HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐỖ THỊ HỒNG NGÁT

NGHIÊN CỨU XÂY DỰNG HỆ THỐNG TỰ ĐỘNG TỔNG HỢP THÔNG TIN VỀ DỊCH BỆNH SỬ DỤNG SCRAPY FRAMEWORK

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - NĂM 2024

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐỖ THỊ HỒNG NGÁT

NGHIÊN CỨU XÂY DỰNG HỆ THỐNG TỰ ĐỘNG TỔNG HỢP THÔNG TIN VỀ DỊCH BỆNH SỬ DỤNG SCRAPY FRAMEWORK

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH

MÃ SỐ: 8.48.01.01

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. VŨ VĂN THỎA

HÀ NỘI – NĂM 2024

LỜI CAM ĐOAN

Học viên xin cam đoan đề án: "Nghiên cứu hệ thống tự động tổng hợp thông tin về dịch bệnh sử dụng Scrapy framework" là do học viên nghiên cứu và hoàn thành dưới sự hướng dẫn của TS. Vũ Văn Thỏa. Nội dung có tham khảo từ các tạp chí và trang web đáng tin cậy. Các tài liệu này cung cấp thông tin chính xác và được nghiên cứu kĩ lưỡng bao gồm cả số liệu, dữ liệu được thu thập và kết quả nghiên cứu.

Học viên xin chịu trách nhiệm về lời cam đoan này.

Hà nội, ngày 27 tháng 03 năm 2024

Tác giả đề án tốt nghiệp

(ký và ghi rõ họ tên)

Đỗ Thị Hồng Ngát

LỜI CẢM ƠN

Trong quá trình học tập và hoàn thành đề án tốt nghiệp, học viên muốn bày tỏ lòng biết ơn chân thành đến sự hỗ trợ, động viên từ thầy cô, gia đình và bạn bè.

Đầu tiên, học viên muốn bày tỏ lòng biết ơn sâu sắc đến TS. Vũ Văn Thỏa, người thầy tận tâm đã cung cấp cho học viên những nhận xét, hướng dẫn quan trọng để học viên có thể lựa chọn đề tài và hướng dẫn trực tiếp trong suốt quá trình nghiên cứu và hoàn thành đề án tốt nghiệp.

Học viên chân thành cám ơn các thầy giáo, cô giáo trong học viện đã tạo điều kiện và nhiệt tình giúp đỡ học viên hoàn thành khóa học cao học này.

Xin chân thành cám ơn các lãnh đạo, các đồng nghiệp tại nơi công tác, gia đình và bạn bè tạo mọi điều kiện để học viên hoàn thành khóa học.

Em xin chân thành cảm ơn !

Hà nội, ngày 27 tháng 05 năm 2024

Tác giả đề án tốt nghiệp

(ký và ghi rõ họ tên)

Đỗ Thị Hồng Ngát

MỤC LỤC

DANH MỤC CÁC BẢNG BIỂU

DANH MỤC CÁC HÌNH VẼ

MỞ ĐẦU

Cùng với sự cải thiện đáng kể về chất lượng cuộc sống, ngày nay, sức khỏe cá nhân và của gia đình trở thành ưu tiên hàng đầu của mọi người. Vào thời đại số hóa, Internet đã trở thành một nguồn thông tin vô cùng quý báu về y tế và sức khỏe. Tuy nhiên, sự phát triển của công nghệ đã tạo điều kiện cho xuất hiện đông đảo các nguồn thông tin trên mạng, điều này đặt ra một thách thức lớn: làm thế nào để phân biệt thông tin chính xác, tin cậy và thông tin không chính thống hoặc giả mạo.

Năm 2020 chứng kiến một trong những biến đổi lớn nhất trong lịch sử loài người - đại dịch Covid-19. Tình hình khẩn cấp này đã buộc thế giới phải thích nghi nhanh chóng với mô hình trực tuyến. Việt Nam, không ngoại lệ, đã tận dụng công nghệ thông tin và Internet để đối phó với đại dịch này. Các công văn quan trọng của chính phủ đã được chuyển từ dạng giấy sang công điện, tiết kiệm thời gian và tài nguyên. Trong bối cảnh đó, sự lan rộng của thông tin chính xác và đáng tin cậy liên quan đến dịch bệnh trở nên cực kỳ quan trọng. Chính phủ và các cơ quan chức năng đã thiết lập các kênh truyền thông trực tuyến chính thức như các trang Facebook để đính chính và cung cấp thông tin đáng tin cậy về dịch bệnh. Điều này đã giúp người dân có nguồn thông tin tin cậy để nắm bắt tình hình dịch bệnh, biện pháp phòng chống, và hướng dẫn cách bảo vệ bản thân và cộng đồng.

Mặt khác, không chỉ trong thời của đại dịch, việc sử dụng công nghệ thông tin và truyền thông trực tuyến đã và đang trở thành một phần quan trọng của cuộc sống hiện đại. Nó mang lại sự linh hoạt, tiết kiệm thời gian và tạo điều kiện thuận lợi cho việc truyền tải thông tin đến mọi người một cách nhanh chóng và hiệu quả.

Trước những yêu cầu trên, hệ thống tự động tổng hợp thông tin về dịch bệnh ra đời để tổng hợp và cung cấp thông tin về dịch bệnh chính thống, đáng tin cậy và dễ dàng tiếp cận cho mọi người. Người dân có thể sử dụng hệ thống này để tìm kiếm thông tin về tình hình dịch bệnh, những bài viết hướng dẫn sức khỏe hữu ích và thậm chí là các bệnh viện, cơ sở y tế có chuyên môn. Các bác sĩ và nhân viên y tế có thể sử dụng hệ thống để cung cấp thông tin đáng tin cậy cho bệnh nhân và cộng đồng. Chính

quyền địa phương có thể tích hợp hệ thống này vào hệ thống quản lý y tế của họ để cung cấp dịch vụ tốt hơn cho người dân.

Với những lý do trên, học viên chọn đề tài "**NGHIÊN CỨU XÂY DỰNG HỆ THỐNG TỰ ĐỘNG TỔNG HỢP THÔNG TIN VỀ DỊCH BỆNH SỬ DỤNG SCRAPY FRAMEWORK**" làm đề án tốt nghiệp cao học của mình.

*** Mục đích, đối tượng và phạm vi nghiên cứu**

- Mục đích nghiên cứu:

+ Khảo sát về hệ thống thông tin, xu hướng phát triển hệ thống tự động tổng hợp thông tin về dịch bệnh trên thế giới và ở Việt Nam.

+ Nghiên cứu về cấu trúc, phương thức hoạt động của Scrapy framework phục vụ cho việc thu thập dữ liệu tự động và một số công cụ, công nghệ để phân tích, thiết kế, xây dựng hệ thống website.

+ Dự kiến sẽ xây dựng một hệ thống website theo kiến trúc tiểu dịch vụ hoàn thiện, với các chức năng cơ bản dành cho người dùng như: tìm kiếm bệnh viện, bài báo, đăng ký tài khoản, thay đổi thông tin cá nhân, đổi mật khẩu, ... Những chức năng cho quản trị viên như: quản lý danh sách bệnh viện, quản lý danh sách bài báo, quản lý người dùng, thu thập dữ liệu tự động tại các trang web khác…Ngoài ra hệ thống sẵn sàng để có thể mở rộng, nâng cấp để phục vụ nhiều người, tích hợp với các hệ thống có sẵn tại các cơ sở triển khai để quản lý dễ dàng hơn.

- ***Đối tượng nghiên cứu:*** Nghiên cứu Scrapy framework, những bệnh viện trên cả nước Việt Nam, các tin tức dịch bệnh bằng tiếng việt.

- ***Phạm vi nghiên cứu:*** cấu trúc, phương thức hoạt động của Scrapy framework, và đề xuất mô hình thử nghiệm hệ thống tại bệnh viện trên cả nước Việt Nam, song tập trung tại các thành phố lớn, nơi có nhiều cơ sở y tế đủ điều kiện triển khai hệ thống.

*** Phương pháp nghiên cứu:**

- Về mặt lý thuyết: tập hợp, khảo sát, phân tích các tài liệu và thông tin có liên quan đến Scrapy framework, các trang web chính thống có thông tin về bệnh viện và tin tức dịch bệnh.

- ***Về mặt thực nghiệm***: Khảo sát tình hình thực tế thu thập dữ liệu tại một số trang web chính thống và đưa ra đề xuất giải pháp phù hợp để triển khai hệ thống tổng hợp thông tin về dịch bệnh.

*** Cấu trúc của đề án gồm 3 chương chính:**

Chương 1: Tổng quan về hệ thống tự động tổng hợp thông tin về dịch bệnh và các vấn đề liên quan

Chương 2: Phân tích và thiết kế hệ thống

Chương 3: Triển khai và xây dựng hệ thống

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG TỰ ĐỘNG TỔNG HỢP THÔNG TIN VỀ DỊCH BỆNH

Nội dung Chương 1 của đề án tập trung vào việc giới thiệu chung về hệ thống tổng hợp thông tin, khảo sát tổng quan về Scrapy framework và các vấn đề liên quan. Nội dung chương 1 sẽ làm cơ sở cho các nghiên cứu tiếp theo của đề án.

1.1. Giới thiệu chung về hệ thống tổng hợp thông tin

Trong mục này đề án sẽ khảo sát các khái niệm liên quan đến hệ thống tổng hợp thông tin, các hệ thống đã được triển khai trong thực tế, một số công nghệ được sử dụng, hình thức và đối tượng triển khai trong hệ thống tổng hợp thông tin. Từ đó thực hiện phân tích một số ưu điểm và hạn chế của hệ thống tổng hợp thông tin.

1.1.1. Khái niệm về hệ thống tổng hợp thông tin

Hệ thống tổng hợp thông tin là một cụm từ mô tả một loạt các công nghệ, quy trình và phương pháp được sử dụng để tổng hợp, xử lý và trình bày thông tin từ nhiều nguồn khác nhau một cách có tổ chức và hợp lý. Mục đích chính của hệ thống này là giúp người dùng thu thập và tiếp cận thông tin một cách hiệu quả [1].

Hệ thống tổng hợp thông tin thường bao gồm các thành phần sau[1]:

- Thu thập thông tin: Các công cụ và kỹ thuật được sử dụng để tự động hoặc thủ công thu thập thông tin từ các nguồn khác nhau như trang web, cơ sở dữ liệu,…

- Xử lý và phân tích thông tin: Dữ liệu thu thập được được xử lý và phân tích để trích xuất thông tin hữu ích, loại bỏ dữ liệu không cần thiết và tổ chức thông tin một cách có cấu trúc.

- Trình bày thông tin: Thông tin được trình bày một cách logic và dễ tiếp cận cho người dùng cuối thông qua các giao diện người dùng, báo cáo, biểu đồ, và các định dạng khác.

- Lưu trữ và quản lý dữ liệu: Dữ liệu được lưu trữ một cách an toàn và có thể được truy xuất dễ dàng khi cần thiết, đồng thời cũng được quản lý để đảm bảo tính nhất quán và bảo mật.

Hệ thống tổng hợp thông tin có thể được áp dụng trong nhiều lĩnh vực như công nghệ thông tin, y tế, kinh doanh, giáo dục, và nghiên cứu. Các ứng dụng cụ thể

có thể là các công cụ tìm kiếm, hệ thống quản lý tri thức, hệ thống gợi ý nội dung, và nhiều hơn nữa. Trong môi trường kinh doanh, hệ thống tổng hợp thông tin có thể giúp các doanh nghiệp nắm bắt thông tin về thị trường, đối thủ cạnh tranh, và xu hướng ngành công nghiệp để hỗ trợ quyết định chiến lược và phát triển kinh doanh.

1.1.2. Các hệ thống tổng hợp thông tin đã được triển khai trong thực tế

- Google Search: Google sử dụng thuật toán phức tạp để tổng hợp và hiển thị các kết quả tìm kiếm từ hàng tỷ trang web trên Internet.

- RSS Readers: Các ứng dụng RSS như Feedly cho phép người dùng tổng hợp và đọc các tin tức từ nhiều nguồn khác nhau trong một nền tảng duy nhất.

- Hệ thống quản lý kiến thức: Như Microsoft SharePoint hay Confluence của Atlassian, giúp tổ chức và chia sẻ thông tin nội bộ của doanh nghiệp.

- Hệ thống gợi ý nội dung cá nhân: Như Spotify, Netflix sử dụng dữ liệu cá nhân hóa để tổng hợp và gợi ý nội dung phù hợp với người dùng.

1.1.3. Một số công nghệ sử dụng trong hệ thống tổng hợp thông tin

- Thuật toán Machine Learning và AI: Được sử dụng để phân loại, phân tích và gợi ý thông tin dựa trên dữ liệu lịch sử và hành vi người dùng.

- Natural Language Processing (NLP): Sử dụng để hiểu và xử lý ngôn ngữ tự nhiên, giúp tổng hợp và trích xuất thông tin từ văn bản.

- Công nghệ Big Data: Được sử dụng để xử lý và phân tích lượng lớn dữ liệu từ nhiều nguồn khác nhau để tạo ra thông tin hữu ích.

- Web Scraping và Data Crawling: Sử dụng để tự động thu thập thông tin từ các trang web và nguồn dữ liệu trực tuyến khác.

1.1.4. Hình thức triển khai và đối tượng của hệ thống tổng hợp thông tin

- Hình thức triển khai của hệ thống [1]:

+ Ứng dụng web: Hệ thống tổng hợp thông tin có thể triển khai dưới dạng một ứng dụng web trực tuyến, cho phép người dùng truy cập thông tin từ bất kỳ nơi nào có kết nối internet.

+ Ứng dụng di động: Có thể phát triển ứng dụng di động cho các nền tảng như iOS và Android, giúp người dùng truy cập thông tin một cách thuận tiện trên điện thoại di động hoặc máy tính bảng.

+ Phần mềm trên máy tính: Hệ thống tổng hợp thông tin cũng có thể triển khai dưới dạng một ứng dụng độc lập trên máy tính, cho phép người dùng truy cập thông tin mà không cần kết nối internet.

- Đối tượng của hệ thống [1]:

+ Cá nhân: Người dùng cá nhân có thể sử dụng hệ thống tổng hợp thông tin để tìm kiếm thông tin về một chủ đề cụ thể, theo dõi tin tức hoặc nghiên cứu.

+ Doanh nghiệp: Các tổ chức và doanh nghiệp có thể triển khai hệ thống tổng hợp thông tin để thu thập và phân tích thông tin thị trường, cạnh tranh và xu hướng ngành công nghiệp để hỗ trợ quyết định chiến lược kinh doanh.

+ Cơ quan chính phủ và tổ chức phi chính phủ: Các cơ quan chính phủ và tổ chức phi chính phủ có thể sử dụng hệ thống tổng hợp thông tin để thu thập và phân tích dữ liệu từ nhiều nguồn khác nhau để đưa ra các chính sách và quyết định chính trị, xã hội và kinh tế.

+ Người tiêu dùng và công chúng: Người tiêu dùng và công chúng có thể sử dụng hệ thống tổng hợp thông tin để tìm kiếm thông tin về sản phẩm, dịch vụ, sự kiện và các chủ đề khác mà họ quan tâm.

1.1.5. Ưu điểm và hạn chế của các hệ thống tổng hợp thông tin

- Ưu điểm của hệ thống [1]:

+ Tăng hiệu suất tìm kiếm và tiếp cận thông tin: Hệ thống tổng hợp thông tin giúp người dùng tiết kiệm thời gian và công sức trong việc tìm kiếm và tiếp cận thông tin cần thiết.

+ Tổ chức thông tin hiệu quả: Cung cấp cách tổ chức thông tin có cấu trúc giúp người dùng dễ dàng tìm kiếm và sử dụng.

+ Tính cá nhân hóa: Các hệ thống có thể cung cấp thông tin được tùy chỉnh dựa trên sở thích và nhu cầu của người dùng.

- Hạn chế của hệ thống [1]:

+ Nguy cơ thông tin không chính xác hoặc thiếu trung thực: Các hệ thống tổng hợp thông tin có thể phụ thuộc vào nguồn thông tin không đáng tin cậy hoặc có thể bị ảnh hưởng bởi thông tin giả mạo.

+ Nguy cơ mất quyền riêng tư: Sử dụng dữ liệu cá nhân để tạo ra thông tin cá nhân hóa có thể gây lo ngại về quyền riêng tư.

+ Hạn chế về phạm vi và độ chi tiết: Các hệ thống tổng hợp thông tin có thể không thể tổng hợp hoặc hiển thị mọi thông tin một cách đầy đủ và chi tiết.

1.2. Xu hướng phát triển hệ thống tự động tổng hợp thông tin về dịch bệnh trên thế giới và tại Việt Nam.

1.2.1. Xu hướng phát triển hệ thống tổng hợp thông tin về dịch bệnh trên thế giới

Các quốc gia phát triển trên toàn cầu đã ứng dụng mạnh mẽ hệ thống tổng hợp thông tin về dịch bệnh để cung cấp thông tin, cập nhật tình hình dịch bệnh đang diễn ra cho quốc gia của mình.

*** Hệ thống tổng hợp thông tin về dịch bệnh tại Mỹ:**

Mỹ đã phát triển và triển khai nhiều hệ thống tổng hợp thông tin về dịch bệnh như CDC COVID Data Tracker và những ứng dụng di động như COVID Symptom Tracker và nhiều loại dịch bệnh như cúm, sốt xuất huyết và các bệnh truyền nhiễm khác.

Xu hướng phát triển tại Mỹ tập trung vào việc sử dụng trí tuệ nhân tạo và phân tích dữ liệu để dự đoán và quản lý dịch bệnh, đồng thời cung cấp thông tin cập nhật và hướng dẫn sức khỏe cho người dân.

*** Hệ thống tổng hợp thông tin về dịch bệnh tại Trung Quốc:**

Trung Quốc đã triển khai một loạt các hệ thống tổng hợp thông tin về dịch bệnh, được quản lý bởi cơ quan y tế và chính phủ địa phương. Trung Quốc có các hệ thống tổng hợp thông tin về nhiều loại dịch bệnh truyền nhiễm, từ cúm đến bệnh sốt xuất huyết.

Xu hướng phát triển tại Trung Quốc tập trung vào việc sử dụng công nghệ như trí tuệ nhân tạo và phân tích dữ liệu để theo dõi và kiểm soát các làn sóng dịch bệnh, đồng thời cung cấp các dịch vụ hỗ trợ và tư vấn sức khỏe trực tuyến.

*** Hệ thống tổng hợp thông tin về dịch bệnh tại Hàn Quốc:**

Hàn Quốc đã triển khai một hệ thống tổng hợp thông tin về dịch bệnh thông qua các ứng dụng di động như KCDC COVID-19 và các trang web của cơ quan y tế quốc gia.

Xu hướng phát triển tại Hàn Quốc tập trung vào việc sử dụng công nghệ để theo dõi và phản ứng nhanh chóng với các trường hợp nhiễm bệnh, cùng với việc cung cấp thông tin cập nhật và hướng dẫn sức khỏe cho cộng đồng.

*** Hệ thống tổng hợp thông tin về dịch bệnh tại các quốc gia khác:**

Nhiều quốc gia khác trên thế giới cũng đang phát triển các hệ thống tổng hợp thông tin về dịch bệnh truyền nhiễm và các vấn đề y tế cộng đồng khác, sử dụng các công nghệ và phương pháp tương tự như Mỹ, Trung Quốc và Hàn Quốc.

Xu hướng phát triển ở các quốc gia này thường tập trung vào việc tăng cường khả năng theo dõi, phát hiện và đáp ứng với các dịch bệnh thông qua sự tích hợp dữ liệu đa nguồn và sử dụng trí tuệ nhân tạo và phân tích dữ liệu.

1.2.2. Thực trạng phát triển và ứng dụng hệ thống tổng hợp thông tin về dịch bệnh tại Việt Nam

Tính đến thời điểm hiện tại, Việt Nam đã có sự phát triển đáng kể trong việc triển khai và ứng dụng các hệ thống tổng hợp thông tin về dịch bệnh. Dưới đây là một số điểm nổi bật về thực trạng phát triển và ứng dụng của hệ thống này tại Việt Nam:

Chính phủ và các cơ quan y tế tại Việt Nam đã triển khai và quản lý các hệ thống thông tin chính thức như trang web của Bộ Y tế và Ứng dụng tiêm chủng điện tử. Những hệ thống này cung cấp thông tin cập nhật về tình hình dịch bệnh, số liệu thống kê, hướng dẫn sức khỏe, và các biện pháp phòng chống cho người dân.

Nhiều ứng dụng di động được phát triển tại Việt Nam để cung cấp thông tin về dịch bệnh và hỗ trợ trong việc theo dõi sức khỏe cá nhân, như ứng dụng Bluezone và Vietnam Health Declaration. Các công nghệ như trí tuệ nhân tạo và phân tích dữ

liệu cũng được tích hợp vào các hệ thống này để cung cấp dự đoán và phản ứng nhanh chóng với tình hình dịch bệnh.

Việt Nam đã hợp tác với các tổ chức quốc tế như WHO và CDC để chia sẻ dữ liệu và kinh nghiệm trong việc phát triển và ứng dụng hệ thống tổng hợp thông tin về dịch bệnh. Sự hợp tác này giúp nâng cao khả năng phát triển và cập nhật thông tin, đồng thời đảm bảo tính chính xác và tin cậy của dữ liệu.

Hệ thống tổng hợp thông tin về dịch bệnh tại Việt Nam không chỉ cung cấp thông tin cho người dân mà còn được sử dụng trong quản lý và phản ứng với dịch bệnh từ cấp cao đến cấp cơ sở. Các thông tin từ hệ thống này được sử dụng để đưa ra các quyết định chính sách và các biện pháp phòng chống tại cộng đồng.

Mặc dù đã có sự phát triển, nhưng vẫn còn thách thức trong việc nâng cao tính chính xác và khả năng phản ứng của các hệ thống này, đặc biệt là trong bối cảnh các biến thể mới của virus và tình hình dịch bệnh biến động liên tục.Việt Nam cần tiếp tục cải thiện hệ thống, đào tạo nhân lực chuyên môn, và tăng cường sự hợp tác cả trong nước và quốc tế để đối phó hiệu quả với các thách thức của dịch bệnh.

1.3. Các công nghệ sử dụng

Trong mục này đề án sẽ đi sâu và chi tiết về Scrapy framework, đồng thời trình bày một số nội dung về các công nghệ sử dụng liên quan để xây dựng hệ thống tự động tổng hợp thông tin về dịch bệnh.

1.3.1. Scrapy framework

Hệ thống tự động tổng hợp thông tin về dịch bệnh sử dụng công nghệ xử lý, thu thập dữ liệu phổ biến, mạnh mẽ Scrapy Framework để thu thập dữ liệu về dịch bệnh, cụ thể là dịch sốt xuất huyết.

1.3.1.1. Lịch sử hình thành

Scrapy Framework là một framework mã nguồn mở được phát triển bởi Pablo Hoffman vào năm 2008 [6]. Nó được thiết kế nhằm mục đích thu thập dữ liệu từ trang web một cách tự động và linh hoạt. Cách tiếp cận này đã giúp cho việc trích xuất thông tin từ web trở nên dễ dàng và hiệu quả hơn cho các nhà phân tích dữ liệu và nhà phát triển ứng dụng web.

Lịch sử hình thành của Scrapy [6] có thể được phân thành các giai đoạn chính:

- Giai đoạn sơ khai (2008 - 2010): Scrapy được tạo ra bởi Pablo Hoffman khi ông đang làm việc tại Scrapinghub, một công ty chuyên cung cấp dịch vụ web scraping. Giai đoạn này chứng kiến việc phát triển và hoàn thiện các tính năng cơ bản của Scrapy là xây dựng một cơ chế linh hoạt để trích xuất dữ liệu từ các trang web.

- Phát triển và mở rộng (2010 - 2015): Scrapy nhanh chóng trở thành một công cụ phổ biến trong cộng đồng web scraping và crawling. Các nhà phát triển đã đóng góp vào mã nguồn mở của Scrapy, cung cấp các tính năng mới và cải thiện hiệu suất của framework. Scrapy đã đạt được sự ổn định và uy tín trong việc xử lý việc thu thập dữ liệu trên web, và được sử dụng rộng rãi trong các dự án nghiên cứu và thương mại.

- Sự phát triển tiếp theo (2015 - nay): Scrapy tiếp tục phát triển và cập nhật, với việc ra mắt các phiên bản mới và cập nhật định kỳ. Cộng đồng người dùng Scrapy ngày càng phát triển, sự đóng góp của các nhà phát triển và nhà nghiên cứu từ nhiều quốc gia khác nhau. Scrapy vẫn duy trì vị thế là một trong những framework hàng đầu cho việc thu thập dữ liệu trên web, với sự ổn định, linh hoạt và hiệu suất cao.

Như vậy, qua các giai đoạn phát triển khác nhau, Scrapy framework đã trở thành một công cụ mạnh mẽ và phổ biến trong lĩnh vực web scraping và crawling, giúp cho việc thu thập dữ liệu trên web trở nên dễ dàng và hiệu quả hơn.

1.3.1.2. Công nghệ sử dụng trong Scrapy framework

Scrapy framework sử dụng một số công nghệ chính để cung cấp khả năng thu thập dữ liệu hiệu quả từ các trang web. Dưới đây là một số công nghệ chính được sử dụng trong Scrapy [12]:

- Python: Scrapy được viết bằng ngôn ngữ lập trình Python. Python là một ngôn ngữ lập trình mạnh mẽ, dễ đọc và dễ hiểu, được sử dụng rộng rãi trong lĩnh vực phát triển web và khoa học dữ liệu. Sự linh hoạt của Python đã giúp Scrapy trở thành một công cụ phổ biến cho việc thu thập dữ liệu từ web.

- Twisted: Scrapy sử dụng Twisted, một thư viện mạng bất đồng bộ trong Python, để xử lý các yêu cầu web và truy xuất dữ liệu từ các trang web. Twisted cung

cấp khả năng xử lý đa luồng và không đồng bộ, giúp Scrapy trở nên hiệu quả và linh hoạt trong việc thu thập dữ liệu từ nhiều trang web cùng một lúc.

- XPath và CSS Selectors: Scrapy hỗ trợ việc trích xuất dữ liệu từ các trang web thông qua việc sử dụng XPath và CSS Selectors. Đây là hai ngôn ngữ truy vấn mạnh mẽ được sử dụng để định vị và lấy dữ liệu từ các thành phần HTML trên trang web. Việc hỗ trợ cả hai ngôn ngữ này giúp cho Scrapy linh hoạt và dễ dàng trong việc trích xuất dữ liệu từ các trang web có cấu trúc phức tạp.

- Middleware: Scrapy cung cấp middleware cho phép người dùng thực hiện các xử lý trước và sau khi yêu cầu được gửi đến trang web mục tiêu. Middleware cho phép tùy chỉnh và mở rộng khả năng của Scrapy bằng cách thêm các chức năng tùy chỉnh, như xử lý lỗi, lọc dữ liệu, và ghi log.

- Pipelines: Scrapy cung cấp pipelines để xử lý dữ liệu thu thập sau khi nó được trích xuất từ các trang web. Pipelines cho phép bạn thực hiện các xử lý như lọc dữ liệu, xử lý lỗi, và lưu trữ vào cơ sở dữ liệu. Pipelines giúp tổ chức và xử lý dữ liệu thu thập một cách linh hoạt và hiệu quả.

1.3.1.3. Các tính năng của Scrapy framework

Các tính năng chính của Scrapy framework [12] giúp Scrapy trở thành một công cụ mạnh mẽ và linh hoạt cho việc thu thập dữ liệu từ các trang web một cách tự động và hiệu quả:

- Crawling đa luồng (Multithreaded Crawling): Scrapy hỗ trợ crawling đa luồng, cho phép truy cập nhiều trang web cùng một lúc. Điều này giúp tăng tốc độ thu thập dữ liệu và làm cho quá trình thu thập trở nên hiệu quả hơn.

- Cấu trúc cơ bản: Scrapy cung cấp cấu trúc cơ bản và linh hoạt, giúp bạn dễ dàng tổ chức mã nguồn và quản lý các quy trình thu thập dữ liệu. Scrapy thúc đẩy việc tuân thủ nguyên tắc thiết kế SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion).

- Thích ứng với thay đổi trang web: Scrapy có khả năng thích ứng với các thay đổi trong cấu trúc của trang web mục tiêu. Điều này giúp duy trì hiệu suất của quá trình thu thập dữ liệu khi các trang web mục tiêu thay đổi cấu trúc hoặc định dạng.

- Định vị dữ liệu: Scrapy hỗ trợ XPath và CSS Selectors, cho phép dễ dàng định vị và trích xuất dữ liệu từ các trang web. Điều này giúp cho việc trích xuất dữ liệu từ các trang web trở nên đơn giản và linh hoạt hơn.

- Middleware và Pipelines:Scrapy cung cấp middleware và pipelines cho phép thực hiện các xử lý trước và sau khi thu thập dữ liệu. Middleware cho phép thêm các chức năng tùy chỉnh như xử lý lỗi hoặc ghi log. Pipelines cho phép xử lý dữ liệu sau khi nó được trích xuất, như lọc dữ liệu hoặc lưu trữ vào cơ sở dữ liệu.

- Đa nền tảng: Scrapy có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS, cho phép bạn phát triển và triển khai ứng dụng của mình trên nhiều môi trường khác nhau.

- Thiết kế modul và mở rộng: Scrapy được thiết kế với kiến trúc modul và dễ dàng mở rộng. Người dùng có thể thêm các plugin hoặc module mới để mở rộng tính năng của Scrapy theo nhu cầu của dự án.

1.3.1.4. Các thành phần của Scrapy framework

Scrapy Framework được xây dựng dựa trên một số thành phần chính để hỗ trợ việc thu thập dữ liệu từ các trang web một cách hiệu quả [12]. Dưới đây là các thành phần chính của Scrapy:

- Spider (Nhện): Spider là thành phần chính của Scrapy, nó định nghĩa cách mà Scrapy sẽ thu thập dữ liệu từ các trang web. Mỗi spider đại diện cho một trang web cụ thể và chứa các quy tắc và luật để trích xuất dữ liệu. Spider định nghĩa cách các URL sẽ được truy cập, cách dữ liệu sẽ được trích xuất từ các trang web, và cách Scrapy sẽ xử lý dữ liệu đã trích xuất.

- Downloader (Bộ tải về): Downloader là thành phần được sử dụng để tải về các trang web từ các URL đã được Spider xác định. Downloader thực hiện việc gửi yêu cầu HTTP đến máy chủ web và nhận các trang web tương ứng. Nó cũng xử lý việc tuân thủ các quy tắc robots.txt và các quy định về tần suất truy cập trang web.

- Scheduler (Lập lịch): Scheduler là thành phần của Scrapy quản lý hàng đợi các URL cần được tải về. Nó đảm bảo rằng Scrapy không gửi quá nhiều yêu cầu cùng một lúc đến cùng một trang web và tuân thủ các quy tắc về tần suất truy cập. Scheduler

cũng quản lý việc tạo các yêu cầu mới dựa trên các URL đã được trích xuất từ các trang web trước đó.

- Item Pipeline (Ống dữ liệu): Item Pipeline là một loạt các xử lý mà dữ liệu trích xuất từ các trang web sẽ đi qua trước khi được lưu trữ hoặc sử dụng. Item Pipeline có thể được sử dụng để lọc, kiểm tra tính hợp lệ và lưu trữ dữ liệu, cũng như để thực hiện các tác vụ khác như xử lý lỗi và ghi log.

- Middleware: Middleware là một loạt các bộ lọc được áp dụng cho cả yêu cầu đến và phản hồi từ trang web. Chúng có thể được sử dụng để thực hiện các xử lý trước hoặc sau khi các yêu cầu được gửi và nhận. Middleware có thể được sử dụng để thay đổi hoặc mở rộng hành vi của Scrapy, như thêm thông tin HTTP header vào yêu cầu, xử lý lỗi, hoặc chuyển hướng URL.

- Engine (Bộ máy): Engine là trái tim của Scrapy, điều phối hoạt động của tất cả các thành phần khác. Nó quản lý việc gửi yêu cầu, quản lý hàng đợi URL, điều khiển quá trình crawl và phối hợp các Middleware, Spider và Scheduler.

1.3.1.5. Luồng dữ liệu của Scrapy framework

Luồng dữ liệu trong Scrapy Framework [12] bao gồm một loạt các bước và thành phần mà dữ liệu đi qua từ khi bắt đầu thu thập cho đến khi được xử lý và lưu trữ. Dưới đây là mô tả về luồng dữ liệu trong Scrapy:

- Bước 1: Spider khởi động:

Quá trình bắt đầu với việc khởi động một hoặc nhiều Spider. Mỗi Spider được xác định để thu thập dữ liệu từ một số trang web cụ thể. Spider gửi yêu cầu đến Scheduler để bắt đầu quá trình crawl.

- Bước 2: Scheduler quản lý hàng đợi URL:

Scheduler nhận các yêu cầu từ Spider và quản lý hàng đợi các URL cần được tải về. Nó đảm bảo rằng các yêu cầu được gửi đi một cách hợp lý và tuân thủ các quy tắc về tần suất truy cập.

- Bước 3: Downloader lấy dữ liệu:

Downloader nhận các yêu cầu từ Scheduler và tải về các trang web tương ứng. Nó gửi yêu cầu HTTP đến máy chủ web và nhận phản hồi chứa dữ liệu HTML của trang web.

- Bước 4: Spider trích xuất dữ liệu:

Spider nhận dữ liệu từ Downloader và bắt đầu quá trình trích xuất dữ liệu từ các trang web. Spider sử dụng các quy tắc và luật đã được định nghĩa trước để trích xuất thông tin cần thiết từ dữ liệu HTML như các phần tử, văn bản, hoặc hình ảnh.

- Bước 5: Xử lý dữ liệu với Item Pipeline:

Dữ liệu trích xuất từ các trang web được đưa qua các Item Pipeline. Item Pipeline là một loạt các xử lý mà dữ liệu sẽ đi qua trước khi được lưu trữ hoặc sử dụng. Các xử lý này có thể bao gồm lọc, kiểm tra tính hợp lệ, và lưu trữ dữ liệu vào cơ sở dữ liệu.

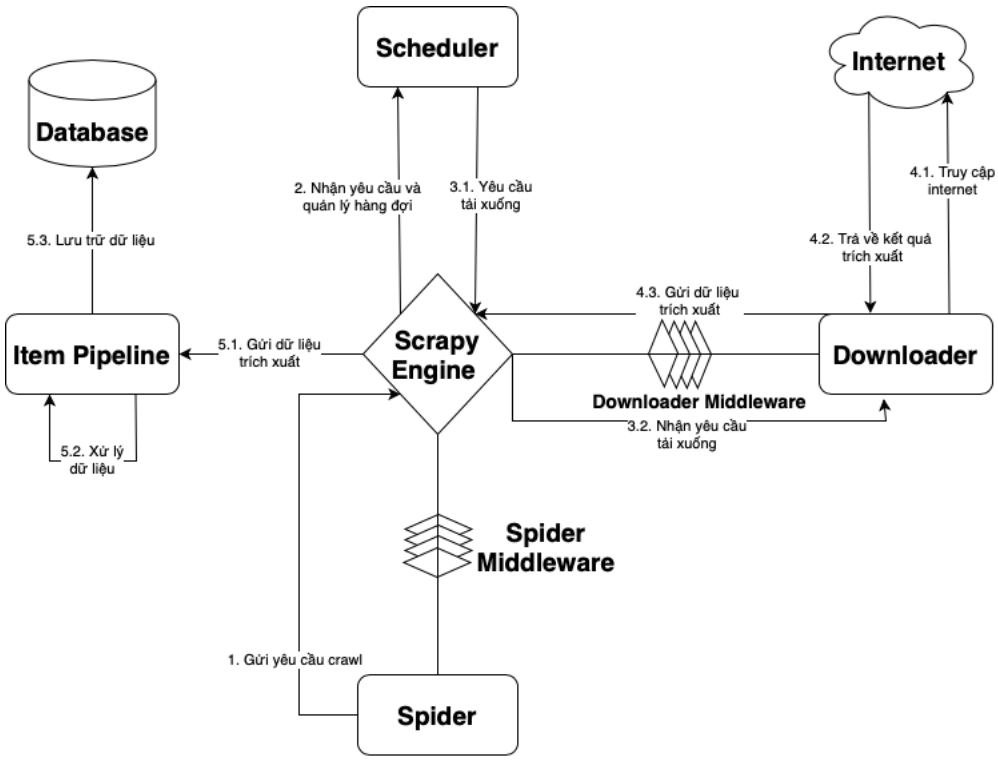
- Bước 6: Lặp lại quá trình:

Quá trình này lặp lại cho đến khi tất cả các URL trong hàng đợi được xử lý hoàn thành. Mỗi lần lặp lại, Spider có thể thu thập dữ liệu mới từ các trang web và đưa vào Item Pipeline để xử lý.

- Bước 7: Kết thúc quá trình:

Khi tất cả các URL đã được xử lý hoặc khi quá trình thu thập dữ liệu được kích hoạt dừng lại, quá trình crawl kết thúc. Scrapy có thể xuất ra dữ liệu đã được trích xuất vào các định dạng như JSON hoặc CSV, hoặc lưu trữ vào cơ sở dữ liệu để sử dụng cho mục đích sau này.

Hình 1.1 dưới đây mô tả luồng dữ liệu trong Scrapy framework:



Hình 1.1: Mô tả luồng dữ liệu trong Scrapy framework

1.3.2. Kiến trúc tiểu dịch vụ

Kiến trúc tiểu dịch vụ (Microservices[8]) là một kiểu kiến trúc mà cấu trúc hệ thống như một tập các dịch vụ. Vì vậy, nó giúp giảm thiểu quá trình phức tạp hóa trong các hệ thống lớn, như thuật toán chia để trị (divide and conquer), kiến trúc này chia hệ thống thành các phần nhỏ, độc lập để có thể dễ dàng đóng gói, triển khai, quản lý, kiểm thử, nâng cấp, bảo trì. Vì vậy sẽ tiết kiệm thời gian phát triển cũng như là bảo trì. Trong kiến trúc tiểu dịch vụ, các ứng dụng dịch vụ sẽ giao tiếp với nhau thông qua thủ tục gọi từ xa (Remote Procedure Call – RPC) hay các giao diện lập trình ứng dụng (Application Programming Interface – API). Bên cạnh đó, kiến trúc này cũng có một vài nhược điểm do nó là hệ thống phân tán: hệ thống có thể gặp sự cố kết nối chậm do các dịch vụ giao tiếp với nhau qua RPC hay API, hay nếu ứng dụng không đủ lớn thì việc sử dụng kiến trúc microservices sẽ phản tác dụng, làm chậm và tốn nhân lực phát triển.

1.3.3. Ngôn ngữ Java, Spring framework

Back-end của hệ thống tự động tổng hợp thông tin về dịch bệnh được phát triển dựa trên ngôn ngữ Java[4] là một trong những ngôn ngữ lập trình hướng đối tượng. Nó là một nền tảng máy tính được sử dụng nhiều trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động. Chương trình Java có thể triển khai trên mọi nền tảng khác nhau thông qua một môi trường thực thi nếu nó thích hợp hỗ trợ nền tảng đó.

Hệ thống back-end kết hợp sử dụng Spring framework[10]. Spring có rất nhiều các dự án con, giúp cho việc xây dựng hệ thống một cách dễ dàng: Spring MVC (thiết kế dành cho việc xây dựng web), Spring Security (cung cấp các cơ chế xác thực, phân quyền), Spring Boot (giúp phát triển, chạy ứng dụng một cách nhanh chóng), Spring Data (cung cấp các công nghệ để truy cập dữ liệu), Spring Cloud (cung cấp các công cụ để phát triển hệ thống phân tán),…

1.3.4. Ngôn ngữ TypeScript, thư viện Redux

Front-end của hệ thống được xây dựng dựa trên ngôn ngữ TypeScript[3] là một dự án mã nguồn mở được phát triển bởi Microsoft, được xây dựng dựa trên Javascript, một trong những công nghệ lõi của web, công cụ được sử dụng nhiều nhất trên thế giới, bằng cách thêm các định nghĩa kiểu tĩnh và lớp hướng đối tượng. TypeScript có thể sử dụng để phát triển các ứng dụng chạy ở máy khách và máy chủ

Kết hợp với việc sử dụng Redux[2] là một thư viện độc lập, có thể sử dụng với bất kì khung giao diện (UI layer) hay framework nào, dùng để quản lý trạng thái khi mà ứng dụng Javascrip ngày càng trở nên phức tạp. React với DOM ảo (Virtual DOM) giúp cho ứng dụng hoạt động nhanh hơn, được đánh giá cao về tốc độ ngang với VueJS.

1.3.5. Ngôn ngữ Python, Flask Framework

Trong các công nghệ thu thập dữ liệu đang có hiện nay, Python sở hữu Scrapy Framework là một mã nguồn mở được dùng để thu thập dữ liệu từ các trang web với một tốc độ cao. Nó cũng có thể trích xuất dữ liệu từ API, sử dụng trong nhiều mục

đích, từ khai thác dữ liệu đến giám sát và kiểm thử tự động. Vì vậy, trong kiến trúc tiểu dịch vụ, hệ thống chứa một thực thể dịch vụ được triển khai bằng Python[7].

Thực thể dịch vụ này sử dụng Flask framework để tạo ra một ứng dụng web, giao tiếp với các ứng dụng web của thực thể dịch vụ triển khai bằng Java đã mô tả ở phía trên. Flask[5] là một Web Framework rất nhẹ của Python, giúp lập trình viên dễ dàng tạo ra một website nhỏ, song nó cũng dễ dàng có thể mở rộng để triển khai những ứng dụng web lớn hơn.

1.3.6. Hệ quản trị cơ sở dữ liệu MySQL

Hệ thống tự động tổng hợp thông tin về dịch bệnh lưu trữ dữ liệu về thông tin người dùng, lịch sử thu thập dữ liệu của quản trị viên thông qua MySQL[9]. Nó là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở (RDBMS) dựa trên ngôn ngữ truy vấn có cấu trúc (SQL) được phát triển, phân phối và hỗ trợ bởi tập đoàn Oracle. MySQL chạy trên hầu hết tất cả các nền tảng, bao gồm cả Linux , UNIX và Windows. SQL là ngôn ngữ phổ biến nhất để thêm, truy cập và quản lý nội dung trong cơ sở dữ liệu. Nó được chú ý nhất vì khả năng xử lý nhanh, độ tin cậy đã được chứng minh, dễ sử dụng và linh hoạt.

1.3.7. Elasticsearch

Ngoài lưu trữ dữ liệu qua MySQL, hệ thống lưu trữ dữ liệu bệnh viện, bài báo về dịch bệnh trên Elasticsearch [11] là một công cụ tìm kiếm (search-engine) rất mạnh mẽ, nó cũng có thể coi là một "document oriented database", nó chứa dữ liệu giống như một cơ sở dữ liệu và thực hiện tìm kiếm trên những dữ liệu đó. Elasticsearch được viết bằng Java, hoạt động như một cloud server theo cơ chế RESTful. Các dữ liệu được lưu vào Elasticsearch đều được đánh chỉ mục (index), chính vì vậy nó rất thích hợp để tìm kiếm trong các trường hợp: tìm kiếm văn bản thông thường, tìm kiếm gần đúng và dữ liệu có cấu trúc, tổng hợp dữ liệu, tìm kiếm theo tọa độ, tìm kiếm với dữ liệu lớn. Ngoài ra, elasticsearch tìm dữ liệu rất nhanh chóng, mạnh mẽ dựa trên Apache Lucene (near-realtime searching); có khả năng phân tích dữ liệu; hỗ trợ Structured Query DSL, cung cấp việc đặc tả câu truy vấn phức tạp một cách cụ thể và rõ ràng bằng JSON.

1.4. Kết luận chương 1

Chương 1 của đề án đã tiến hành một khảo sát toàn diện về hệ thống tổng hợp thông tin, Scrapy framework và các vấn đề liên quan. Đề án đã khảo sát xu hướng phát triển hệ thống tổng hợp thông tin về dịch bệnh trên toàn cầu cũng như tình hình phát triển và ứng dụng của hệ thống tổng hợp thông tin về dịch bệnh tại Việt Nam. Đề án cũng nghiên cứu về Scrapy framework và các công nghệ để triển khai hiệu quả hệ thống tự động tổng hợp thông tin về dịch bệnh trong thực tế.

Dựa vào nội dung chương 1, các vấn đề liên quan đến hệ thống tự động tổng hợp thông tin về dịch bệnh sẽ được phân tích và thiết kế chi tiết trong chương 2.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Trong chương 2, tác giả phân tích và thiết kế hệ thống tự động tổng hợp thông tin về dịch bệnh, xác định tổng quan hệ thống, đặc tả chi tiết và biểu đồ các ca sử dụng, xây dựng kiến trúc hệ thống và mẫu đặc tả dữ liệu để trích xuất thông tin về dịch bệnh cụ thể là dịch sốt xuất huyết.

2.1. Tổng quan về hệ thống

Do nhu cầu có thể sử dụng mọi lúc mọi nơi nên hệ thống sẽ được viết trên nền tảng web. Hệ thống được phân tích và xây dựng với mục tiêu có thể hỗ trợ người dân trong việc tìm bệnh viện, tìm các tin tức liên quan đến dịch bệnh, giúp quản trị viên quản lý cũng như là thu thập dữ liệu dễ dàng. Vì phạm vi triển khai là trên cả nước Việt Nam nên hệ thống cần xử lý một lượng lớn các yêu cầu (request).

Về giao diện: Hệ thống sẽ cung cấp các cửa sổ cần thiết để người dùng tương tác, màu sắc mẫu nhập đơn giản dễ dàng nhận diện.

Về thao tác: Hệ thống cung cấp các thao tác đơn giản, tìm kiếm bệnh viện, các bài báo cho người dùng, cung cấp các công cụ đa dạng để quản trị viên quản lý nội dung trang web và thu thập dữ liệu.

2.1.1. Yêu cầu chức năng

Hệ thống có các chức năng chính sau:

Chức năng tìm kiếm: Người dùng có thể tìm kiếm bệnh viện theo tên, địa chỉ, số điện thoại, website, theo khoa và các dịch vụ khám bệnh. Hay tìm kiếm các bài báo theo tên bệnh, bệnh viện, nội dung bài báo…

Chức năng quản lý hệ thống: Quản trị viên có quyền quản lý dữ liệu về bệnh viện, bài báo, quản lý người dùng, cấp quyền quản trị viên cho người dùng. Hệ thống cho phép quản trị viên có thể tìm kiếm, xem chi tiết, sửa, xóa, thêm thông tin bệnh viện, bài báo, tìm kiếm, xem thông tin người dùng.

Chức năng hỗ trợ thu thập dữ liệu: Quản trị viên có thể thu thập dữ liệu từ các trang web tin cậy khác.

2.1.2. Yêu cầu phi chức năng

Hệ thống cũng phải đáp ứng yêu cầu phi chức năng như sau:

Yêu cầu về cấu hình tương thích: Hệ thống cần phải tương thích với hầu hết các trình duyệt web phổ biến hiện nay như chrome, firefox, opera mini...với các phiên bản khác nhau của trình duyệt mấy năm gần nay.

Yêu cầu về sự phản hồi: Thời gian phản hồi đảm bảo không quá 10 giây và việc cập nhật những hoạt động nhận diện được lên ứng dụng không quá 1 phút.

Yêu cầu về phần cứng: Hệ thống cần phải hoạt động bình thường trên tất cả máy tính có cầu hình trung bình trong vòng 4 năm trở lại đây.

Yêu cầu về an ninh: Hệ thống cần phải đảm bảo dữ liệu được bảo mật đặc biệt là các thông tin nhạy cảm như thông tin cá nhân của người dùng .

2.2. Xác định danh sách tác nhân và ca sử dụng

Hệ thống có 4 tác nhân sau đây:

Quản trị viên (admin): Là người chịu trách nhiệm quản lý hệ thống về mặt nội dung và kỹ thuật.

Người vãng lai (guest): Là người truy cập hệ thống nhưng chưa có tài khoản.

Người dùng (user): Là người đã đăng ký tài khoản và có thể sử dụng thêm một số chức năng của hệ thống mà người vãng lai không có.

Các hệ thống liên kết ngoài (third-party system): Là những hệ thống ngoài có kết nối với hệ thống.

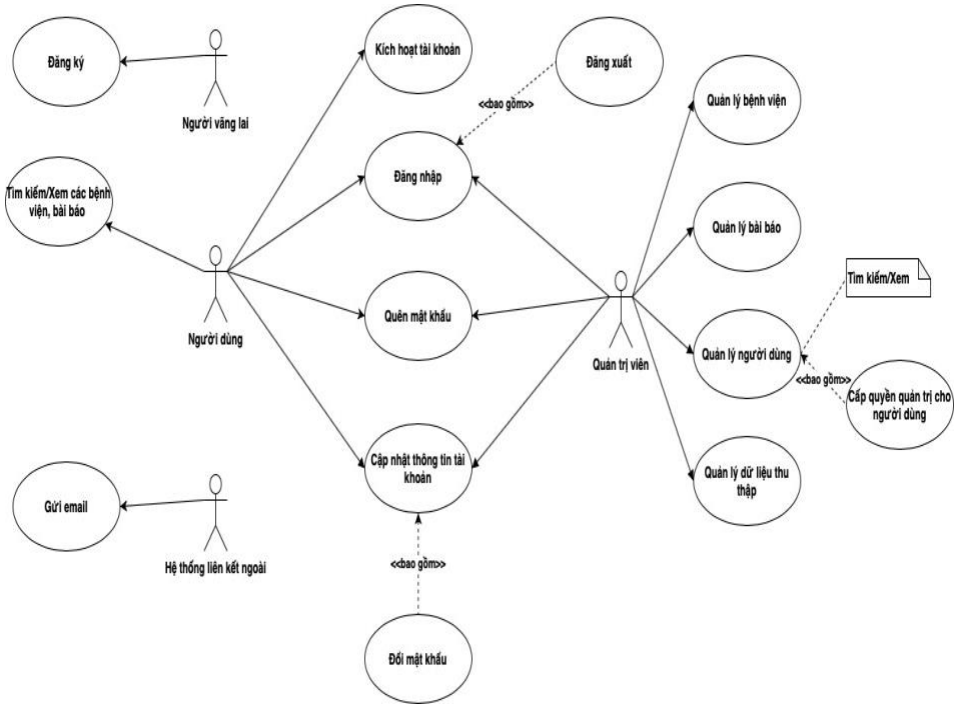
Hệ thống có 12 ca sử dụng chính, chia làm 3 nhóm:

Nhóm 1 là các ca sử dụng liên quan đến tài khoản bao gồm: đăng ký, đăng nhập, quên mật khẩu, đổi mật khẩu, cập nhật thông tin tài khoản, đăng xuất.

Nhóm 2 là các ca sử dụng dùng cho người dùng bao gồm: tìm kiếm/xem bệnh viện, tìm kiếm/xem bài báo.

Nhóm 3 là các ca sử dụng dùng cho quản trị viên bao gồm: quản lý bệnh viện, quản lý bài báo, quản lý người dùng, quản lý dữ liệu thu thập.

Quan hệ giữa các tác nhân và ca sử dụng được mô tả ở hình 2.1 dưới đây:



Hình 2.1: Mô hình quan hệ các tác nhân và ca sử dụng

2.3. Đặc tả chi tiết và biểu đồ ca sử dụng

2.3.1. Đăng ký

Mã usecase: UC-A1 **Mã usecase liên quan:** Không

Mô tả ngắn gọn: Để sử dụng ứng dụng trước tiên tác nhân cần đăng ký tài khoản, ca sử dụng mô tả thao tác tác nhân đăng ký sử dụng.

Các tác nhân: Người vãng lai

Tiền điều kiện: Không

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn chức năng đăng ký.

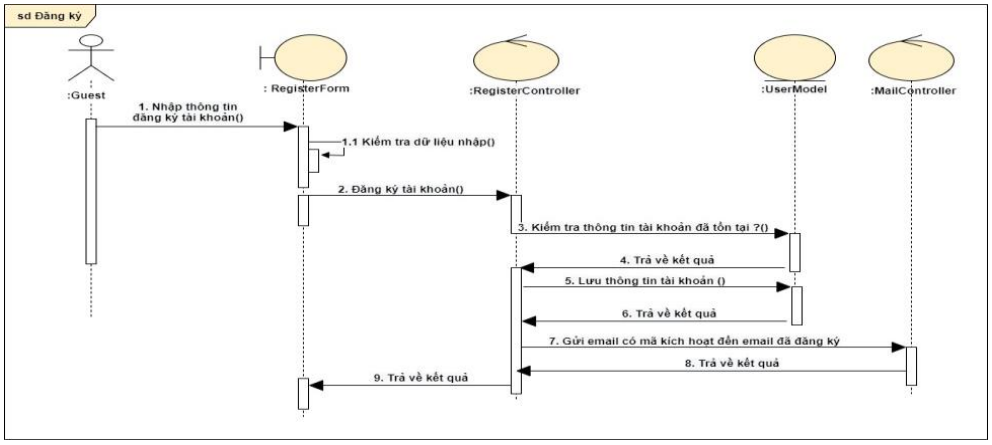
1. Tác nhân nhập các thông tin để đăng ký tài khoản.

2. Hệ thống lưu lại thông tin tác nhân đã đăng ký vào cơ sở dữ liệu, đồng thời gửi email chứa mã kích hoạt tài khoản đến email vừa đăng ký.

Luồng rẽ nhánh: Ở bước 1, nếu tác nhân nhập sai định dạng, hoặc thông tin tài khoản đã tồn tại thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại dữ liệu.

Hậu điều kiện: Hệ thống thông báo đăng ký tài khoản thành công. Tác nhân được đưa đến màn hình xác thực tài khoản.

Biểu đồ tuần tự của ca sử dụng đăng ký được mô tả như hình 2.2 dưới đây:



Hình 2.2: Biểu đồ tuần tự của ca sử dụng đăng ký

2.3.2. Kích hoạt tài khoản

Mã usecase: UC-A2 **Mã usecase liên quan:** UC-A1

Mô tả ngắn gọn: Để sử dụng tài khoản tác nhân cần xác thực email để kích hoạt tài khoản trước, ca sử dụng mô tả thao tác tác nhân kích hoạt tài khoản.

Các tác nhân: Người dùng, quản trị viên.

Tiền điều kiện: Tác nhân đã đăng ký tài khoản tại hệ thống

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân xác thực tài khoản.

1. Nhập mã xác thực.

    1.1. Tác nhân nhập mã xác thực được gửi đến email đăng ký tài khoản để xác thực tài khoản.

    1.2. Hệ thống kiểm tra mã xác thực và kích hoạt tài khoản cho tác nhân.

2. Gửi lại mã xác thực.

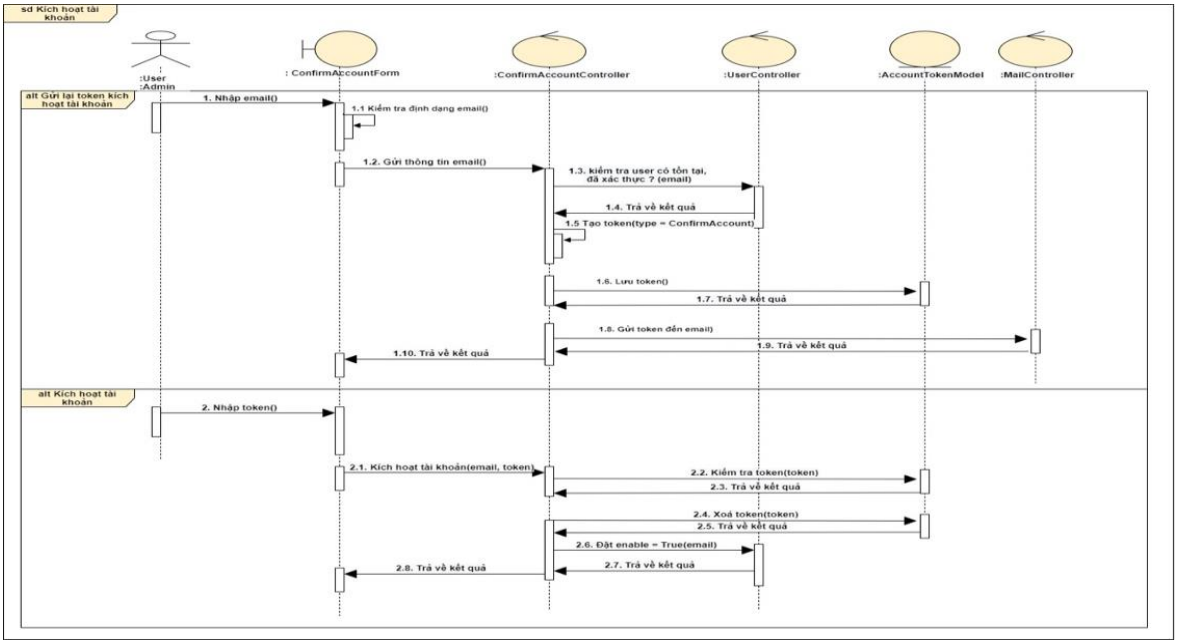
    2.1. Tác nhân chọn chức năng gửi lại mã xác thực.

    2.2. Hệ thống gửi mã xác thực mới đến email của tác nhân.

Luồng rẽ nhánh: Ở bước 1.2, nếu tác nhân nhập sai mã xác thực thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

Hậu điều kiện: Hệ thống thông báo tác nhân kích hoạt tài khoản thành công.

Biểu đồ tuần tự của ca sử dụng kích hoạt tài khoản được mô tả như hình 2.3.



Hình 2.3: Biểu đồ tuần tự của ca sử dụng kích hoạt tài khoản

2.3.3. Quên mật khẩu

Mã usecase: UC-A3 **Mã usecase liên quan:** UC-A1

Mô tả ngắn gọn: Ca sử dụng mô tả thao tác tác nhân quên mật khẩu.

Các tác nhân: Người dùng, quản trị viên

Tiền điều kiện: Tác nhân đã đăng ký tài khoản tại hệ thống

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn quên mật khẩu.

1. Tác nhân nhập nhập thông tin theo các trường trên mẫu quên mật khẩu.

2. Hệ thống sẽ gửi mã xác thực vào email đăng ký tài khoản.

3. Tác nhân nhập mã xác thực và mật khẩu mới để đổi mật khẩu.

4. Lưu mật khẩu mới vào cơ sở dữ liệu, đăng xuất phiên đăng nhập cũ.

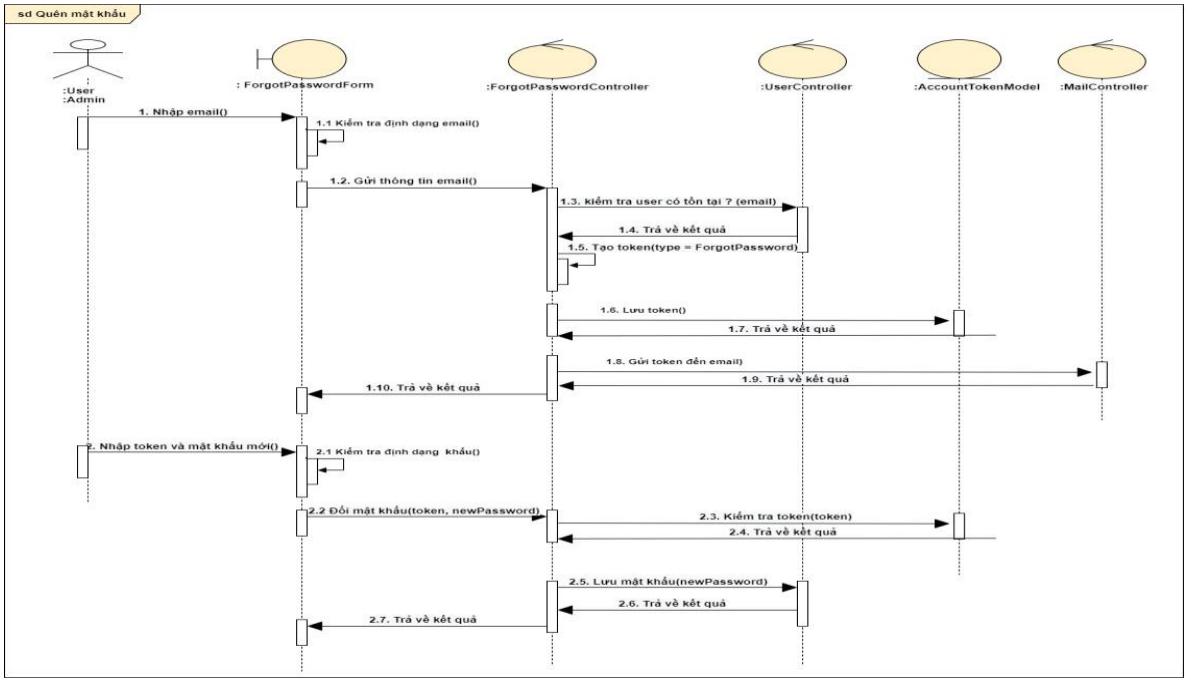
Luồng rẽ nhánh:

- Ở bước 1, Nếu tác nhân nhập sai thông tin tài khoản thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

- Ở bước 2, nếu tác nhân nhập sai mã xác thực hoặc sai định dạng mật khẩu mới thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

Hậu điều kiện: Hệ thống thông báo tác nhân thay đổi mật khẩu thành công, chuyển đến màn hình đăng nhập.

Biểu đồ tuần tự của ca sử dụng quên mật khẩu được mô tả như hình 2.4.



Hình 2.4: Biểu đồ tuần tự của ca sử dụng quên mật khẩu

2.3.4. Đăng nhập

Mã usecase: UC-A4 **Mã usecase liên quan:** UC-A2

Mô tả ngắn gọn: Để sử dụng ứng dụng, tác nhân cần đăng nhập vào hệ thống, ca sử dụng mô tả thao tác tác nhân đăng nhập.

Các tác nhân: Người vãng lai

Tiền điều kiện: Không

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn chức năng đăng nhập.

1. Tác nhân nhập email, mật khẩu.

2. Hệ thống sẽ xác thực và phân quyền thông tin đăng nhập của tác nhân.

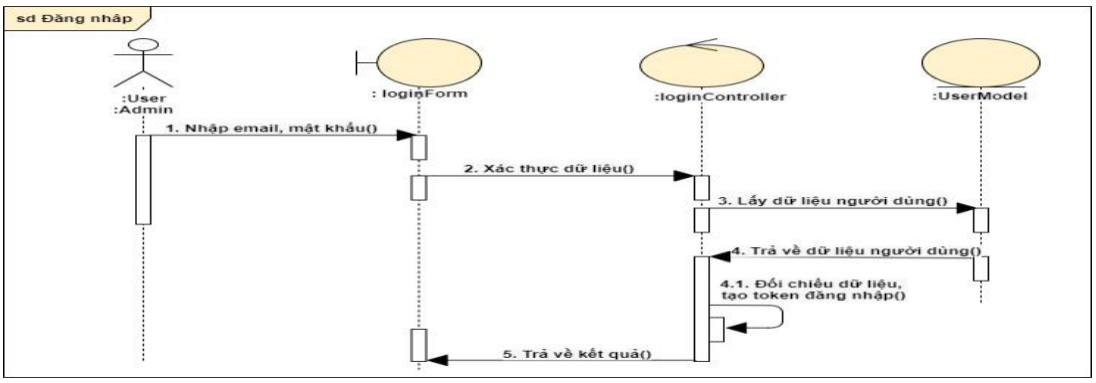
Luồng rẽ nhánh:

- Ở bước 1, nếu tác nhân nhập sai thông tin tài khoản, mật khẩu thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

- Nếu tác nhân chưa xác thực tài khoản, hệ thống sẽ chuyển đến trang xác thực tài khoản để tác nhân xác thực tài khoản trước khi đăng nhập

Hậu điều kiện: Tác nhân là quản trị viên sẽ chuyển đến trang ứng dụng của quản trị viên. Tác nhân là người dùng sẽ chuyển đến trang ứng dụng của người dùng.

Biểu đồ tuần tự của ca sử dụng đăng nhập được mô tả như hình 2.5 dưới đây:



Hình 2.5: Biểu đồ tuần tự của ca sử dụng đăng nhập

2.3.5. Đổi mật khẩu

Mã usecase: UC-A5 **Mã usecase liên quan:** UC-A4

Mô tả ngắn gọn: Ca sử dụng mô tả thao tác tác nhân đổi mật khẩu.

Các tác nhân: Người dùng, quản trị viên

Tiền điều kiện: Tác nhân đã đăng nhập tài khoản thành công tại hệ thống

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn chức năng đổi mật khẩu.

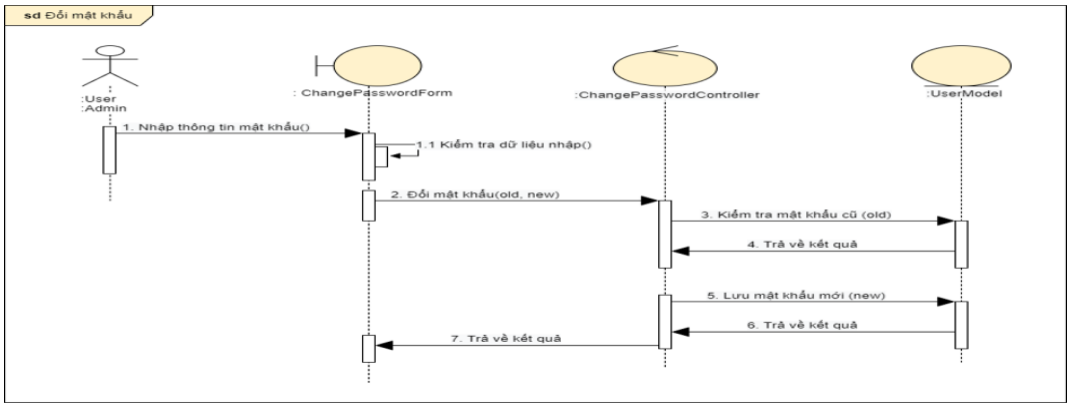
1. Tác nhân nhập thông tin theo các trường trên mẫu đổi mật khẩu.

2. Hệ thống lưu mật khẩu mới vào cơ sở dữ liệu.

Luồng rẽ nhánh: Ở bước 1, nếu tác nhân nhập dữ liệu không hợp lệ thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

Hậu điều kiện: Hệ thống thông báo tác nhân thay đổi mật khẩu thành công.

Biểu đồ tuần tự của ca sử dụng đổi mật khẩu được mô tả như hình 2.6.



Hình 2.6: Biểu đồ tuần tự của ca sử dụng đổi mật khẩu

2.3.6. Thay đổi thông tin cá nhân

Mã usecase: UC-A6 **Mã usecase liên quan:** UC-A4

Mô tả ngắn gọn: Ca sử dụng mô tả thao tác tác nhân sửa thông tin cá nhân.
Các tác nhân: Người dùng, quản trị viên

Tiền điều kiện: Tác nhân đã đăng nhập tài khoản thành công tại hệ thống

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn sửa thông tin cá nhân.

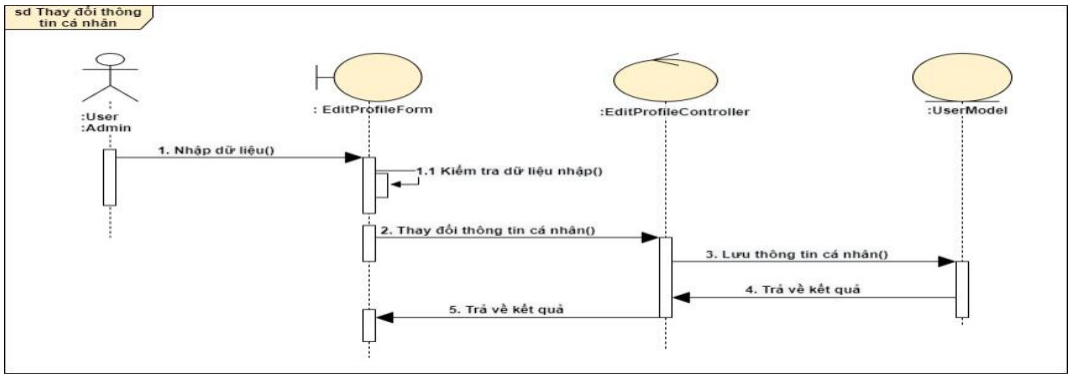
1. Tác nhân nhập thông tin theo các trường trên mẫu sửa thông tin cá nhân.

2. Hệ thống lưu thông tin cá nhân mới vào cơ sở dữ liệu.

Luồng rẽ nhánh: Ở bước 1, nếu tác nhân nhập dữ liệu không hợp lệ thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

Hậu điều kiện: Hệ thống thông báo tác nhân sửa thông tin cá nhân thành công.

Biểu đồ tuần tự của ca sử dụng thay đổi thông tin cá nhân mô tả ở hình 2.7.



Hình 2.7: Biểu đồ tuần tự của ca sử dụng thay đổi thông tin cá nhân

2.3.7. Đăng xuất

Mã usecase: UC-A7 **Mã usecase liên quan:** UC-A4

Mô tả ngắn gọn: Ca sử dụng mô tả thao tác tác nhân đăng xuất khỏi hệ thống.

Các tác nhân: Người dùng, quản trị viên

Tiền điều kiện: Không

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn chức năng đăng xuất.

1. Tác nhân chọn chức năng đăng xuất.

2. Hệ thống hiển thị cửa sổ xác nhận đăng xuất.

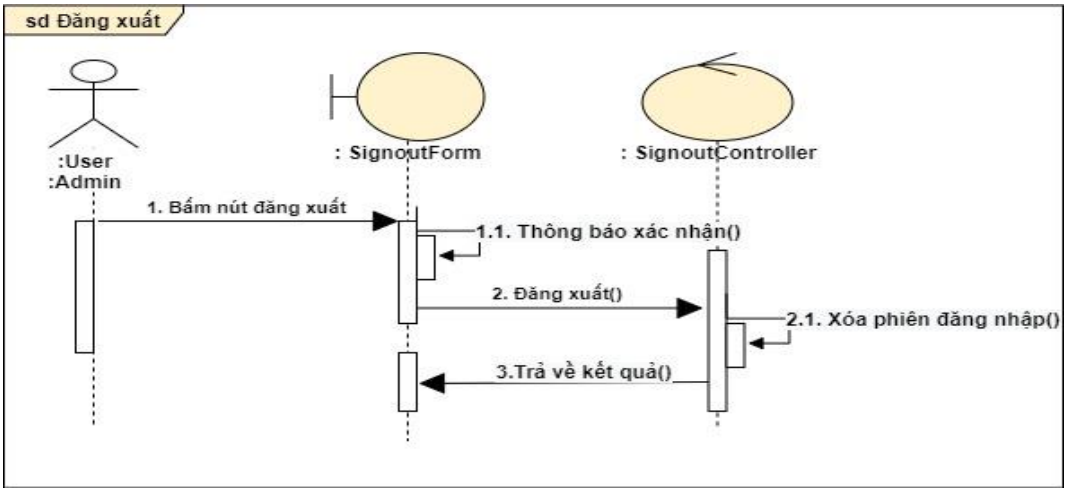
3. Tác nhân nhấn đồng ý.

4. Hệ thống sẽ xóa phiên đăng nhập và đăng xuất tài khoản ra khỏi hệ thống.

Luồng rẽ nhánh: Không

Hậu điều kiện: Hệ thống chuyển đến trang đăng nhập.

Biểu đồ tuần tự của ca sử dụng đăng xuất được mô tả như hình 2.8 dưới đây:



Hình 2.8: Biểu đồ tuần tự của ca sử dụng đăng xuất

2.3.8. Tìm kiếm/xem bệnh viện/bài báo

Mã usecase: UC-B1 **Mã usecase liên quan:** UC-A2

Mô tả ngắn gọn: Ca sử dụng mô tả thao tác tìm kiếm bệnh viện/bài báo.

Các tác nhân: Người dùng, quản trị viên.

Tiền điều kiện: Tác nhân đã đăng nhập thành công vào hệ thống.

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn chức năng tìm kiếm.

1. Hệ thống hiển thị danh sách tất cả các bệnh viện, bài báo.

2. Tác nhân nhập dữ liệu bộ lọc trên thanh tìm kiếm và nhấn nút “Tìm kiếm”.

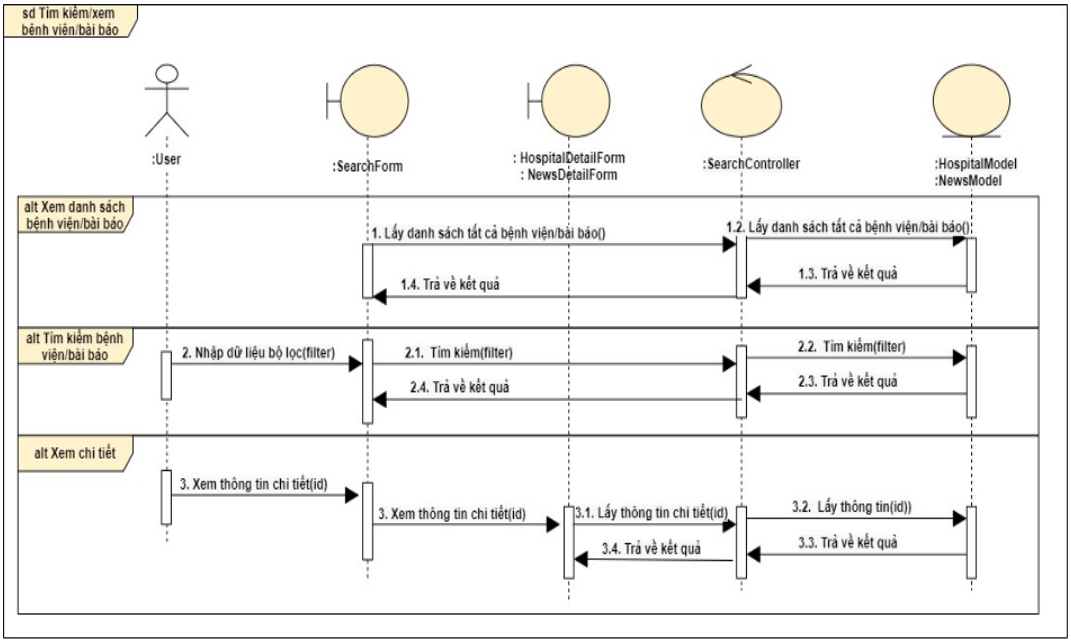
3.Tác nhân nhấn vào chức năng xem chi tiết bệnh viện/bài báo ở cột chức năng của hàng tương ứng.

4. Hệ thống hiển thị thông tin chi tiết của bệnh viện/bài báo mà tác nhân chọn.

Luồng rẽ nhánh: Ở bước 2, nếu tác nhân nhập dữ liệu không hợp lệ thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

Hậu điều kiện: Hệ thống cập nhật lại danh sách bệnh viện/bài báo và hiển thị dữ liệu mới.

Biểu đồ tuần tự của ca sử dụng tìm kiếm bệnh viện/bài báo mô tả như hình 2.9



Hình 2.9: Biểu đồ tuần tự của ca sử dụng tìm kiếm bệnh viện/bài báo

2.3.9. Quản lý bệnh viện

Mã usecase: UC-C1 **Mã usecase liên quan:** UC-A4

Mô tả ngắn gọn: Ca sử dụng mô tả thao tác tác nhân quản lý bệnh viện.

Các tác nhân: Quản trị viên.

Tiền điều kiện: Tác nhân đã đăng nhập tài khoản thành công tại hệ thống

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn chức năng quản lý bệnh viện. Các chức năng gồm tìm kiếm, xem, thêm, sửa, xóa bệnh viện

1. Hệ thống hiển thị giao diện quản lý bệnh viện bao gồm: danh bệnh viện cùng các nút xem chi tiết, sửa, xóa ở cột chức năng của từng hàng. Ngoài ra còn có thanh tìm kiếm, nút tạo mới, nút xóa hàng loạt bệnh viện ở phía trên.

2. Tác nhân chọn chức năng mong muốn

2.1. Tìm kiếm

2.1.1. Tác nhân nhập dữ liệu trên thanh tìm kiếm và nhấn “Tìm kiếm”.

2.1.2. Hệ thống sẽ hiển thị danh sách các bệnh viện thỏa mãn bộ lọc.

2.2. Xem chi tiết

2.2.1. Tác nhân chọn chức năng xem chi tiết tại cột chức năng ở hàng tương ứng bệnh viện muốn xem

2.2.2. Hệ thống sẽ hiển thị cửa sổ chứa thông tin chi tiết bệnh viện.

2.3. Sửa

2.3.1. Tác nhân chọn chức năng sửa tại cột chức năng ở hàng tương ứng bệnh viện muốn sửa.

2.3.2. Hệ thống sẽ hiển thị cửa sổ chứa thông tin chi tiết bệnh viện tác nhân vừa chọn.

2.3.3 Tác nhân nhập dữ liệu để sửa thông tin bệnh viện

2.3.4. Hệ thống lưu dữ liệu mới vào cơ sở dữ liệu và hiển thị lại danh sách các bệnh viện sau khi cập nhật

2.4. Xóa

2.4.1. Tác nhân chọn chức năng xóa tại cột chức năng ở hàng tương ứng bệnh viện muốn xóa

2.4.2. Hệ thống sẽ hiển thị cửa sổ xác nhận xóa.

2.4.3. Tác nhân nhấn đồng ý xóa.

2.4.4. Hệ thống xóa bệnh viện vừa chọn trên cơ sở dữ liệu và hiển thị lại danh sách bệnh viện sau khi cập nhật.

2.5. Thêm

2.5.1. Tác nhân chọn chức năng tạo mới.

2.5.2. Hệ thống sẽ hiển thị cửa sổ thêm bệnh viện.

2.5.3. Tác nhân nhập dữ liệu để thêm bệnh viện

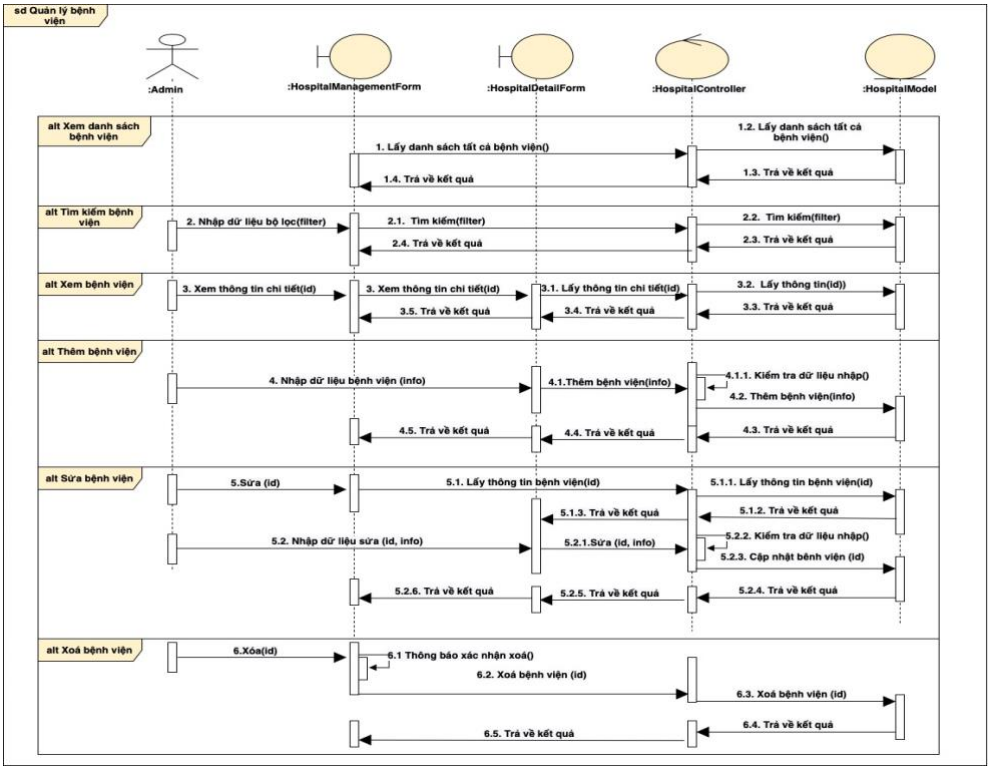
2.5.4. Hệ thống lưu dữ liệu tác nhân vừa nhập vào cơ sở dữ liệu và hiển thị lại danh sách các bệnh viện sau khi cập nhật.

Luồng rẽ nhánh:

- Ở bước 2.1.1, 2.3.3, 2.5.3 nếu tác nhân nhập dữ liệu không hợp lệ thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.
- Ở bước 2.5.3, nếu dữ liệu bệnh viện đã tồn tại thì hệ thống cập nhật dữ liệu.

Hậu điều kiện: Hệ thống cập nhật danh sách bệnh viện, hiển thị dữ liệu mới.

Biểu đồ tuần tự của ca sử dụng quản lý bệnh viện được mô tả như hình 2.10.



Hình 2.10: Biểu đồ tuần tự của ca sử dụng quản lý bệnh viện

2.3.10. Quản lý bài báo

Mã usecase: UC-C2 **Mã usecase liên quan:** UC-A4

Mô tả ngắn gọn: Ca sử dụng mô tả thao tác tác nhân quản lý bài báo.

Các tác nhân: Quản trị viên.

Tiền điều kiện: Tác nhân đã đăng nhập tài khoản thành công tại hệ thống

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn chức năng quản lý bài báo. Các chức năng gồm tìm kiếm, xem, thêm, sửa, xóa bài báo

1. Hệ thống hiển thị giao diện quản lý bài báo bao gồm: danh bài báo cùng các nút xem chi tiết, sửa, xóa ở cột chức năng của từng hàng. Ngoài ra còn có thanh tìm kiếm, nút tạo mới, nút xóa hàng loạt bài báo ở phía trên.

2. Tác nhân chọn chức năng mong muốn

2.1. Tìm kiếm

2.1.1. Tác nhân nhập dữ liệu trên thanh tìm kiếm và nhấn “Tìm kiếm”.

2.1.2. Hệ thống sẽ hiển thị danh sách các bài báo thõa mãn bộ lọc.

2.2. Xem chi tiết

2.2.1. Tác nhân chọn chức năng xem chi tiết tại cột chức năng ở hàng tương ứng bài báo muốn xem

2.2.2. Hệ thống sẽ hiển thị cửa sổ chứa thông tin chi tiết bài báo.

2.3. Sửa

2.3.1. Tác nhân chọn chức năng sửa tại cột chức năng ở hàng tương ứng bài báo muốn sửa.

2.3.2. Hệ thống sẽ hiển thị cửa sổ chứa thông tin chi tiết bài báo.

2.3.3. Tác nhân nhập dữ liệu để sửa thông tin bài báo

2.3.4. Hệ thống lưu dữ liệu và hiển thị lại danh sách sau khi cập nhật.

2.4. Xóa

2.4.1. Tác nhân chọn chức năng xóa tại cột chức năng ở hàng tương ứng bài báo muốn xóa

2.4.2. Hệ thống sẽ hiển thị cửa sổ xác nhận xóa.

2.4.3. Tác nhân nhấn đồng ý xóa.

2.4.4. Hệ thống xóa bài báo và hiển thị lại danh sách sau khi cập nhật.

2.5. Thêm

2.5.1. Tác nhân chọn chức năng tạo mới.

2.5.2. Hệ thống sẽ hiển thị cửa sổ thêm bài báo.

2.5.3. Tác nhân nhập dữ liệu để thêm bài báo

2.5.4. Hệ thống lưu dữ và hiển thị lại danh sách sau khi cập nhật.

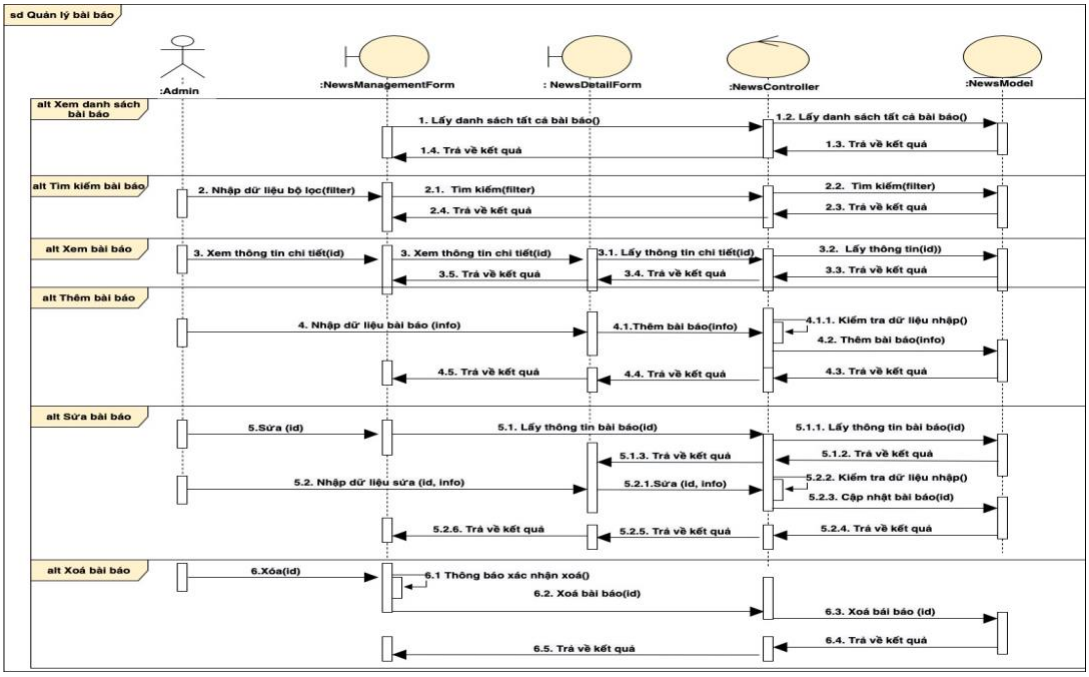
Luồng rẽ nhánh:

- Ở bước 2.1.1, 2.3.3, 2.5.3 nếu tác nhân nhập dữ liệu không hợp lệ thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

- Ở bước 2.5.3, nếu dữ liệu bài báo đã tồn tại thì hệ thống sẽ coi như là cập nhật bài báo.

Hậu điều kiện: Hệ thống cập nhật lại danh sách bài báo, hiển thị dữ liệu mới.

Biểu đồ tuần tự của ca sử dụng quản lý bài báo được mô tả như hình 2.11.



Hình 2.11: Biểu đồ tuần tự của ca sử dụng quản lý bài báo

2.3.11. Quản lý người dùng

Mã usecase: UC-C3 **Mã usecase liên quan:** UC-A4

Mô tả ngắn gọn: Ca sử dụng mô tả thao tác tác nhân quản lý người dùng.

Các tác nhân: Quản trị viên.

Tiền điều kiện: Tác nhân đã đăng nhập tài khoản thành công tại hệ thống

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn chức năng quản lý người dùng. Các chức năng gồm tìm kiếm, xem, xóa người dùng

1. Hệ thống hiển thị giao diện quản lý người dùng bao gồm: danh sách người dùng cùng các nút xem chi tiết, cấp quyền quản trị viên ở cột chức năng của từng hàng. Ngoài ra còn có thanh tìm kiếm, ở phía trên.

2. Tác nhân chọn chức năng mong muốn

2.1. Tìm kiếm

2.1.1. Tác nhân nhập dữ liệu trên thanh tìm kiếm và nhấn “Tìm kiếm”.

2.1.2. Hệ thống sẽ hiển thị danh sách người dùng thõa mãn bộ lọc.

2.2. Xem chi tiết

2.2.1. Tác nhân chọn chức năng xem chi tiết tại cột chức năng ở hàng tương ứng thông tin người dùng muốn xem

2.2.2. Hệ thống sẽ hiển thị cửa sổ chứa thông tin chi tiết người dùng.

2.3. Cấp quyền quản trị viên

2.3.1. Tác nhân chọn chức năng cấp quyền quản trị viên tại cột chức năng ở hàng tương ứng với người dùng cần cấp quyền.

2.3.2. Hệ thống sẽ hiển thị cửa sổ xác nhận cấp quyền.

2.3.3. Tác nhân nhấn đồng ý.

2.3.4. Hệ thống cấp quyền quản trị viên cho người dùng vừa chọn và hiển thị lại danh sách người dùng sau khi cập nhật.

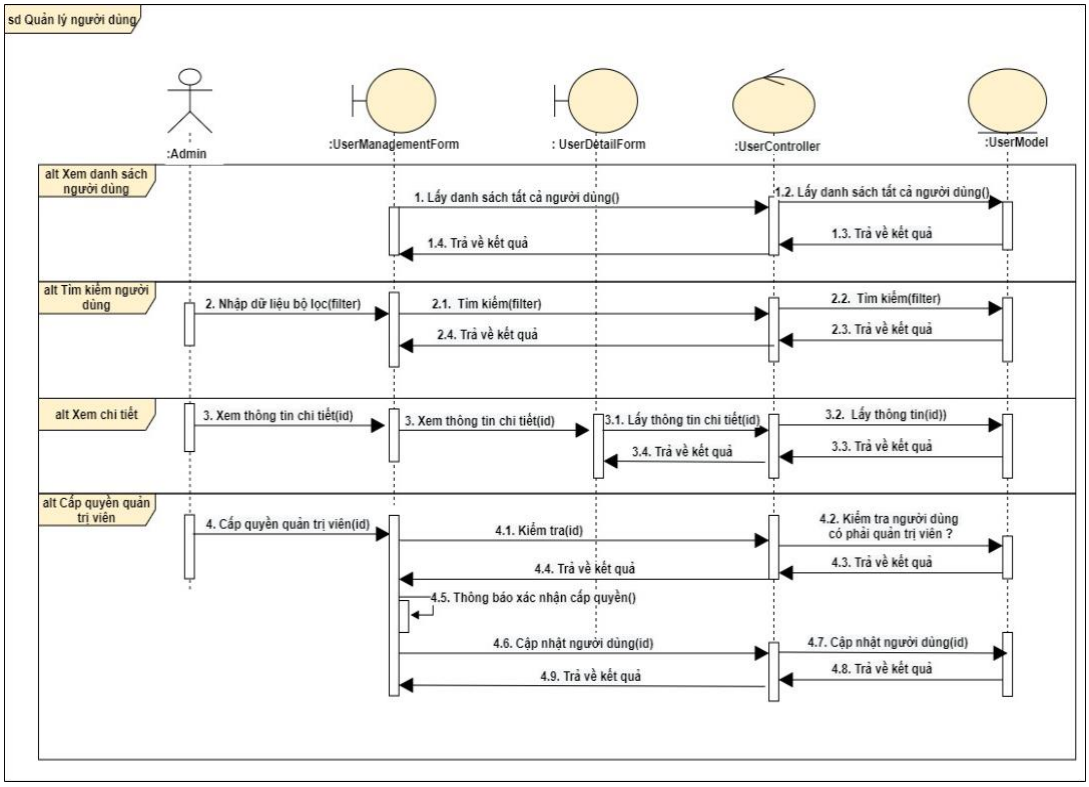
Luồng rẽ nhánh:

Ở bước 2.1.1, nếu tác nhân nhập dữ liệu không hợp lệ thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

Ở bước 2.3, nếu người dùng đang là quản trị viên thì hệ thống sẽ ẩn nút cấp quyền ở hàng tương ứng.

Hậu điều kiện: Hệ thống cập nhật danh sách người dùng, hiển thị dữ liệu mới.

Biểu đồ tuần tự của ca sử dụng quản lý người dùng được mô tả như hình 2.12.



Hình 2.12: Biểu đồ tuần tự của ca sử dụng quản lý người dùng

2.3.12. Quản lý dữ liệu thu thập

Mã usecase: UC-C4 **Mã usecase liên quan:** UC-A4

Mô tả ngắn gọn: Ca sử dụng mô tả thao tác tác nhân thu thập dữ liệu từ các trang web khác về hệ thống.

Các tác nhân: Quản trị viên

Tiền điều kiện: Tác nhân đã đăng nhập tài khoản thành công tại hệ thống

Luồng cơ bản: Ca sử dụng bắt đầu khi tác nhân chọn thu thập dữ liệu.

1. Tác nhân nhập thông tin theo các trường trên mẫu thu thập dữ liệu để mô tả các dữ liệu cần thu thập.

2. Thực hiện thu thập và lưu dữ liệu vào cơ sở dữ liệu.

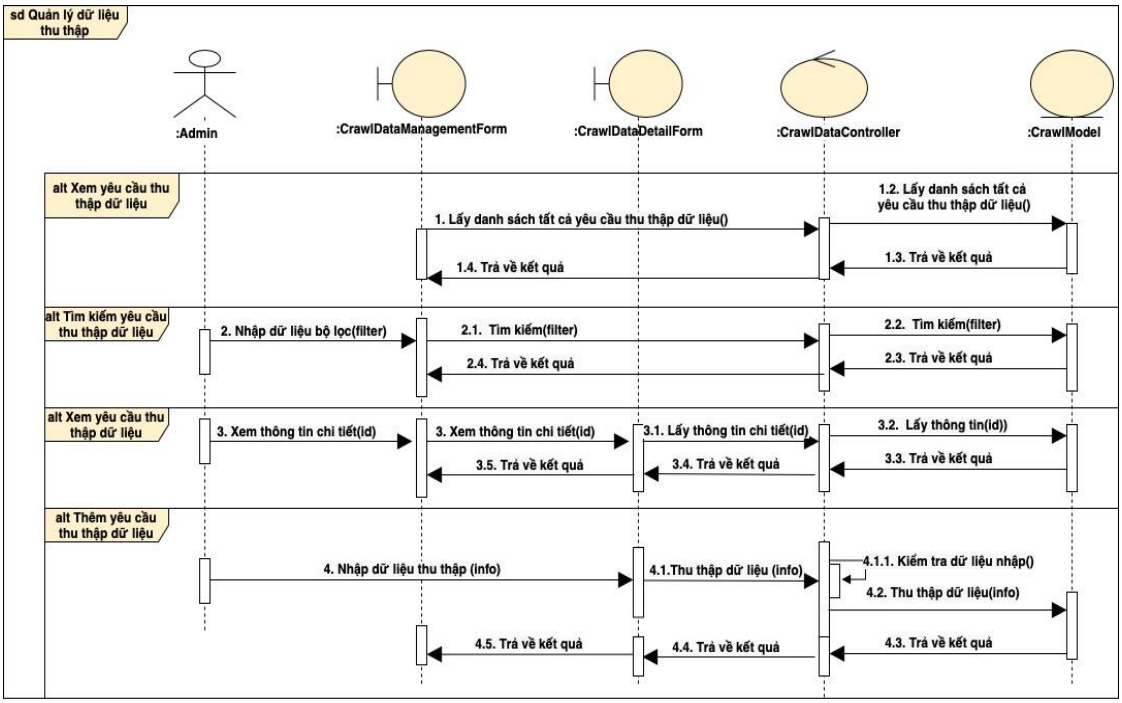
Luồng rẽ nhánh:

- Ở bước 1, nếu tác nhân nhập dữ liệu không hợp lệ thì hệ thống sẽ đưa ra cảnh báo yêu cầu tác nhân nhập lại thông tin.

- Ở bước 2, nếu hệ thống thu thập được dữ liệu bị trùng lặp với dữ liệu đã có sẵn trong cơ sở dữ liệu thì hệ thống sẽ bỏ qua dữ liệu này.

Hậu điều kiện: Hệ thống trả lại thông báo tiếp nhận thành công.

Biểu đồ tuần tự của ca sử dụng thu thập dữ liệu được mô tả như hình 2.13.



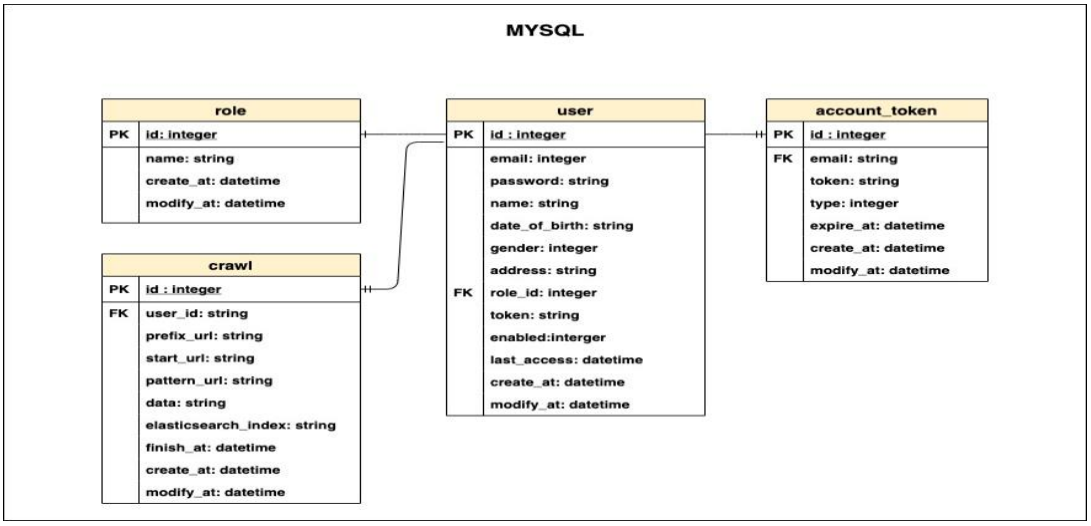
Hình 2.13: Biểu đồ hoạt động của ca sử dụng thu thập dữ liệu

2.4. Thiết kế cơ sở dữ liệu

2.4.1. Mô hình thực thể liên kết

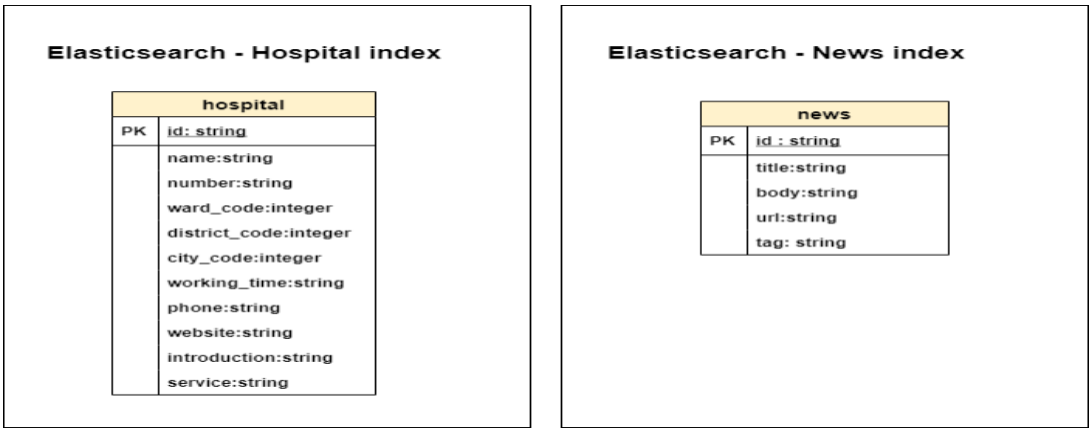
Cơ sở dữ liệu chứa thông tin người dùng, thông tin về lịch sử thu thập dữ liệu được lưu trong hệ quản trị cơ sở dữ liệu MySQL gồm 4 bảng: Role, User, Account_Token, Crawl.

Quan hệ của các bảng được mô tả như hình vẽ 2.14 bên dưới:



Hình 2.14: Mô hình thực thể liên kết

Dữ liệu bệnh viện, bài báo lưu trữ trên Elastricsearch mô tả như hình vẽ 2.15.



Hình 2.15: Mô tả dữ liệu chỉ mục trên elasticsearch

2.4.2. Các bảng cơ sở dữ liệu

**** Bảng Account_Token***

Bảng Account_Token chứa dữ liệu về thông tin các mã xác thực khi người dùng quên mật khẩu, kích hoạt tài khoản, cấu trúc bảng được mô tả trong bảng 2.1.

Bảng 2.1: Bảng Account_Token

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	INT	Primary key, Not null, Auto increment	Khóa chính của bảng Account_Token
2	Email	STRING	Not null	Email người dùng
3	Token	STRING	Not null	Mã xác thực
4	Type	INT	Not null	Loại mã xác thực, 1= loại kích hoạt tài khoản, 2 = loại quên mật khẩu.
5	Expire_at	DATETIME	Not null	Thời gian hết hạn mã
6	Create_at	DATETIME	Not null	Thời gian tạo
7	Modify_at	DATETIME		Thời gian sửa

**** Bảng Role***

Danh sách các vai trò tương ứng với các tác nhân của hệ thống được lưu trong bảng Role, cấu trúc bảng được mô tả trong bảng 2.2 sau đây:

Bảng 2.2: Bảng Role

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	INT	Primary key, Not null, Auto increment	Khóa chính của bảng Role
2	Name	VARCHAR(45)	Not null	Tên vai trò
3	Create_at	DATETIME	Not null	Thời gian tạo
4	Modify_at	DATETIME		Thời gian sửa

**** Bảng Crawl***

Bảng Crawl chứa lịch sử thu thập dữ liệu của quản trị viên, cấu trúc bảng được mô tả trong bảng 2.3 sau đây:

Bảng 2.3: Bảng Crawl

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	INT	Primary key, Not null, Auto increment	Khóa chính của bảng Crawl
2	Prefix_url	INT	Not null	Tiền tố đường dẫn hợp lệ
3	Start_url	LONGTEXT	Not null	Đường dẫn bắt đầu thu thập
4	Pattern_url	STRING	Not null	Mẫu của đường dẫn đến trang thu thập cuối cùng theo chuẩn Python Regular Expression [13]
5	Data	STRING		JSON chứa xpath đến các dữ liệu muốn thu thập
6	Elasticsearch _index			Tên index lưu trên elasticsearch
7	Finish_at			Thời gian kết thúc thu thập dữ liệu
8	Create_at	DATETIME	Not null	Thời gian tạo
9	Modify_at	DATETIME		Thời gian sửa

**** Bảng User***

Bảng User chứa dữ liệu về thông tin tài khoản của người dùng hệ thống, cấu trúc bảng được mô tả trong bảng 2.4 sau đây:

Bảng 2.4: Bảng User

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	INT	Primary key, Not null, Auto increment	Khóa chính của bảng User

2	Email	VARCHAR(60)	Not null	Email đăng ký
3	Password	VARCHAR(255)	Not null	Mật khẩu đã mã hóa
4	Name	VARCHAR(45)	Not null	Họ và tên
5	Date_of_birth	DATETIME	Not null	Ngày, tháng, năm sinh
6	Gender	INT	Not null	Giới tính:0=Nam, 1=Nữ
7	Address	TEXT		Địa chỉ
8	Role_id	INT	Not null	Mã vai trò
9	Enabled	INT	Not null	0=Chưa kích hoạt tài khoản, 1=Đã xác thực tài khoản
10	Last_access	DATETIME		Thời gian cuối cùng đăng nhập
11	Create_at	DATETIME	Not null	Thời gian tạo
12	Modify_at	DATETIME		Thời gian sửa

**** Bảng News***

Bảng News chứa dữ liệu về thông tin các bài báo, cấu trúc bảng được mô tả trong bảng 2.5 sau đây:

Bảng 2.5: Bảng News

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	INT	Primary key, Not null, Auto increment	Khóa chính của bảng News
2	Title	INT	Not null	Tiêu đề bài báo
3	body	LONGTEXT	Not null	Nội dung bài báo
4	url	STRING	Not null	Đường dẫn đến bài báo
5	Tag	STRING		Chủ đề bài báo

**** Bảng Hospital***

Bảng Hospital chứa dữ liệu về thông tin các bệnh viện, cấu trúc bảng được mô tả trong bảng 2.6 sau đây:

Bảng 2.6: Bảng Hospital

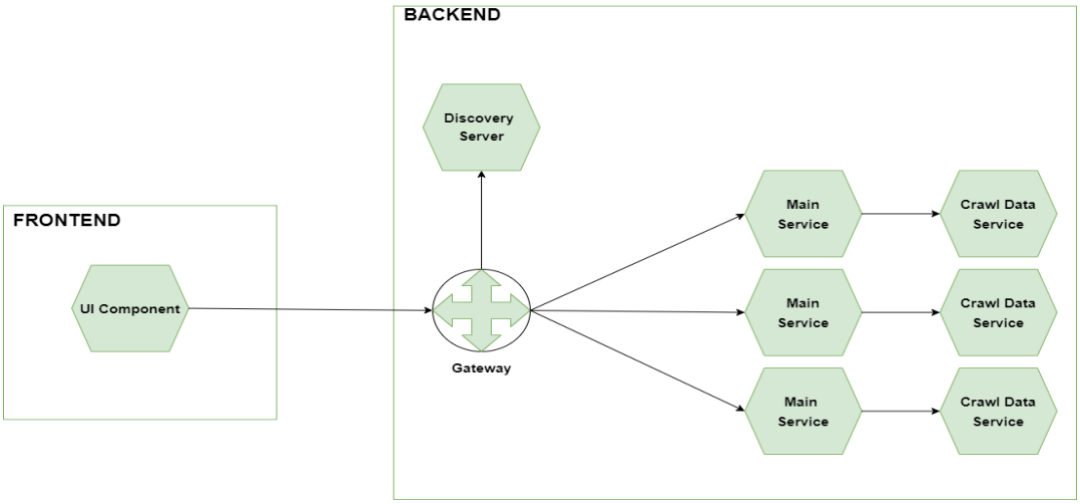
STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	INT	Primary key, Not null, Auto increment	Khóa chính của bảng Hospital
2	Name	VARCHAR(45)	Not null	Tên bệnh viện
3	Number	String		Tên địa chỉ(số nhà, ngách, ngõ..)
4	Ward_code	INT		Mã phường/xã
5	District_code	INT		Mã quận/huyện
6	City_code	INT	Not null	Mã tỉnh/thành phố
7	Working_time	String		Thời gian làm việc
8	Phone	VARCHAR(255)		Số điện thoại
9	Website	VARCHAR(255)		Địa chỉ website
10	Introduction	LONGTEXT		Giới thiệu bệnh viện
11	Service	LONGTEXT		Mô tả các dịch vụ

2.5. Kiến trúc hệ thống

Hệ thống tự động tổng hợp thông tin về dịch bệnh được xây dựng dựa trên kiến trúc tiểu dịch vụ gồm 2 tầng kiến trúc: tầng giao diện (Front-end), tầng ứng dụng (Back-end) như trong hình 2.16. Trong đó, tầng front-end chứa thực thể dịch vụ UI Compoment là hệ thống giao diện cho ứng dụng tự động tổng hợp thông tin về dịch bệnh. Tầng back-end gồm 4 thực thể dịch vụ: Gateway, Discovery server, Main Service, Crawl Data Service.

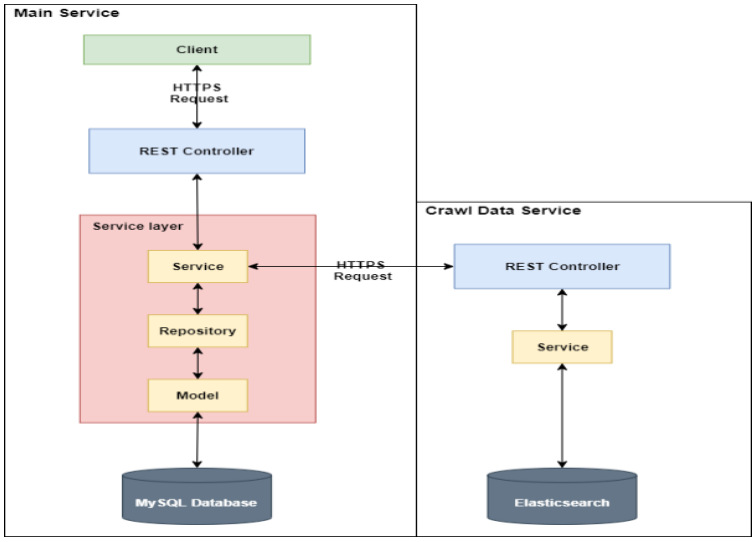
Mỗi thực thể dịch vụ đều có một địa chỉ định danh để các thực thể dịch vụ khác trong kiến trúc tiểu dịch vụ giao tiếp tới, trong thực tế, chúng có xu hướng thay

đổi liên tục để không bị lộ ra ngoài [10]. Vì vậy, discovery server sinh ra để lưu trữ thông tin và vị trí mạng của các tiểu dịch vụ này. Sau đó, mỗi khi UI Component gửi một yêu cầu (request) đến back-end, thực thể dịch vụ này sẽ giao tiếp với Gateway. Tại đây, Gateway sẽ xác thực, phân quyền, dùng thông tin lấy được tại Discovery server để định tuyến và cân bằng tải các yêu cầu đến các thực thể dịch vụ tương ứng. Main Service là thực thể dịch vụ chứa các giao diện lập trình ứng dụng (Application Programming Interface - API) để xử lý logic cho hệ thống tự động tổng hợp thông tin về dịch bệnh, có thể có nhiều thể hiện (instances) để giúp hệ thống xử lý được nhiều yêu cầu một lúc. Crawl Data Service là thực thể dịch vụ chứa hệ thống thu thập dữ liệu và quản lý dữ liệu trên Elasticsearch, giao tiếp với Main Service qua API để thực hiện chức năng quản lý bệnh viện, quản lý bài báo.



Hình 2.16: Kiến trúc tổng quan của hệ thống

Khi Gateway gửi yêu cầu đến thực thể dịch vụ Main Service. Luồng xử lý yêu cầu được mô tả như hình 2.17. Đầu tiên, yêu cầu sẽ được ánh xạ, phân tích và xử lý tại REST Controller bằng cách lấy dữ liệu từ tầng dịch vụ (Service). Tiếp theo, tầng dịch vụ chịu trách nhiệm kiểm tra dữ liệu đầu vào, xử lý logic và gọi xuống tầng quản lý dữ liệu (Repository) để lấy dữ liệu. Cuối cùng, tầng quản lý dữ liệu sẽ thực hiện các câu truy vấn dựa trên các thực thể (Model) và tương tác với cơ sở dữ liệu (Database) để lấy dữ liệu. Với các yêu cầu tìm kiếm, quản lý bệnh viện/bài báo, tầng dịch vụ sẽ gọi sang API của Crawl Data Service để lấy dữ liệu, Crawl Data Service sẽ giao tiếp với API của Elasticsearch để thực hiện yêu cầu.



Hình 2.17: Luồng xử lý yêu cầu tại Main Service

2.6. Trích xuất thông tin dịch sốt xuất huyết

2.6.1. Luồng trích xuất thông tin

Khi trích xuất thông tin về dịch sốt xuất huyết tại một website bất kỳ, do cấu trúc DOM của mỗi website là khác nhau, quản trị viên cần phải tìm ra những giá trị của Xpath chứa thông tin cần thu thập và gửi lên hệ thống.

Sau khi dữ liệu được thu thập, nếu có dữ liệu đã tồn tại (bài viết có cùng đường dẫn đã tồn tại) thì hệ thống sẽ ghi đè dữ liệu mới lên dữ liệu cũ.

2.6.2. Mẫu đặc tả trích xuất thông tin về dịch sốt xuất huyết

Hệ thống thu thập dữ liệu bằng framework Scrapy sẽ đi đến tất cả các đường dẫn có trong các trang đi qua. Để tránh việc thu thập các trang web khác, có thể gây lỗi do định dạng không đúng, ta cần chỉ định rõ các tên miền, hay tiền tố của các đường dẫn hợp lệ, gọi là **prefixUrl**.

Khi hệ thống tự động thu thập dữ liệu, quản trị viên cũng cần chỉ rõ đường dẫn bắt đầu cho việc thu thập, gọi là **startUrl**.

Để đi tới được trang web chứa các bài báo, có thể hệ thống sẽ được dẫn tới một trang web trung gian chứa các chuyên mục, ở đây cũng cần phải chỉ ra mẫu của đường dẫn tại trang cần thu thập cuối cùng, định dạng theo chuẩn Python Regular Expression [13], gọi là **patternUrl**.

Khi một đường dẫn nào đó thỏa mãn điều kiện, hệ thống sẽ tải về và kiểm tra dữ liệu, nếu thỏa mãn sẽ lưu lại trên elasticsearch, vì ở đây có hai dữ liệu về bệnh viện và bài báo nên cần lưu trên hai index khác nhau. Quản trị viên cũng cần chỉ rõ tên của index muốn lưu, gọi là **elasticsearchIndex**. Trong hệ thống này, quy định nếu thu thập dữ liệu bệnh viện thì elasticsearchIndex là "hospital", nếu thu thập dữ liệu bài báo thì elasticsearchIndex là "news". Hệ thống sẽ thu thập đầy đủ các trường theo định dạng mà hệ thống có sẵn. Ví dụ: dữ liệu bệnh viện sẽ có các trường: name, number, ward, district, city, workingTime, introduction, services, department, website, link, phone. Dữ liệu bài báo sẽ có các trường: title, body, tag, link.

Dữ liệu các trường sẽ nằm ở vị trí khác nhau trên các trang web, vì vậy quản trị viên cần phân tích trang web trước và lấy ra được xpath của những dữ liệu cần lấy. Những thông tin đó sẽ được đặt trong mục gọi là **data**, bao gồm xpath của title, tag, body. Ngoài ra cần quy định keyword có trong body, cụ thể trong đề án này keyword sẽ là "sốt xuất huyết" để tổng hợp thông tin về dịch sốt xuất huyết.

Mẫu đặc tả dữ liệu để trích xuất thông tin về dịch sốt xuất huyết cho trang web https://www.vinmec.com/vi/tin-tuc/ được viết dưới dạng json như sau:

```
{
  "prefixUrl": "https://www.vinmec.com/vi/tin-tuc/",
  "startUrl":"https://www.vinmec.com/vi/tin-tuc/",
  "patternUrl":"^https://www.vinmec.com/vi/tin-tuc/(([a-z]|[-])+)/(([a-z]|[-])+)/((([a-z]|[-])+)/)*$",
  "elasticsearchIndex":"news",
  "data": {
     "title":"normalize-space(//*[@id=\"vue-bootstrap\"]/div[2]/div[2]/h1/text())",
     "tag":"//*[@id=\"vue-bootstrap\"]/div[2]/div[3]/div[3]/a/text()",
     "body":"//*[@id=\"vue-bootstrap\"]/div[2]/div[3]/div[1]/div[1]",
     keyword:"Sốt xuất huyết"
  }
}
```

2.7. Kết luận chương 2

Trong chương 2, đề án đã phân tích thiết kế hệ thống hệ thống tổng hợp thông tin về dịch bệnh và xây dựng được mẫu đặc tả dữ liệu để trích xuất thông tin về dịch sốt xuất huyết trên Scrapy framework.

Dựa trên nội dung đã được phân tích và thiết kế trong chương 2, đề án sẽ triển khai xây dựng hệ thống tổng hợp thông tin về dịch bệnh cụ thể là dịch sốt xuất huyết.

CHƯƠNG 3: TRIỂN KHAI VÀ XÂY DỰNG HỆ THỐNG

Trong chương 3 đề án đi sâu và chi tiết về việc xây dựng hệ thống tự động tổng hợp thông tin về dịch bệnh theo kiến trúc tiểu dịch vụ sử dụng Scrapy framework và các công cụ lập trình web đã giới thiệu ở chương 1. Dựa trên mẫu đặc tả dữ liệu để trích xuất thông tin ở chương 2, đề án sẽ triển khai thu thập dữ liệu mẫu từ hai trang web https://timbenhvien.vn và https://www.vinmec.com.

3.1 Kịch bản triển khai

3.1.1 Các nội dung triển khai

Hệ thống tự động tổng hợp thông tin về dịch bệnh sẽ triển khai với các nội dung và tính năng chính sau:

- Thu thập dữ liệu tự động: Hệ thống tự động thu thập thông tin từ các nguồn đáng tin cậy như tổ chức y tế quốc gia, tổ chức y tế thế giới, các cơ quan y tế địa phương. Thực hiện triển khai trích xuất các thông tin liên quan về bệnh sốt xuất huyết trên hai trang web: https://timbenhvien.vn/ và https://www.vinmec.com/vi/tin-tuc/.

- Cập nhật thông tin liên tục: Hệ thống cập nhật thông tin mới nhất về bệnh sốt xuất huyết từ các nguồn thông tin đáng tin cậy và đảm bảo người dùng luôn nhận được thông tin mới nhất.

- Tính năng tìm kiếm và lọc thông tin: Cung cấp tính năng tìm kiếm và lọc thông tin để người dùng có thể tìm kiếm thông tin cụ thể về bệnh sốt xuất huyết hoặc về tình hình dịch bệnh theo địa điểm và thời gian.

- Tính năng quản lý người dùng, dữ liệu thu thập về bệnh viện và tin tức cho quản trị viên.

- Triển khai hệ thống theo mô hình kiến trúc tiểu dịch vụ để dễ dàng sửa, mở rộng, tích hợp với hệ thống khác sẵn có.

3.1.2. Thu thập dữ liệu tự động

3.1.2.1. Thu thập dữ liệu bệnh viện

Dữ liệu bệnh viện được thu thập từ trang web: https://timbenhvien.vn/

Dữ liệu được thu thập thông qua Scrapy framework, tuy nhiên phần lấy số điện thoại đã bị trang web che đi, cần phải thực hiện click vào nút “Hiển thị số điện thoại”, nên

phần thu thập số điện thoại được thực hiện bằng Selenium framework [14], mô phỏng lại một chorme driver cho các thao tác truy cập đường dẫn, bấm vào nút "Hiển thị số điện thoại" để lấy dữ liệu.

Kết quả thu thập được: Tổng có 8157 bản ghi, được lưu trên elasticseach với các thông tin được mô tả dưới bảng 3.1.

Bảng 3.1: Bảng mô tả thu thập dữ liệu bệnh viện

STT	Tên cột	Giải thích	Ví dụ
1	Id	Mã định danh	42ATD3kBZLAYSuhRazKy
2	Name	Tên bệnh viện	Bảo Hà Spa - CS3
3	Number	Tên đường/phố	109/6 Nguyễn Bỉnh Khiêm
4	Ward	Tên phường/xã	Phường Đa Kao
5	District	Tên quận/huyện	Quận 1
6	City	Tên Tỉnh/Thành Phố	Hồ Chí Minh
7	workingTime	Thời gian làm việc	T2,T3,T4,T5,T6: 09:00 - 18:00
8	Introduction	Giới thiệu về bệnh viện	Bảo Hà Spa luôn làm việc bằng cả tấm lòng yêu thương, đồng cảm với khách hàng,…
9	Services	Dịch vụ của bệnh viện	Dịch vụ chăm sóc mẹ bầu, Dịch vụ chăm sóc mẹ sau sinh,
10	Department	Các khoa của bệnh viện	Khoa Da Liễu, Nhi Khoa, Sản Khoa
11	Website	Trang web của bệnh viện	http://baohaspa.vn
12	Link	Trang web thu thập dữ liệu	https://timbenhvien.vn/chi-tiet/bao-ha-spa---cs3/7364
13	Phone	Số điện thoại	0941.958.186

3.1.2.2. Thu thập dữ liệu bài báo

Dữ liệu bài báo được thu thập tại: https://www.vinmec.com/vi/tin-tuc/

Kết quả thu thập được: Tổng có 14138 bản ghi, trong đó có 626 bản ghi có nội dung liên quan đến dịch sốt xuất huyết được lưu trên elasticseach với các thông tin được mô tả dưới bảng 3.2.

Bảng 3.2: Bảng mô tả thu thập dữ liệu bài báo

STT	Tên cột	Giải thích	Ví dụ
1	Id	Mã định danh	5k3OvY0BiLF8oB1QQbTR
2	Title	Tiêu đề bài báo	Sốt xuất huyết không được uống thuốc gì?
3	Body	Nội dung bài báo	Đặc điểm bệnh sốt xuất huyết là không có thuốc đặc trị. Do vậy người mắc bệnh cần hết sức lưu ý khi điều trị bằng thuốc. Bài viết dưới đây cung cấp thông tin sốt xuất huyết không được uống thuốc gì.
4	Link	Địa chỉ bài báo	https://www.vinmec.com/vi/tin-tuc/thong-tin-suc-khoe/suc-khoe-tong-quat/sot-xuat-huyet-khong-duoc-uong-thuoc-gi/
5	Tag	Chủ đề bài báo	["Paracetamol","Thuốc hạ sốt", "Oresol","Điều trị sốt xuất huyết","Truyền nhiễm","Sốt xuất huyết"]

3.1.3 Các yêu cầu cần đạt của hệ thống

Hệ thống tự động tổng hợp thông tin về dịch bệnh cần đạt các yêu cầu sau:

- Độ chính xác và đáng tin cậy: Hệ thống cần thu thập, xử lý và hiển thị thông tin một cách chính xác và đáng tin cậy từ các nguồn dữ liệu uy tín và chính thống.

- Tính cập nhật: Hệ thống cần cập nhật thông tin mới nhất và liên tục về dịch bệnh, bao gồm số ca nhiễm, số ca tử vong, số ca hồi phục và các biến động của dịch bệnh.

- Dễ sử dụng: Giao diện người dùng của hệ thống cần được thiết kế một cách trực quan và dễ sử dụng, giúp người dùng dễ dàng tìm kiếm, truy cập và hiểu thông tin.

- Tính linh hoạt: Hệ thống cần có khả năng tùy chỉnh và điều chỉnh theo nhu cầu của người dùng và tình hình dịch bệnh, bao gồm tính năng tìm kiếm, lọc thông tin và cập nhật nhanh chóng.

- Bảo mật thông tin: Hệ thống cần đảm bảo bảo mật thông tin cá nhân và dữ liệu người dùng, đồng thời tuân thủ các quy định và chuẩn mực về bảo vệ dữ liệu.

3.2. Triển khai xây dựng tầng ứng dụng hệ thống

3.2.1. Môi trường

Hệ thống được triển khai trên môi trường:

- MacOS Catalina, 3.1GHz Dual-Core Intel Core i7, 16GB 1867 MHz DDR3

- Java 1.8, Python 3.9.0, Elasticsearch 7.10.1, MySQL 8.0.22, Yarn 1.22.10

3.2.2. Ứng dụng hệ thống

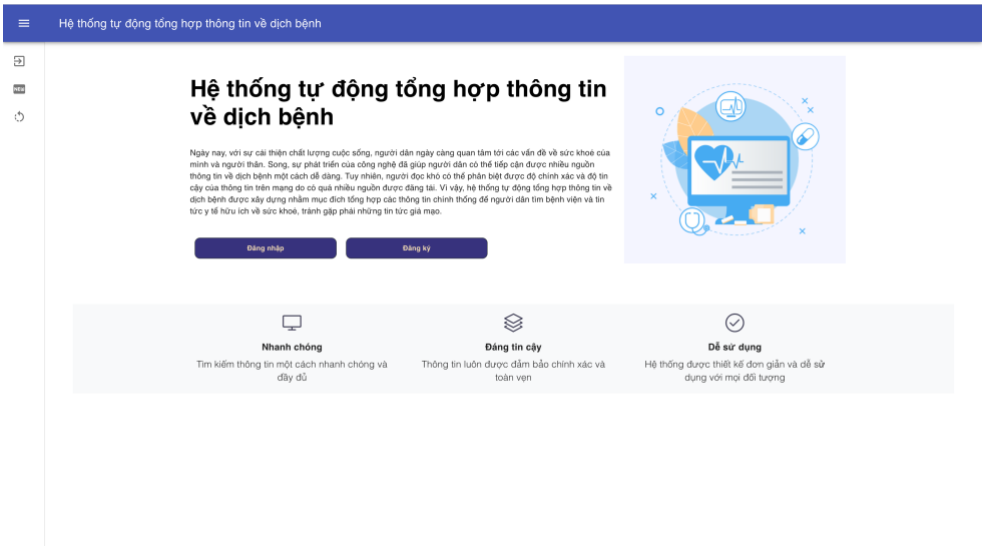
Hệ thống sử dụng kiến trúc tiểu dịch vụ cho phép chia một hệ thống lớn thành một vài thành phần nhỏ hơn để độc lập về phát triển, kiểm thử và triển khai ứng dụng. Vậy nên đề án này sử dụng Spring Cloud [10] là một dự án con trong họ nhà Spring để phát triển các ứng dụng phân tán, được tạo từ Spring Boot để giảm thiểu các cấu hình phức tạp giúp dễ cài đặt hơn. Cụ thể: Discovery Server sẽ được triển khai bằng Spring Cloud Netflix Eureka, Gateway được triển khai bằng Spring Cloud Netflix Zuul Proxy, kết hợp với Spring Security, sử dụng Json Web Token để xác thực và phân quyền. Ứng dụng dịch vụ Main Service sử dụng Spring Data cung cấp các API

dạng RESTful để giao tiếp với tầng giao diện qua giao thức HTTP. Ứng dụng Crawl Data Service được viết bằng Python sử dụng Flask Framework để tạo ra ứng dụng web chứa các API, giao tiếp với thực thể dịch vụ Main Service. Ngoài ra Crawl Data Service sử dụng framework mạnh mẽ nhất để thu thập dữ liệu hiện nay là sử dụng Framework Scrapy. Dữ liệu sau khi thu thập sẽ được chuẩn hóa rồi đẩy lên elasticsearch để lưu trữ.

3.3. Triển khai xây dựng tầng giao diện hệ thống

3.3.1. Giao diện trang chủ hệ thống

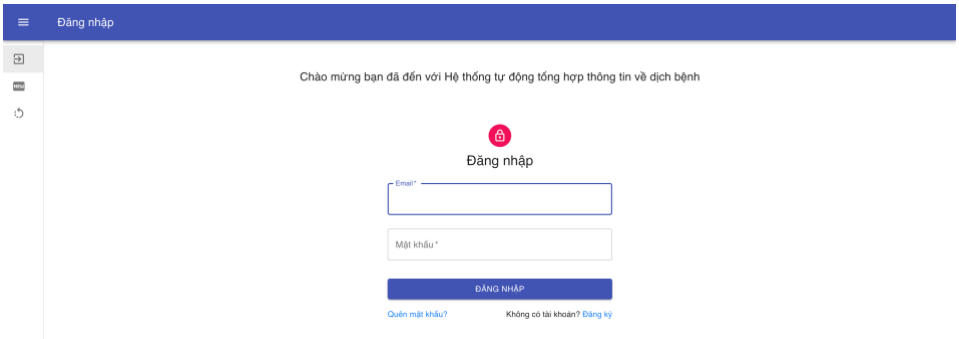
Hình 3.1 mô tả giao diện trang chủ của hệ thống



Hình 3.1: Giao diện trang chủ

3.3.2. Giao diện đăng nhập

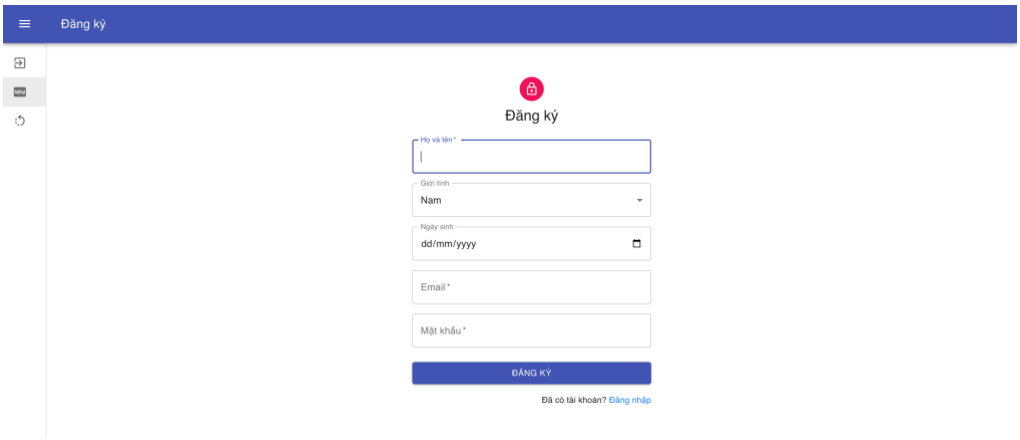
Hình 3.2 mô tả giao diện đăng nhập của hệ thống



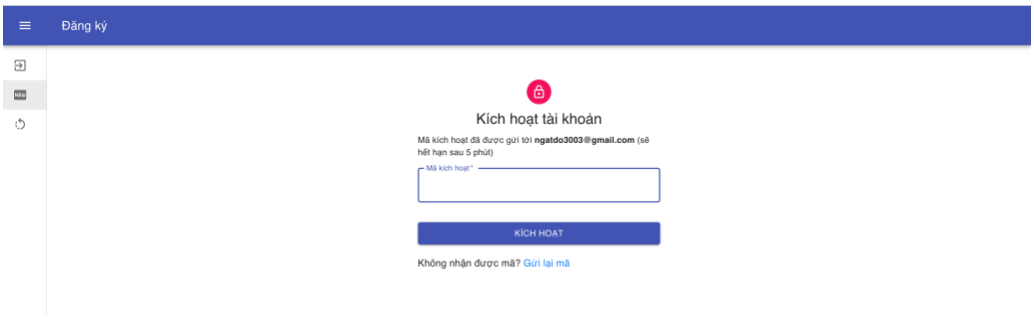
Hình 3.2: Giao diện đăng nhập

3.3.3. Giao diện đăng ký tài khoản, kích hoạt tài khoản

Hình 3.3 mô tả giao diện đăng ký tài khoản. Giao diện diện kích hoạt tài khoản mô tả như hình 3.4 được hiển thị sau khi người dùng đăng ký tài khoản thành công, hoặc nếu người dùng đăng nhập mà chưa kích hoạt tài khoản.



Hình 3.3: Giao diện đăng ký tài khoản

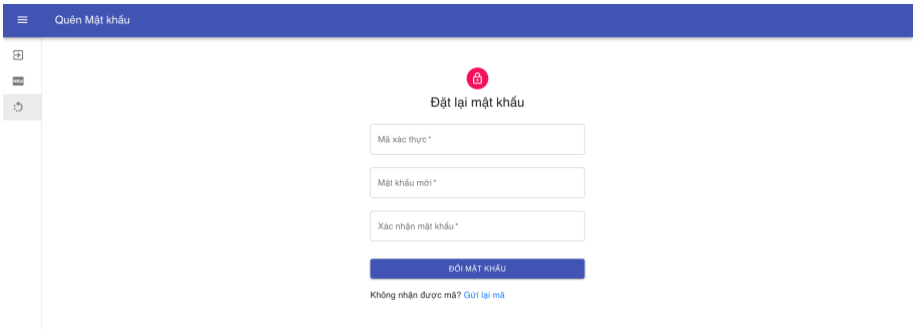


Hình 3.4: Giao diện kích hoạt tài khoản

3.3.4. Giao diện quên mật khẩu

Hình 3.5 mô tả giao diện khi người dùng đặt lại mật khẩu, người dùng cần nhập email để lấy lại mật khẩu. Hệ thống sẽ gửi mail chứa mã xác thực đến email vừa nhập. Người dùng sẽ dùng mã xác thực đó để lấy lại mật khẩu như trong hình 3.6.

Hình 3.5: Giao diện quên mật khẩu

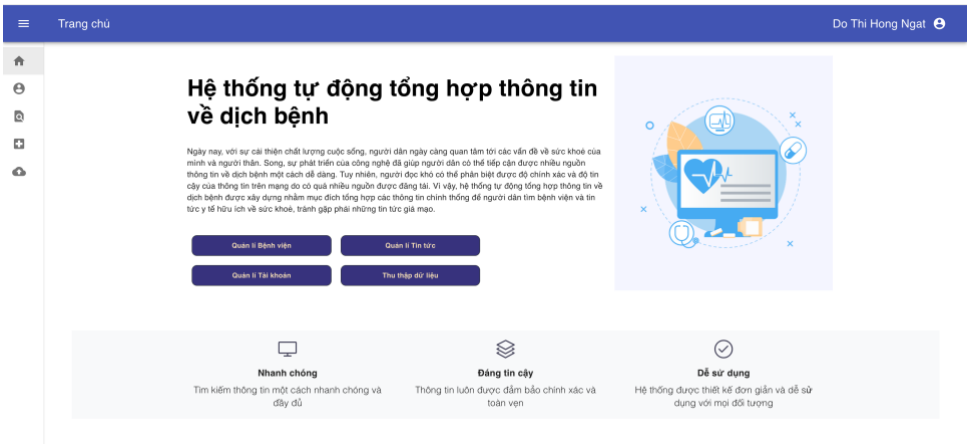


Hình 3.6: Giao diện nhập mã xác thực và đặt lại mật khẩu

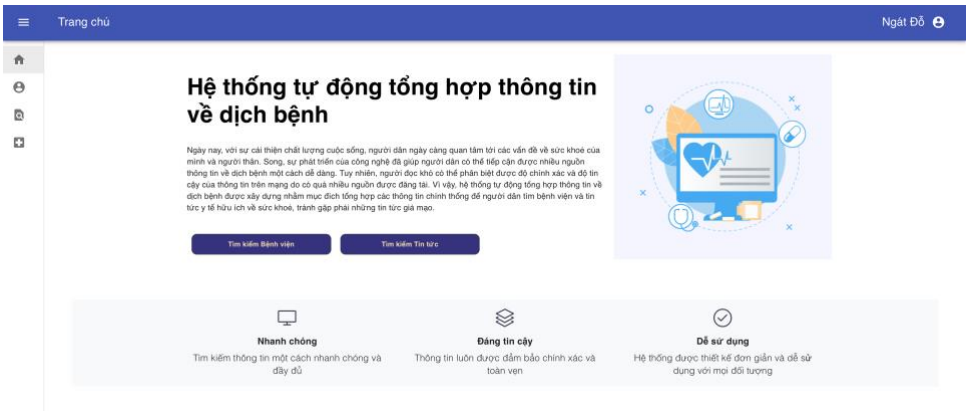
Nếu mã xác thực và mật khẩu mới hợp lệ thì hệ thống sẽ hiển thị giao diện đặt lại mật khẩu thành công.

3.3.5. Giao diện trang chủ sau khi đăng nhập

Hình 3.7 mô tả giao diện trang chủ của quản trị viên, hình 3.8 mô tả giao diện trang chủ của người dùng sau khi đăng nhập thành công.



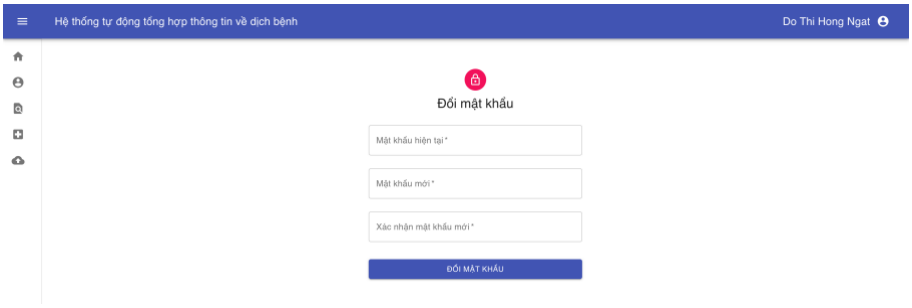
Hình 3.7: Giao diện trang chủ của quản trị viên



Hình 3.8: Giao diện trang chủ của người dùng

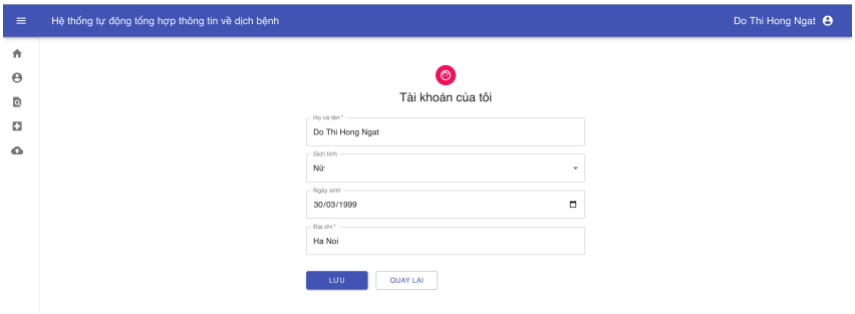
3.3.6. Giao diện đổi mật khẩu, thay đổi thông tin tài khoản

Hình 3.9 mô tả giao diện đổi mật khẩu.



Hình 3.9: Giao diện đổi mật khẩu

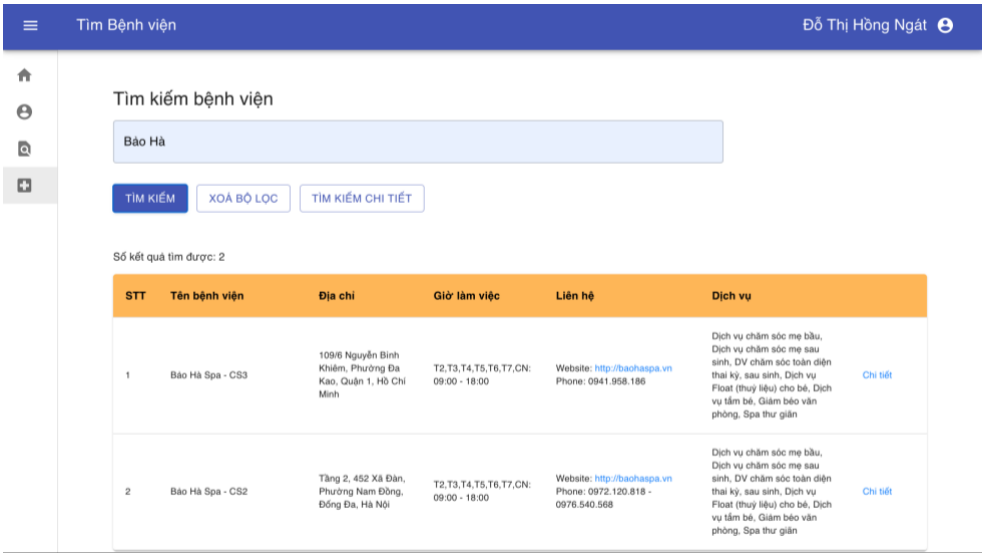
Hình 3.10 mô tả giao diện người dùng thay đổi thông tin tài khoản.



Hình 3.10: Giao diện thay đổi thông tin tài khoản

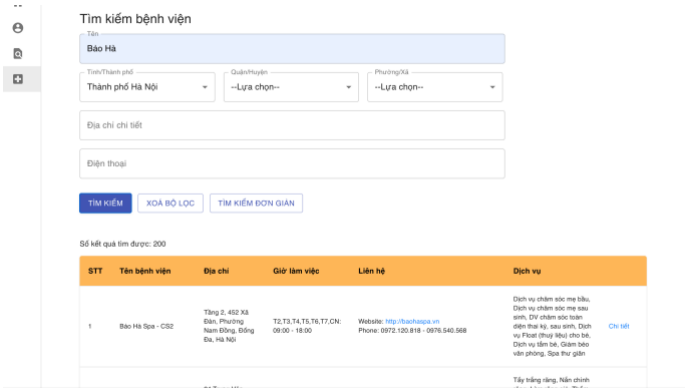
3.3.7. Giao diện tìm kiếm bệnh viện/bài báo của người dùng.

Hình 3.11 mô tả giao diện tìm kiếm mọi trường của bệnh viện.



Hình 3.11: Giao diện tìm kiếm bệnh viện

Tìm kiếm bệnh viện chi tiết được mô tả trong hình 3.12, người dùng sẽ nhập thông tin bộ lọc và hệ thống sẽ trả về kết quả tương ứng.

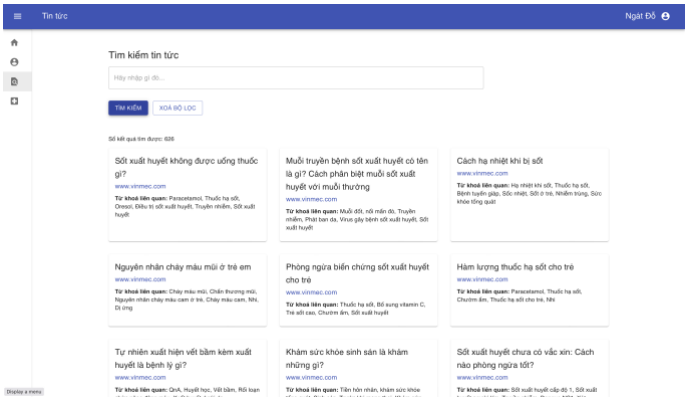


Hình 3.12: Giao diện tìm kiếm chi tiết bệnh viện

Người dùng có thể xem chi tiết thông tin bệnh viện như trong hình 3.13.

Hình 3.13: Giao diện xem chi tiết bệnh viện của người dùng

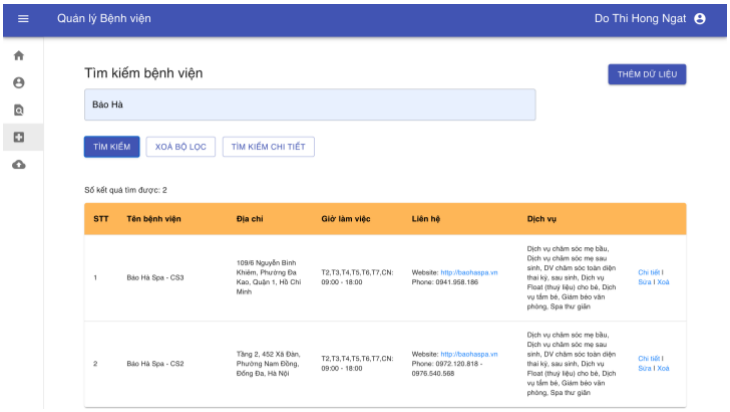
Giao diện tìm kiếm bài báo của người dùng được mô tả như hình 3.14.



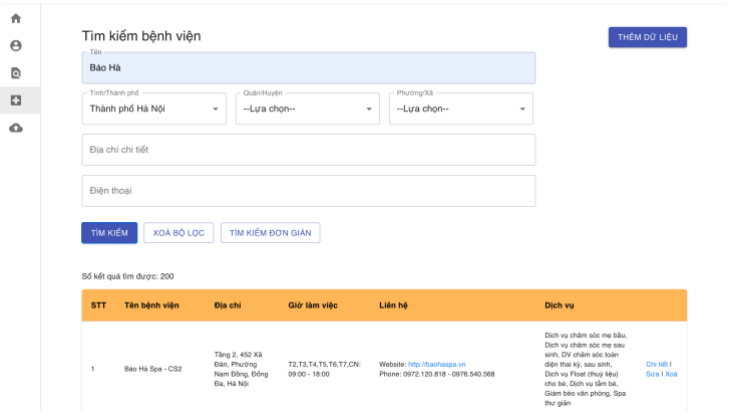
Hình 3.14: Giao diện tìm kiếm bài báo của người dùng

3.3.8. Giao diện quản lý bệnh viện

Hình 3.15 mô tả giao diện quản trị viên tìm kiếm bệnh viện, tìm kiếm chi tiết bệnh viện cũng được mô tả trong hình 3.16.

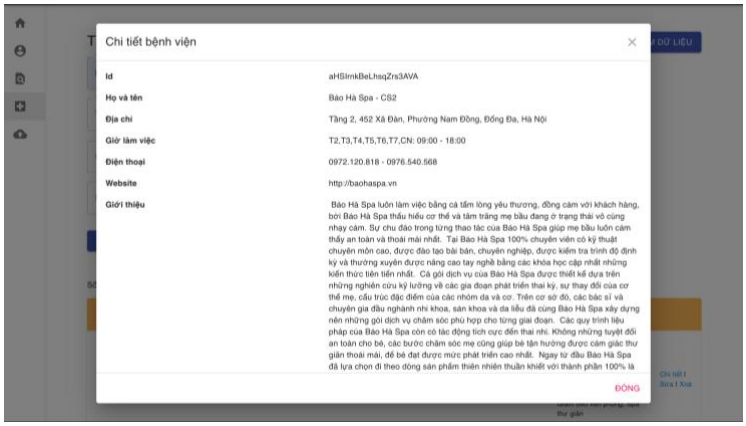


Hình 3.15: Giao diện tìm kiếm bệnh viện của quản trị viên



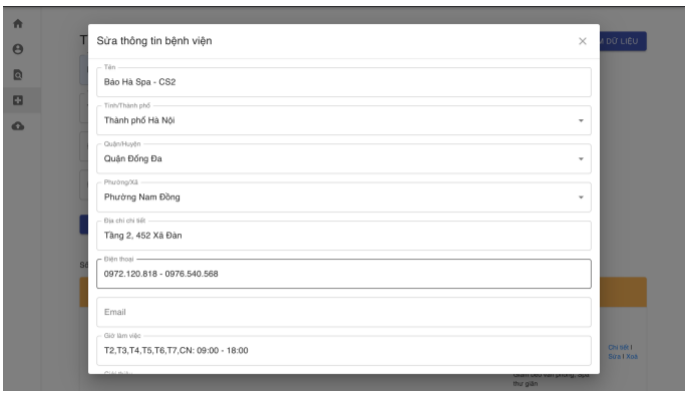
Hình 3.16: Giao diện tìm kiếm chi tiết bệnh viện của quản trị viên

Giao diện xem chi tiết bệnh viện của quản trị viên được mô tả như hình 3.17.



Hình 3.17: Giao diện xem chi tiết bệnh viện của quản trị viên

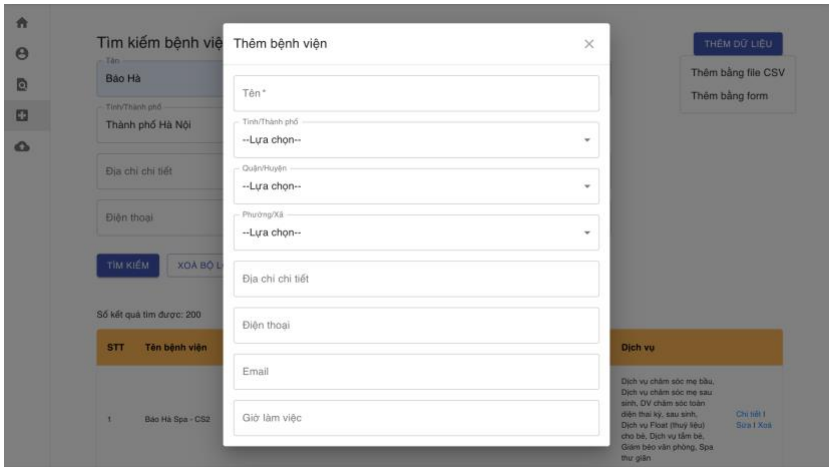
Giao diện sửa thông tin bệnh viện được mô tả trong hình 3.18.



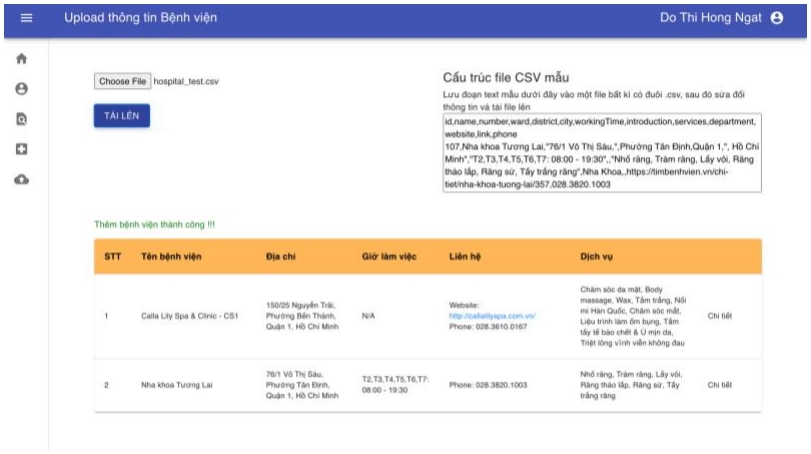
Hình 3.18: Giao diện sửa thông tin bệnh viện

Giao diện thêm thông tin bệnh viện bằng biểu mẫu mô tả như trong hình 3.19.

Giao diện thêm thông tin bệnh viện bằng file CSV mô tả như hình 3.20.

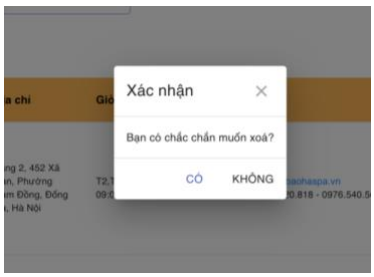


Hình 3.19: Giao diện thêm thông tin bệnh viện bằng biểu mẫu



Hình 3.20: Giao diện thêm thông tin bệnh viện bằng CSV

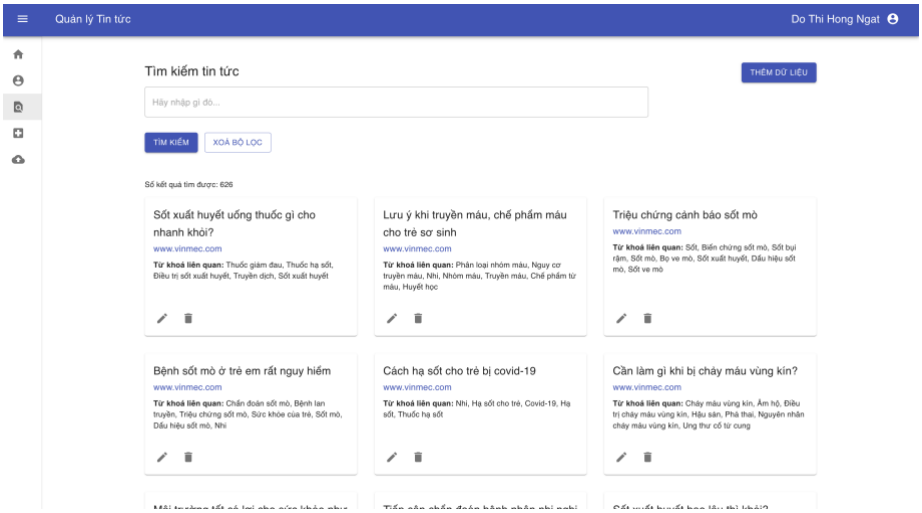
Hình 3.21 mô tả xác nhận khi quản trị viên xóa thông tin bệnh viện.



Hình 3.21: Giao diện xác nhận xóa thông tin bệnh viện

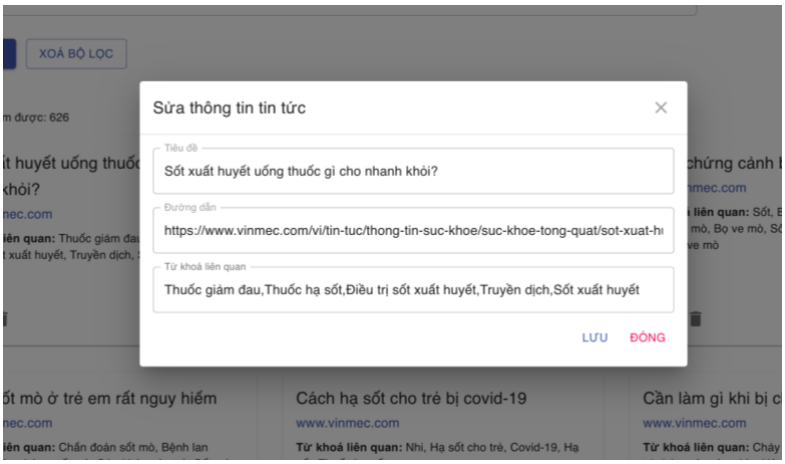
3.2.9. Giao diện quản lý bài báo

Giao diện tìm kiếm bài báo được mô tả như hình 3.22 dưới đây:



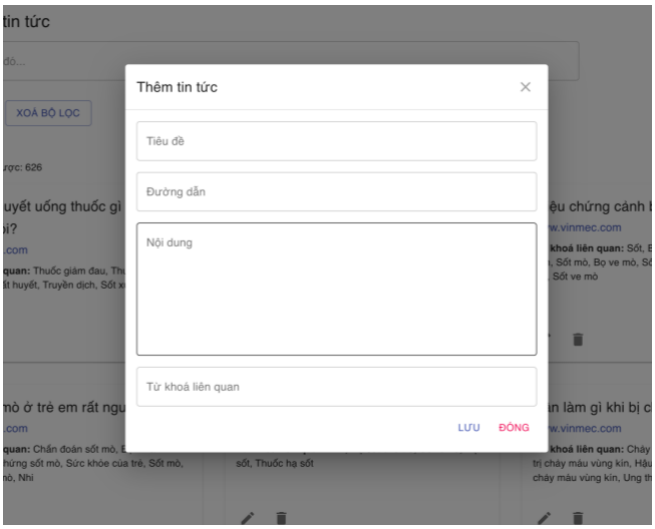
Hình 3.22: Giao diện tìm kiếm bài báo của quản trị viên

Trong hình 3.23 mô tả giao diện sửa thông tin bài báo.

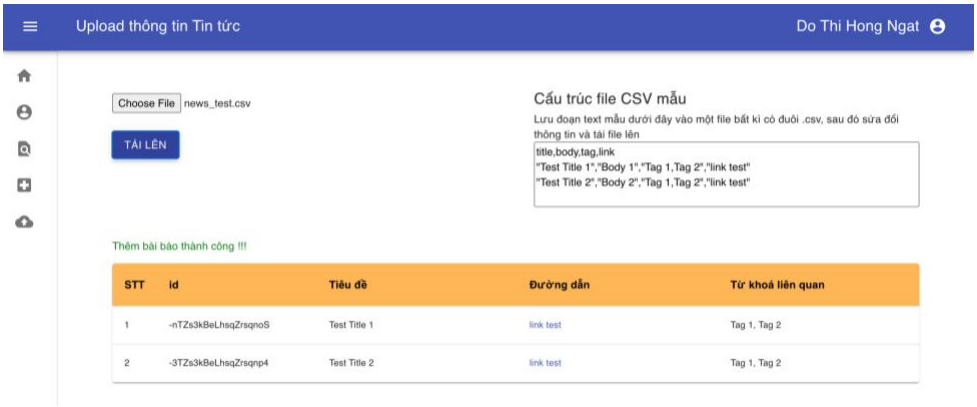


Hình 3.23: Giao diện sửa thông tin bài báo

Trong hình 3.24 mô tả giao diện thêm thông tin bài báo bằng biểu mẫu. hình 3.25 mô tả giao diện thêm thông tin bài báo bằng tệp tin CSV.

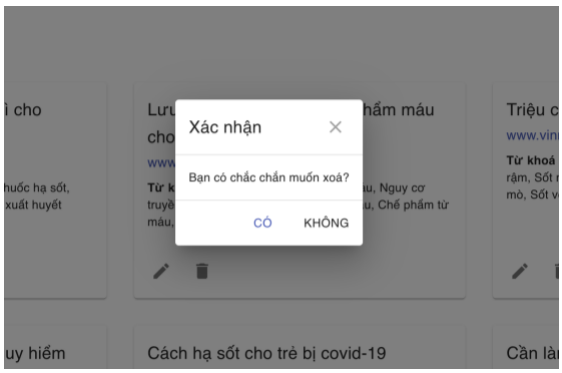


Hình 3.24: Giao diện thêm thông tin bài báo bằng biểu mẫu



Hình 3.25: Giao diện thêm thông tin bài báo bằng CSV

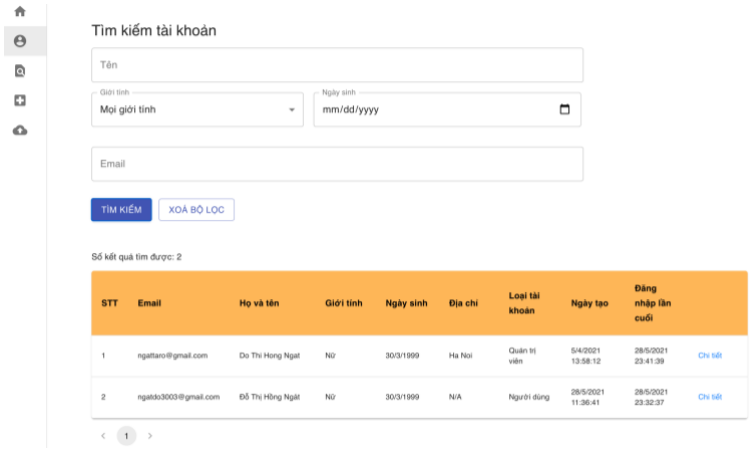
Trong hình 3.26 mô tả giao diện xác nhận quản trị viên xóa thông tin bài báo.



Hình 3.26: Giao diện xác nhận xóa thông tin bài báo

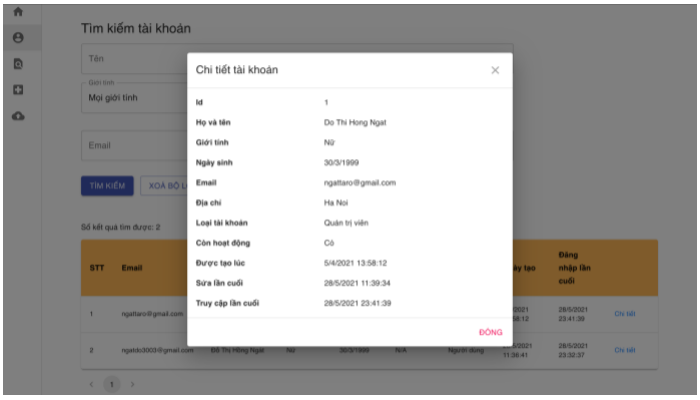
3.3.10. Giao diện quản lý người dùng

Giao diện tìm kiếm thông tin người dùng được mô tả như hình 3.27 dưới đây:



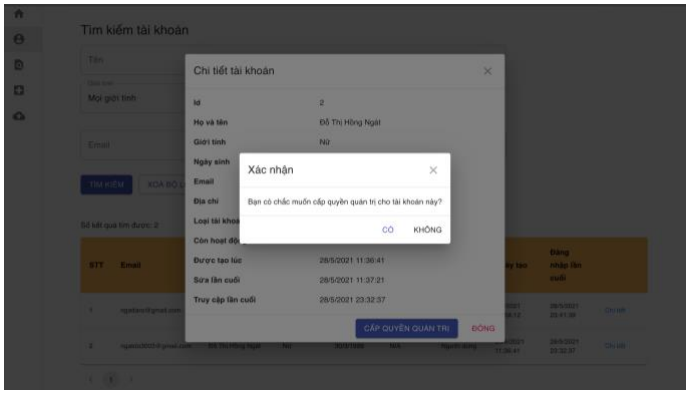
Hình 3.27: Giao diện tìm kiếm thông tin người dùng

Giao diện xem thông tin chi tiết tài khoản người dùng mô tả trong hình 3.28.



Hình 3.28: Giao diện quản trị viên xem chi tiết thông tin người dùng

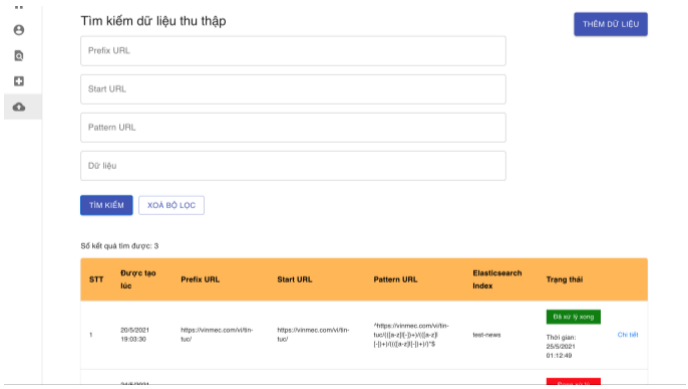
Giao diện cấp quyền quản trị viên cho người dùng mô tả trong hình 3.29.



Hình 3.29: Giao diện xem cấp quyền quản trị viên cho người dùng

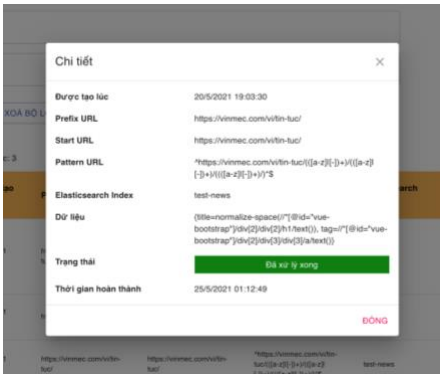
3.3.11. Giao diện quản lý dữ liệu thu thập

Giao diện tìm kiếm lịch sử thu thập dữ liệu được mô tả như hình 3.30, quản trị viên sau khi gửi yêu cầu thu thập dữ liệu sẽ có thể theo dõi trạng thái của yêu cầu.



Hình 3.30: Giao diện tìm kiếm lịch sử thu thập dữ liệu

Quản trị viên có thể xem lại yêu cầu thu thập dữ liệu thông qua chức năng xem chi tiết được mô tả như hình 3.31 dưới đây



Hình 3.31: Giao diện xem chi tiết yêu cầu thu thập

Giao diện tạo yêu cầu thu thập dữ liệu được mô tả như hình 3.32 dưới đây:

Hình 3.32: Giao diện tạo yêu cầu thu thập dữ liệu

3.4. Đánh giá hệ thống

Hệ thống tự động tổng hợp thông tin về dịch bệnh đã đáp ứng các yêu cầu của hệ thống đề ra trong mục 3.1.

Ưu điểm của hệ thống:

- Hệ thống tự động tổng hợp thông tin về dịch bệnh tổng hợp dữ liệu từ các trang web tin cậy, chính thống, đầy đủ. Ngoài ra dữ liệu được cập nhật liên tục tự động theo chu kỳ được định sẵn, hoặc thủ công do quản trị viên.

- Hệ thống tự động tổng hợp thông tin về dịch bệnh có giao diện trực quan và dễ sử dụng.

- Hệ thống có chức năng hoạt động chính xác với độ trễ đúng như thiết kế đặt ra, ổn định và người sử dụng có thể truy cập mọi lúc mọi nơi chỉ cần có Internet.

Nhược điểm của hệ thống:

- Hệ thống còn một số hạn chế về mặt chức năng như: chức năng đăng tải thông tin bài báo, bệnh viện chỉ nhận định dạng CSV, chưa hỗ trợ các loại định dạng khác, ngoài ra độ lớn của tệp tin cũng không quá 500MB.

- Email gửi thông tin mã xác thực chưa được thiết kế đẹp, chuyên nghiệp.

- Hiện tại hệ thống chỉ thu thập dữ liệu tại 2 trang web https://timbenhvien.vn và https://www.vinmec.com như mô tả bên trên nên chưa có sự đa dạng dữ liệu.

- Hệ thống tự động tổng hợp thông tin về dịch bệnh được xây dựng trong thời gian khá ngắn nên giao diện chưa được đẹp mắt. Hệ thống cần phải tiếp tục hoàn thiện.

3.5. Kết luận chương 3

Trong chương 3 đề án đã xây dựng hệ thống tự động tổng hợp thông tin về dịch bệnh theo kiến trúc tiểu dịch vụ sử dụng Scrapy framework và các công cụ lập trình web với dữ liệu dựa trên hai trang web https://timbenhvien.vn và https://www.vinmec.com. Trên cơ sở đó, đề án đề xuất giải pháp triển khai hệ thống phù hợp trong hoạt động quản lý thông tin, cung cấp công cụ tìm kiếm cho người dân trên toàn quốc.

KẾT LUẬN

Các kết quả đạt được của đề án

Với mục tiêu nghiên cứu Scrapy framework và xây dựng hệ thống tự động tổng hợp thông tin về dịch bệnh, đề án đã đạt được một số kết quả sau đây:

- Khảo sát tổng quan về hệ thống tổng hợp thông tin, thực tế triển khai hệ thống tổng hợp thông tin trên thế giới và tại Việt Nam cùng các các vấn đề liên quan.

- Khảo sát tổng quan về xu hướng phát triển Hệ thống tự động tổng hợp thông tin về dịch bệnh trên thế giới và tại Việt Nam.

- Nghiên cứu về Scrapy framework.

- Nghiên cứu một số công nghệ phát triển web: Kiến trúc tiểu dịch vụ, thư viện Redux, Spring framework, Flask framework, MySQL, Elasticsearch

- Tiến hành phân tích và thiết kế hệ thống hệ thống tự động tổng hợp thông tin về dịch bệnh. Xây dựng được mẫu đặc tả để trích xuất thông tin về dịch bệnh trên trang web https://www.vinmec.com/

- Tiến hành triển khai và xây dựng hệ thống tự động tổng hợp thông tin về dịch bệnh và trích xuất dữ liệu thành công trên hai trang web https://timbenhvien.vn và https://www.vinmec.com.

Các kết quả nghiên cứu của đề án có thể sử dụng như một tài liệu tham khảo trong quá trình nâng cấp và mở rộng của hệ thống tự động tổng hợp thông tin.

Hướng phát triển tiếp theo

Tiếp tục nghiên cứu, đề xuất và triển khai thêm chức năng như tích hợp với các hệ thống có sẵn ở bệnh viện triển khai để giảm thiểu việc quản lý, kết nối trực tuyến giữa bác sỹ, nhân viên y tế và người bệnh, giải đáp trực tuyến 24/7.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

Tiếng Anh

[1] Bing Liu, Robert Grossman, Yanhong Zhai (2003), "Mining data records in Web pages", *KDD*, pp. 601 - 606.

[2] Bugl, Daniel (2017), *Learning Redux: Write Maintainable, Consistent, and Easy-to-Test Web Applications*, Packt Publishing, 327 trang.

[3] Cherny, Boris (2019), *Programming TypeScript: Making Your JavaScript Applications Scale*, O'Reilly Media, 324 trang.

[4] Eckel, Bruce (2003), *Thinking in Java 3rd Edition*, Pearson Education, Inc., 1119 trang.

[5] Grinberg, Miguel (2018), *Flask Web Development: Developing Web Applications with Python*, O'Reilly Media, 315 trang.

[6] Hajba, Gábor László (2018), *Website Scraping with Python: Using BeautifulSoup and Scrapy*, Apress, 246 trang.

[7] Lutz, Mark (2013), *Learning Python 5th Edition*, O'Reilly Media, 1648 trang.

[8] Newman, Sam (2019), *Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith*, O'Reilly Media, 355 trang.

[9] Tahaghoghi, Seyed M.M. (Saied), và Hugh Williams (2006), *Learning MySQL: Get a Handle on Your Data*, O'Reilly Media, 420 trang.

Tham khảo từ Internet

[10] https://spring.io/projects/spring-framework, truy nhập ngày 10/11/2023.

[11] https://www.elastic.co/, truy nhập ngày 20/11/2023.

[12] https://www.scrapy.org/, truy nhập ngày 20/11/2023.

[13] https://www.w3schools.com/python/python_regex.asp, truy nhập ngày 5/12/2023.

[14] https://www.selenium.dev/documentation/, truy nhập ngày 17/12/2023.

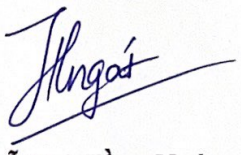
BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án qua phần mềm DoIT một cách trung thực và đạt kết quả mức độ tương đồng 5% toàn bộ nội dung đề án. Bản đề án kiểm tra qua phần mềm là bản cứng đề án đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu toàn bộ các hình thức kỷ luật theo quy định hiện hành của Học viện.

Hà nội, ngày 24 tháng 02 năm 2024

Học viên cao học

(ký và ghi rõ họ tên)



Đỗ Thị Hồng Ngát

KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	Đề án_Đỗ Thị Hồng Ngát_K22_220224
Tác giả:	Ngát Đỗ
Điểm trùng lặp:	5
Thời gian tải lên:	14:55 24/02/2024
Thời gian sinh báo cáo:	14:59 24/02/2024
Các trang kiểm tra:	70/70 trang

Kết quả kiểm tra trùng lặp

Có 5% nội dung trùng lặp

Có 95% nội dung không trùng lặp

Có 0% nội dung người dùng loại trừ

Có 0% nội dung hệ thống bỏ qua

Người hướng dẫn khoa học

Vũ Văn Thuỷ

Học viên cao học

Đỗ Thị Hồng Ngát

Nguồn trùng lặp tiêu biểu

123docz.net tailieu.vn luanvan.moet.gov.vn

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP

Họ và tên học viên: Đỗ Thị Hồng Ngát
Chuyên ngành: Khoa học máy tính
Khóa: 2022

Tên đề tài: Nghiên cứu xây dựng hệ thống tự động tổng hợp thông tin về dịch bệnh sử dụng Scrapy Framework

Người hướng dẫn khoa học: TS. Vũ Văn Thoả

Ngày bảo vệ: 20/03/2024

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Hiệu chỉnh lỗi trình bày, cấu trúc	Học viên đã rà soát, chỉnh sửa lỗi trình bày, cấu trúc
2	Làm rõ nội dung tổng hợp thông tin về dịch bệnh	Học viên đã bổ sung nội dung để làm rõ việc hệ thống tổng hợp thông tin về dịch bệnh

Hà Nội, ngày 27 tháng 03 năm 2024

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
GS.TS. Từ Minh Phương	TS. Trần Tiến Công	TS. Vũ Văn Thoả	Đỗ Thị Hồng Ngát

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 289/QĐ-HV ngày 06 tháng 03 năm 2024 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi 10... giờ 45.. phút, ngày 20 tháng 03 năm 2024 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Đỗ Thị Hồng Ngát**

Tên đề án tốt nghiệp: **Nghiên cứu xây dựng hệ thống tự động tổng hợp thông tin về dịch bệnh sử dụng Scrapy framework**

Chuyên ngành: **Khoa học máy tính** Mã số: **8.48.01.01**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ..5...../ 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HĐ	GHI CHÚ
1	**GS.TS. Từ Minh Phương**	Chủ tịch	
2	**TS. Trần Tiến Công**	Thư ký	
3	**PGS.TS. Phan Xuân Hiếu**	Phản biện 1	
4	**PGS.TS. Hoàng Xuân Dậu**	Phản biện 2	
5	**PGS.TS. Nguyễn Long Giang**	Uỷ viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên (Có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

- giải thích các hàng và các kết nối giữa các thành phần hệ thống
- mô tả thực thể liên kết ở trang 25
- nếu cần bổ sung thêm website khác và có thuật lập thì xử lý thế nào ?

7. Trả lời của học viên:

- Học viên giải thích về thiết kế cơ sở dữ liệu và cách thiết kế các khóa, các liên kết của cơ sở dữ liệu
- cần thu thập thêm thì cần quá trình thêm chưa gọn nếu từng lớp thì có thể xét đến hướng dẫn, chuẩn kiến, tra để thống nhất nội dung

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo).*

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: PGS.TS. Hoàng Xuân Dậu
  2. Uỷ viên Ban kiểm phiếu: TS. Trần Tiến Công
  3. Uỷ viên Ban kiểm phiếu: PGS.TS. Phan Văn Hiến
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 8

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:

- chỉnh sửa lại các lỗi chính tả theo góp ý của các thành viên hội đồng
- làm rõ nội dung tổng hợp về dịch bệnh

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên: .....

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS.....

Buổi làm việc kết thúc vào..... cùng ngày.

Chủ tịch	**Thư ký**
GS.TS. Từ Minh Phương	**TS. Trần Tiến Công**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

(*Dùng cho người phản biện*)

Tên đề tài đề án tốt nghiệp: Nghiên cứu xây dựng hệ thống tự động tổng hợp thông tin về dịch bệnh sử dụng Crapy framework

Chuyên ngành: Khoa học máy tính

Mã chuyên ngành: 8.48.01.01

Họ và tên học viên: Đỗ Thị Hồng Ngát

Họ và tên người nhận xét: Phan Xuân Hiếu

Học hàm, học vị: Phó Giáo sư, Tiến sĩ

Chuyên ngành: Khoa học thông tin và máy tính

Cơ quan công tác: Trường ĐH Công nghệ, ĐHQG Hà Nội

Số điện thoại: 0948855916 E-mail: hieupx@vnu.edu.vn

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

Đề án tìm hiểu và xây dựng hệ thống tự động thu thập, tổng hợp thông tin về dịch bệnh từ Internet dựa vào các công cụ Scrapy cũng như các công nghệ lưu trữ dữ liệu như hệ quản trị CSDL MySQL, nền tảng tìm kiếm Elasticsearch, các nền tảng phát triển ứng dụng web như Spring framework, Flask framework, kiến trúc micro-service .v.v.

Đề án thuần tuý tìm hiểu và triển khai công nghệ, nhưng mục tiêu của đề án có ý nghĩa ứng dụng thực tiễn.

II/ Nội dung của đề án tốt nghiệp, các kết quả đã đạt được:

Đề án tìm hiểu và ứng dụng các công nghệ như hệ quản trị CSDL MySQL, nền tảng tìm kiếm Elasticsearch, các công nghệ phát triển web và giao diện web nhằm phát triển một hệ thống tự động thu thập, tổng hợp tin tức về dịch bệnh. Tuy nhiên nội dung luận án nặng về thiết kế cơ sở dữ liệu và các use-cases của người dùng. Thông thường hệ thống tổng hợp tin tức không cần lưu trữ người dùng, nhưng có thể đề án có những mong muốn phát triển xa hơn về kết nối thông tin bệnh viện, tư vấn trực tuyến .v.v. nên dành nhiều nội dung để thiết kế các kịch bản người dùng.

Phần thu thập và tổng hợp thông tin với công nghệ nguồn mở Scrapy chưa được phân tích nhiều, đặc biệt là việc thu thập và tổng hợp tin tức gần thời gian thực. Đây mới là trọng tâm của đề án nhưng chưa thấy học viên đầu tư đầy đủ.

III/ Những vấn đề cần giải thích thêm:

Đề án thuần tuý về công nghệ, cần phân tích nhiều hơn về tin tức, tổng hợp tin tức về dịch bệnh gần thời gian thực. Ngoài ra việc trực quan hoá theo địa lý, tình trạng, mật độ, lan truyền .v.v. cũng chưa được chú trọng. Cần làm thêm những phần đó.

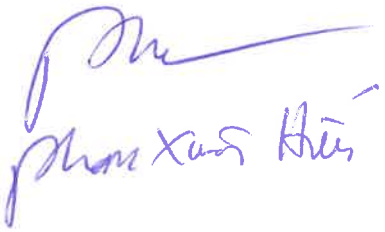
IV/ Kết luận:

Đồng ý cho phép học viên bảo vệ đề án tốt nghiệp.

Đồng ý (hoặc không đồng ý) cho phép học viên bảo vệ đề án tốt nghiệp.

Ngày 18 tháng 03 năm 2024

NGƯỜI NHẬN XÉT



BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
****

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SỸ

(Dùng cho người phản biện)

Tên đề tài đề án: Nghiên cứu xây dựng hệ thống tự động tổng hợp thông tin về dịch bệnh sử dụng Crapy Framework

Chuyên ngành: Khoa học máy tính

Mã số: 8.48.01.01

Tên học viên: Đỗ Thị Hồng Ngát

Họ và tên người nhận xét: Hoàng Xuân Dậu

Học hàm, học vị: PGS. Tiến sỹ, Giảng viên cao cấp

Chuyên ngành: Khoa học máy tính

Cơ quan công tác: Khoa ATTT, Học viện Công nghệ BC-VT

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

- Đề tài có ý nghĩa thực tiễn trong việc thu thập thông tin phục vụ cho phòng chống dịch bệnh.

II/ Về nội dung, chất lượng của đề án, các kết quả đạt được (so với đề cương đã được duyệt):

*** Nội dung đề án:**

Đề án tốt nghiệp của học viên gồm 3 chương chính:

Chương 1 giới thiệu tổng quan về hệ thống tổng hợp thông tin về dịch bệnh, xu hướng phát triển và một số công cụ và nền tảng sử dụng trong đề án như, Crapy framework, Flash framework, ngôn ngữ Python…

Chương 2 mô tả về việc phân tích và thiết kế hệ thống tổng hợp thông tin về dịch bệnh.

Chương 3 mô tả việc xây dựng và thử nghiệm hệ thống.

*** Chất lượng đề án:** Trung bình

*** Các kết quả đạt được:** cơ bản đạt các kết quả theo đề cương được duyệt. Tuy vậy, thông tin thu thập được còn ít, chỉ từ 2 trang web và chưa thực sự có nội dung "tổng hợp".

III/ Những vấn đề cần giải thích thêm:

- Đề án bố cục thành 3 chương là hợp lý, tuy nhiên văn phong của đề án còn lủng củng, nhiều câu tối nghĩa, hoặc không có ý nghĩa, nhất là ở chương 1. Chẳng hạn, ở trang

5, đề án viết "Hệ thống back-end kết hợp sử dụng Spring framework[10] là một mã nguồn mở phát triển ứng dụng phổ biến cho Java Enterprise. Spring là một mã nguồn mở, có kích thước nhỏ, khoảng 2MB", "Hệ thống tự động tổng hợp thông tin về dịch bệnh lưu trữ dữ liệu về thông tin người dùng, lịch sử thu thập dữ liệu của quản trị viên thông qua MySQL[9] là một hệ thống…", "hệ thống lưu trữ dữ liệu bệnh viện, bài báo về dịch bệnh trên Elasticsearch [11] là một công cụ tìm kiếm (search-engine) rất mạnh mẽ"…

- Tên các chương 1, 3 cũng cần rà soát và đổi lại cho trong sáng hơn.
- Bổ sung danh mục các từ viết tắt.
- Đề án còn nhiều hình mờ.
- Hệ thống được thiết kế và cài đặt đơn giản, mới chỉ mang tính thử nghiệm về khả năng thu thập dữ liệu, khả năng ứng dụng còn hạn chế.
- Phần sơ đồ thực thể liên kết và mô tả các bảng dữ liệu còn một số vấn đề, như không rõ các liên kết giữa các thực thể, cũng như giữa các bảng được thực hiện thế nào.
- Luồng xử lý dữ liệu của hệ thống cũng chưa được mô tả rõ ràng.

Câu hỏi:

1. Mô tả liên kết giữa các thực thể sơ đồ thực thể liên kết tại Hình 2.4, trang 35.
2. Hiện đề án mới thu thập thông tin từ 2 website, nếu cần bổ sung các trang web mới làm nguồn dữ liệu thì thực hiện thế nào?
3. Nếu có nhiều mẫu thông tin thu thập bị trùng lặp thì hệ thống xử lý thế nào?

IV/ Kết luận:

Đề án đáp ứng các tiêu chuẩn của một đề án tốt nghiệp thạc sỹ kỹ thuật.

Đồng ý cho phép học viên bảo vệ đề án tốt nghiệp.

Hà nội, ngày 15 tháng 3 năm 2024

Người phản biện

(Ký ghi rõ họ tên)

Hoàng Xuân Dậu