ được thực thi ở file Router (Ở đây chứa các route làm các nhiệm vụ riêng biệt)

```
var express = require('express');
const route = require('.');
var router = express.Router();

var homeController =
require('../app/controllers/HomeController')

router.get('/:id/noidung', homeController.content)

router.get('/admin/delete/:id', homeController.delete);
```

```

router.put('/admin/edit/:id', homeController.update)

router.get('/admin/:id/edit',homeController.edit)

router.post('/admin/store',homeController.store)
router.get('/admin/add',homeController.add)

// router.get('/:id/profile',homeController.edit_profile)
router.get('/profile',homeController.edit_profile)

router.get('/admin', homeController.admin);

router.get('/newpara/:page', homeController.newpara);

router.get('/ketqua',homeController.results)

router.get('/home/:page', homeController.show);

router.get('/',homeController.show)

module.exports = router;

```

-Thứ 4: Chức năng Thêm , Sửa, Xóa:

+ Chức năng Thêm: Em sẽ tạo ra một body mới có tên là Add như bên dưới.

Tiếp đó

Sau khi nhập các Nội dung vào các Ô Input dữ liệu sẽ được trả về Request
Và đến file Controller.

```
<form class="add" method="POST" action="/home/admin/store">
```

```

    <h1 style="display: flex; justify-content: center;
padding: 20px">Thêm điều khoản</h1>

    <div class="form-group">
        <label for="content">Nội dung:</label>
        <textarea type="text" class="form-control"
id="content" name="content" placeholder="Nhập vào nội
dung..."></textarea>
    </div>

    <div class="form-group">
        <label for="chuong">Chương:</label>
        <input type="text" class="form-control"
id="chuong" name="chuong" placeholder="Nhập vào Chương..." >
    </div>

    <div class="form-group">
        <label for="khoan">Khoản:</label>
        <input type="text" class="form-control"
id="khoan" name="khoan" placeholder="Nhập vào Khoản...">
    </div>

    <div class="form-group">
        <label for="dieu">Điều:</label>
        <input type="text" class="form-control" id="dieu"
name="dieu" placeholder="Nhập vào Điều...">
    </div>

    <div class="form-group">

```

```

        <label for="muc">Mục:</label>
        <input type="text" class="form-control" id="muc"
name="muc" placeholder="Nhập vào Mục...">
    </div>

    <div class="form-group">
        <label for="priceabove">Mức phạt trên:</label>
        <input type="text" class="form-control"
id="priceabove" name="priceabove" placeholder="Nhập vào Mức
Phạt trên...">
    </div>

    <div class="form-group">
        <label for="pricebelow">Mức phạt dưới:</label>
        <input type="text" class="form-control"
id="pricebelow" name="pricebelow" placeholder="Nhập vào Mức
Phạt dưới...">
    </div>

    <div class="btnFuntion">
        <a style="text-decoration: none;"
href="/home/admin">
            <input class="btn" type="button"
value="Thoát"/>
        </a>
        <input class="btn btn--success" type="submit"
onclick="showSuccessToast()" value="Thêm Văn bản"/>
    </div>
</form>

```

Ở file Controller dữ liệu nhận về sẽ được lưu tại Monggodb thông qua phương thức Save()

```
// [POST] /admin/store
store(req, res) {
  const law = new Law(req.body);
  setTimeout(() => {
    law.save();
    res.redirect('/home/admin/add');
  }, 1500);
}
```

+ Chức năng chỉnh sửa: Ở chức năng này sẽ khá giống với chức năng Thêm nhưng khi bấm chỉnh sửa vào một điều luật nào đó thì Website sẽ lấy ra id tương ứng và sử dụng phương thức UpdateOne

```
//Sửa
// [GET] /admin/:id/edit
edit(req, res) {
  Law.findById(req.params.id)
    .then((Law) => {
      res.render('body/edit', {
        layout: 'home.hbs',
        law: mongooseToObject(Law),
      });
    });
}

// [PUT] /admin/edit/:id
update(req, res) {
  Law.updateOne({ _id: req.params.id }, req.body)
    .then(() => {
```

```

        setTimeout(() => {
            res.redirect('/home/admin')
        }, 1500);
    })
}

```

+ Chức năng xóa: Ở chức năng xóa request sẽ nhận được yêu cầu xóa theo Id trả về.

```

//Xóa
// [GET] /admin/delete/:id
delete(req, res) {
    Law.deleteOne({ _id: req.params.id },
req.body).then(() =>
        res.redirect('/home/admin')
    );
}

```

- Thứ 5 Chức năng tìm kiếm: Ở chức năng tìm kiếm sẽ tìm theo nội dung được nhập nội dung vào Ô Input. Website sẽ trả về request và nó sẽ xử lý tìm các document chứa nội dung được nhập đó.

```

results(req, res) {
    Law.find({ content: { $regex:
req.query.noidung } }).then((Law) => {
        res.render('body/search', {
            layout: 'home.hbs',
            law: multipleMongooseToObject(Law),
        });
    });
}

```

-Thứ 6 Chức năng Đăng nhập, Đăng ký:

+ Chức năng Đăng ký: Ở đây người dùng sẽ nhập các thông tin yêu cầu Website sẽ trả về request chứa các thông tin đó và trong controller sẽ xử lý Rồi lưu vào database.

```
store(req, res) {
  const user = new User(req.body);
  user.save();
  setTimeout(function () {
    res.redirect('/login');
  }, 1500)
}
```

+Chức năng Đăng nhập: Ở đây người dùng sẽ nhập các thông tin như email, mật khẩu. Website sẽ trả về request ở tại controller sẽ xử lý nếu trùng khớp với dữ liệu trong database thì sẽ đưa người dùng vào tới Trang chủ.

```
// LOGIN function
dangnhap(req, res, next) {
  res.render('body/login', {layout: 'log.hbs'})
}

check(req, res){
  User.find({})
  .then(user => {
    const emailuser = user.map(function (params)
{
    return params.email
  })
})
```

```

const passworduser = user.map(function (Long)
{
    return Long.password
})

const email = req.query.email;
const password = req.query.password;

if(emailuser.includes(email) &&
passworduser.includes(password) ){
    setTimeout(function () {
        res.redirect('/home')
    },3000)
}
else{

}

})

}

```

IV. Kết Luận

4.1 Kết quả đạt được

Trong thời gian thực hiện dự án, nhóm chúng em đã có những kết quả đạt được như sau:

- Áp dụng những kiến thức đã học vào việc khảo sát, thu thập thông tin, phân tích và thiết kế hệ thống đến việc xây dựng hệ thống hoàn chỉnh.
- Thiết kế hệ thống khá hoàn chỉnh để tra cứu thông tin luật pháp với các chức năng cơ bản khá hoàn thiện như: Thêm, Sửa, Xóa, Tìm kiếm, Download các Điều/Khoản hay văn bản luật.
- Tìm hiểu về 1 hệ cơ sở dữ liệu phi quan hệ (NoSQL) MongoDB.
- Rèn luyện kỹ năng làm việc nhóm, biết cách phân chia công việc nhóm hợp lý để mọi người phát huy được điểm mạnh của mình.

4.2 Ưu điểm

- Chương trình mô tả khá sát các yêu cầu của đề tài.
- Thiết kế cơ sở dữ liệu tốt, dễ dàng cho việc truy vấn.
- Hiển thị thông tin chi tiết, trực quan.
- Hoàn thành bài tập đúng hạn, nộp đầy đủ kế hoạch công việc theo tuần.

4.3 Nhược điểm

- Giao diện còn đơn giản, chưa có nhiều cải tiến về mặt hình ảnh hiển thị.
- Chưa xây dựng được chức năng

4.4 Hướng phát triển

- Nghiên cứu thêm nhiều chức năng như:
- Xây dựng trang web sử dụng đa ngôn ngữ cho người dùng. Chẳng hạn như một văn bản có thể đọc bằng tiếng Anh hoặc tiếng Việt. để người bản xứ tham khảo pháp luật Việt Nam.
- Phát triển hệ thống giúp trang web lưu trữ nhiều các văn bản pháp luật khác như: Hiến pháp, Nghị quyết của quốc hội, Pháp lệnh, Sắc lệnh...

V. Đóng góp từng thành viên

Công việc		Người phụ trách
Lập trình Front-end	Trang Đăng Nhập + Trang Đăng Kí	-Nguyễn Viết Tuấn

	Trang chủ	-Nguyễn Viết Tuấn
	Trang Admin: Thêm sửa xóa	-Bùi Khánh Huyền
	Trang quản lí người dùng	-Bùi Khánh Huyền
	Trang Văn bản mới	-Nguyễn Viết Tuấn
Lập trình back-end	Render Dữ liệu từ DB	-Hoàng Hải Long -Cao Thạch Đức Mạnh
	Thêm - sửa – xóa điều luật	-Cao Thạch Đức Mạnh -Hoàng Hải Long
	Đăng nhập – Đăng kí	-Cao Thạch Đức Mạnh -Hoàng Hải Long
	Tìm Kiếm	-Hoàng Hải Long
	Cập nhật thông tin cá nhân	-Cao Thạch Đức Mạnh
Báo cáo mỗi tuần + slide		-Bùi Khánh Huyền
Thiết kế cơ sở dữ liệu		-Nguyễn Viết Tuấn
Nghiên cứu công nghệ		-Hoàng Hải Long -Cao Thạch Đức Mạnh -Nguyễn Viết Tuấn -Bùi Khánh Huyền
Đặc tả Use case + Activity		-Nguyễn Viết Tuấn -Bùi Khánh Huyền
Kiểm thử		-Bùi Khánh Huyền