

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI



BÁO CÁO TỔNG KẾT

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN
NĂM HỌC 2023 - 2024**

XÂY DỰNG HỆ THỐNG TRAO ĐỔI HỌC TẬP ONLINE CHO SINH VIÊN UTC2

Sinh viên thực hiện

1. Nguyễn Bình Minh	Lớp: CQ.62.CNTT	Khoa: Công nghệ thông tin
2. Nguyễn Thành Luân	Lớp: CQ.62.CNTT	Khoa: Công nghệ thông tin
3. Lê Minh Hoàng	Lớp: CQ.62.CNTT	Khoa: Công nghệ thông tin
4. Nguyễn Trần Công Lý	Lớp: CQ.62.CNTT	Khoa: Công nghệ thông tin
5. Trần Nhựt Nam	Lớp: CQ.62.CNTT	Khoa: Công nghệ thông tin

Người hướng dẫn: KS. Trần Quốc Khánh

Tp. Hồ Chí Minh, năm 2024

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI



BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN

NĂM HỌC 2023 - 2024

XÂY DỰNG HỆ THỐNG TRAO ĐỔI HỌC TẬP ONLINE CHO SINH VIÊN UTC2

Sinh viên thực hiện

1. Nguyễn Bình Minh Giới tính: Nam Dân tộc: Kinh

Lớp: CQ.62.CNTT Ngành học: Công nghệ thông tin Năm thứ: 3/4

2. Nguyễn Thành Luân Giới tính: Nam Dân tộc: Kinh

Lớp: CQ.62.CNTT Ngành học: Công nghệ thông tin Năm thứ: 3/4

3. Lê Minh Hoàng Giới tính: Nam Dân tộc: Kinh

Lớp: CQ.63.CNTT Ngành học: Công nghệ thông tin Năm thứ: 2/4

4. Nguyễn Trần Công Lý Giới tính: Nam Dân tộc: Kinh

Lớp: CQ.63.CNTT Ngành học: Công nghệ thông tin Năm thứ: 2/4

5. Trần Nhựt Nam Giới tính: Nam Dân tộc: Kinh

Lớp: CQ.63.CNTT Ngành học: Công nghệ thông tin Năm thứ: 2/4

Người hướng dẫn: KS. Trần Quốc Khánh

Tp. Hồ Chí Minh, năm 2024

MỤC LỤC

MỤC LỤC	i
DANH MỤC HÌNH ẢNH.....	iv
DANH MỤC CHỮ VIẾT TẮT.....	v
GIỚI THIỆU.....	1
CHƯƠNG 1. TỔNG QUAN.....	2
1.1. Lý do chọn đề tài	2
1.2. Mục tiêu nghiên cứu	3
1.3. Đối tượng và phạm vi nghiên cứu	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	4
2.1. Công nghệ sử dụng	4
2.1.1. Javascript.....	4
2.1.2. NodeJS.....	5
2.1.3. ExpressJS.....	6
2.1.4. ReactJS	7
2.1.5. Redux.....	9
2.1.6. MongoDB.....	10
2.2. Bcrypt và JSON Web Token trong bảo mật thông tin.....	13
2.2.1. Mã hóa mật khẩu người dùng bằng thuật toán Bcrypt.....	13
2.2.2. Sử dụng JSON Web Token để giao tiếp an toàn.....	14
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	16
3.1. Biểu đồ phân rã chức năng	16
3.1.1. Các thực thể và thuộc tính.....	17

3.2.	Biểu đồ ca sử dụng.....	17
3.3.	Thiết kế cơ sở dữ liệu	19
3.4.	Biểu đồ tuần tự của hệ thống	19
3.4.1.	Ca sử dụng đăng nhập	20
3.4.2.	Ca sử dụng đăng ký	21
3.4.3.	Ca sử dụng đăng bài viết và tài liệu	22
3.4.4.	Ca sử dụng thích bài viết và tài liệu	23
3.4.5.	Ca sử dụng bình luận bài viết.....	23
3.4.6.	Ca sử dụng theo dõi thành viên	24
3.4.7.	Ca sử dụng chỉnh sửa thông tin cá nhân.....	25
3.4.8.	Ca sử dụng nhấn tin.....	26
CHƯƠNG 4.	TRIỂN KHAI CHƯƠNG TRÌNH.....	27
4.1.	Giao diện Website của người dùng.....	27
4.1.1.	Giao diện trang chủ (thông báo trường).....	27
4.1.2.	Giao diện trang đăng nhập và đăng ký	27
4.1.3.	Giao diện trang tin tức – sự kiện trường	27
4.1.4.	Giao diện trang trao đổi.....	27
4.1.5.	Giao diện đăng bài viết.....	27
4.1.6.	Giao diện trang tài liệu	27
4.1.7.	Giao diện đăng tài liệu	27
4.1.8.	Giao diện trang theo dõi	27
4.1.9.	Giao diện trang	27
4.1.10.	27
4.2.	Giao diện Website của quản trị viên.....	27
4.2.1.	Giao diện trang đăng nhập	27

4.2.2. Giao diện trang chủ (thống kê).....	27
KẾT LUẬN VÀ KIẾN NGHỊ.....	28
PHỤ LỤC	30
TÀI LIỆU THAM KHẢO	31

DANH MỤC HÌNH ẢNH

Hình 2.1 NodeJS	5
Hình 2.2 ReactJS	8
Hình 2.3 Hệ quản trị cơ sở dữ liệu MongoDB	11
Hình 2.4 Sơ đồ cách hoạt động JSON Web Token	15
Hình 3.1 Biểu đồ phân rã chức năng của hệ thống (Quản trị viên).....	16
Hình 3.2 Biểu đồ phân rã chức năng của hệ thống (Người dùng).....	17
Hình 3.3 Biểu đồ ca sử dụng tổng quát của hệ thống.....	18
Hình 3.4 ERD của hệ thống.....	19
Hình 3.5 Biểu đồ tuần tự ca sử dụng đăng nhập	20
Hình 3.6 Biểu đồ tuần tự ca sử dụng đăng ký	21
Hình 3.7 Biểu đồ tuần tự ca sử dụng đăng bài viết và tài liệu.....	22
Hình 3.8 Biểu đồ tuần tự ca sử dụng thích bài viết và tài liệu	23
Hình 3.9 Biểu đồ tuần tự ca sử dụng bình luận bài viết	23
Hình 3.10 Biểu đồ tuần tự ca sử dụng theo dõi thành viên	24
Hình 3.11 Biểu đồ tuần tự ca sử dụng chỉnh sửa thông tin cá nhân.....	25
Hình 3.12 Biểu đồ tuần tự ca sử dụng nhấn tin	26

DANH MỤC CHỮ VIẾT TẮT

KÝ HIỆU	THUẬT NGỮ ĐẦY ĐỦ	GIẢI THÍCH
AI	Artificial intelligence	Trí tuệ nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
ERD	Entity – Relationship Diagram	Sơ đồ mối quan hệ thực thể
UML	Unified Modeling Language	Ngôn ngữ mô hình mô tả hệ thống thông tin cấu trúc và cách hoạt động của thực thể

GIỚI THIỆU

Công Trong bối cảnh công nghệ thông tin ngày càng phát triển, đóng góp vào nhiều lĩnh vực của cuộc sống từ máy tính đến trí tuệ nhân tạo, mạng xã hội đã trở thành một phần không thể thiếu, tạo ra sự kết nối toàn cầu và truy cập thông tin nhanh chóng. Sự phát triển của công nghệ thông tin đã có ảnh hưởng sâu rộng đến nhiều lĩnh vực, bao gồm giáo dục, y tế, kinh doanh, và giao thông. Việc kết hợp giữa công nghệ thông tin và các ngành này đã tạo ra những tiến bộ đáng kể, mang lại lợi ích đáng kể cho cuộc sống hàng ngày và thúc đẩy phát triển kinh tế xã hội.

Tuy nhiên, cùng với những lợi ích đó, công nghệ thông tin cũng đối mặt với những thách thức, như bảo mật thông tin, quyền riêng tư, và sự phụ thuộc quá mức vào công nghệ. Giải quyết những thách thức này là cơ hội để phát triển bền vững và an toàn trong lĩnh vực công nghệ thông tin.

Nhu cầu sử dụng phần mềm trên nền tảng web ngày càng tăng, yêu cầu tính truy cập dễ dàng, đa nền tảng, di động và tương tác, cũng như cập nhật và bảo mật dễ dàng. Các ứng dụng trên nền tảng web bao gồm ứng dụng thương mại điện tử, công cụ quản lý dự án và làm việc nhóm, giáo dục trực tuyến, y tế trực tuyến, mạng xã hội, và ngân hàng trực tuyến.

Đối với trường Đại học Giao thông Vận tải – Phân hiệu tại TP. Hồ Chí Minh (UTC2), việc thiếu một hệ thống mạng xã hội hoặc diễn đàn trao đổi học tập đã tạo ra một khoảng cách trong việc truy cập thông tin và tương tác học tập giữa sinh viên và giữa sinh viên với nhà trường. Điều này có thể làm giảm sự hứng thú và sự hợp tác giữa sinh viên, gây trì hoãn trong quá trình học tập và phát triển cá nhân.

Do đó, việc xây dựng một hệ thống trao đổi học tập trực tuyến đầy đủ tính năng và hiệu quả, trở thành một nguồn tài nguyên không thể thiếu cho sinh viên UTC2, không chỉ giúp họ dễ dàng tiếp cận thông tin từ nhà trường mà còn thúc đẩy sự tương tác, hợp tác và học hỏi từ nhau, từ đó nâng cao hiệu quả học tập và sự hiểu biết chung của cả cộng đồng sinh viên.

CHƯƠNG 1. TỔNG QUAN

1.1. Lý do chọn đề tài

Trong thời đại số 4.0 đầy phát triển như ngày nay, mạng xã hội không chỉ là một phần không thể thiếu mà còn là một trung tâm linh hoạt của cuộc sống hàng ngày. Dưới sự lan tỏa mạnh mẽ của Internet, việc kết nối và giao tiếp trực tuyến đã trở nên mạnh mẽ và phổ biến hơn bao giờ hết. Từ các nền tảng lớn như Facebook, Twitter cho đến các cộng đồng học thuật như Studocu, Academia, mạng xã hội đã trở thành điểm đến không thể thiếu để chia sẻ thông tin, tương tác với bạn bè, gia đình, cũng như khám phá và chia sẻ kiến thức.

Một số trường đại học đã phát triển và triển khai các hệ thống mạng xã hội hoặc diễn đàn trao đổi học tập, Trường Đại học Giao thông Vận tải – Phân hiệu tại TP. Hồ Chí Minh (UTC2) hiện vẫn chưa có một nền tảng tương tự. Sự thiếu hụt này đã tạo ra một khoảng cách trong việc truy cập thông tin và tương tác học tập giữa các sinh viên với nhau và giữa sinh viên với nhà trường.

Trong khi các trường khác có thể tận dụng các hệ thống này để tăng cường sự kết nối và chia sẻ kiến thức, sinh viên của UTC2 đang gặp khó khăn trong việc truy cập vào nguồn thông tin và tài nguyên học tập từ cộng đồng của mình. Việc thiếu một môi trường trực tuyến tập trung vào học tập có thể làm giảm sự hứng thú và sự hợp tác giữa sinh viên, đồng thời làm trì hoãn quá trình học tập và phát triển cá nhân.

Trong cộng đồng sinh viên năng động tại UTC2, nhu cầu về một nền tảng trực tuyến để trao đổi thông tin học tập và chia sẻ kiến thức ngày càng trở nên trọng yếu hơn bao giờ hết. Mặc dù Internet đã cung cấp rất nhiều tài nguyên, việc tìm kiếm thông tin cụ thể vẫn là một thách thức đối với sinh viên. Việc có một nền tảng mạng xã hội riêng, tập trung vào nhu cầu học tập của sinh viên, không chỉ giúp họ dễ dàng tiếp cận thông tin của trường mà còn tạo điều kiện thuận lợi cho việc chia sẻ tài liệu, trao đổi kinh nghiệm học tập cùng bạn bè cùng trường.

Với nhận thức sâu rộng về nhu cầu này và sự phát triển không ngừng của mạng xã hội trong đời sống hiện đại, việc xây dựng một hệ thống trao đổi học tập trực tuyến cho sinh viên UTC2 không chỉ là một ý tưởng mà là một sự cần thiết. Một hệ thống như vậy không chỉ giúp sinh viên dễ dàng tiếp cận thông tin từ nhà trường và tài liệu

học tập mà còn thúc đẩy sự tương tác, hợp tác và học hỏi từ nhau, từ đó nâng cao hiệu quả học tập và sự hiểu biết chung của cả cộng đồng sinh viên. Đó chắc chắn là một bước tiến quan trọng trong việc thúc đẩy sự phát triển cá nhân và chất lượng giáo dục tại trường.

1.2. Mục tiêu nghiên cứu

Xây dựng một hệ thống trao đổi học tập online đầy đủ tính năng và hiệu quả, trở thành một nguồn tài nguyên không thể thiếu cho sinh viên UTC2 trong quá trình học tập và nghiên cứu. Mục tiêu cụ thể là:

- Tạo ra một nền tảng mạng xã hội học tập đa dạng và phong phú, cung cấp các tính năng như diễn đàn và thư viện tài liệu.
- Đảm bảo tính bảo mật và quản lý nội dung hiệu quả, giúp người dùng cảm thấy an tâm khi chia sẻ thông tin và tương tác trực tuyến.
- Khuyến khích sự tương tác và hợp tác giữa các sinh viên và giáo viên, tạo ra một môi trường học tập tích cực và chuẩn mực.

1.3. Đối tượng và phạm vi nghiên cứu

...

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Công nghệ sử dụng

2.1.1. Javascript

JavaScript là một ngôn ngữ lập trình phía client thông dịch (interpreted), đa mô hình và đa nền tảng. Ban đầu, JavaScript được phát triển để tạo ra các hiệu ứng động trên trang web, nhưng ngày nay, nó đã trở thành một ngôn ngữ lập trình phổ biến và mạnh mẽ, có thể được sử dụng để phát triển ứng dụng web đa dạng

❖ Ưu điểm:

- *Đa năng*: JavaScript có thể được sử dụng để phát triển ứng dụng web, ứng dụng di động, ứng dụng máy chủ và nhiều hơn nữa. Điều này cho phép lập trình viên sử dụng cùng một ngôn ngữ cho nhiều mục đích khác nhau. Hỗ trợ web tương tác hiệu quả với người truy cập.
- *Tích hợp tốt với HTML và CSS*: JavaScript được thiết kế để tương tác tốt với HTML và CSS, giúp tạo ra trang web động và tương tác người dùng một cách dễ dàng.
- *Thư viện và framework phong phú*: JavaScript có nhiều thư viện và framework mạnh mẽ như React, Angular và Vuejs giúp đơn giản hóa việc phát triển ứng dụng web và tăng hiệu suất của dự án.
- *Cú pháp dễ học*: JavaScript có cú pháp tương đối dễ học và đọc, giúp người mới bắt đầu nhanh chóng tiếp cận và phát triển.
- *Hỗ trợ mạnh mẽ cho xử lý sự kiện*: JavaScript cho phép xử lý sự kiện người dùng như nhấp chuột, gõ phím và hơn thế, giúp tạo ra trải nghiệm tương tác trên trang web.

❖ Nhược điểm:

- *An ninh*: JavaScript chạy trên trình duyệt của người dùng, điều này có thể tạo ra lỗ hổng bảo mật nếu không được kiểm soát cẩn thận. Điều này đòi hỏi lập trình viên phải thận trọng và tuân thủ các quy tắc về an ninh.
- *Khả năng mở rộng*: Một ứng dụng JavaScript lớn và phức tạp có thể khó khăn trong việc bảo trì và mở rộng. Tuy nhiên, việc sử dụng các framework như React hoặc Angular có thể giúp giảm đi vấn đề này.

- *Phụ thuộc vào trình duyệt:* JavaScript có thể chạy khác nhau trên các trình duyệt khác nhau và phiên bản trình duyệt, có thể tạo ra sự không nhất quán và khó khăn trong việc kiểm thử và tối ưu hóa ứng dụng.
- *Hiệu suất:* Trong một số trường hợp, JavaScript có thể chậm.

2.1.2. NodeJS

NodeJS là một nền tảng được phát triển độc lập trên V8 JavaScript Engine – trình thông thực thi mã JavaScript. Nhờ NodeJS mà việc xây dựng các ứng dụng Web trở nên đơn giản và dễ dàng hơn rất nhiều. Ngoài ra, NodeJS còn được biết tới là một mã nguồn mở và là một môi trường cho các máy chủ và ứng dụng mạng.

NodeJS cung cấp một kiến trúc hướng sự kiện (event-driven architecture) và non-blocking IO API, tối ưu hóa băng thông của một ứng dụng và có khả năng mở rộng. Node.js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Web server mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS. Mọi hàm trong Node.js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing).



Hình 2.1 NodeJS

❖ Ưu điểm:

- *Xử lý đa kết nối với một single-thread:* NodeJS có khả năng nhận và xử lý nhiều kết nối cùng một lúc chỉ bằng một luồng làm việc. Điều này giúp giảm tải RAM và tăng tốc độ hệ thống bằng cách không cần tạo ra các

luồng làm việc mới cho mỗi kết nối. Node.js cũng sử dụng non-blocking I/O của Javascript để tận dụng tối đa tài nguyên máy chủ mà không gây ra độ trễ như PHP.

- *JSON APIs*: Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Webs làm bằng JSON.
- *Ứng dụng trên một trang*: Với khả năng xử lý nhiều yêu cầu từ người dùng đồng thời thời gian phản hồi nhanh.
- *Shelling tools unix*: NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn tiến trình và trả ra 1 luồng khiến cho hiệu suất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.
- *Streamming Data (Luồng dữ liệu)*: NodeJS có khả năng xây dựng các proxy để quản lý và phân vùng dữ liệu lớn, đảm bảo tối ưu hóa hoạt động cho các luồng dữ liệu khác nhau.
- *Ứng dụng web và thời gian thực*: Với sự ra đời của các ứng dụng di động & HTML 5 nên NodeJS rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng nhắn tin, các dịch vụ mạng xã hội như Facebook, Twitter,...

❖ **Nhược điểm:**

- Khi xử lý các ứng dụng tốn tài nguyên CPU như mã hóa video, chuyển đổi tập tin, hoặc giải mã mã hóa, không nên sử dụng NodeJS do thời gian xử lý có thể tăng lên một chút khi cần thông qua trình biên dịch của NodeJS.
- So sánh với các ngôn ngữ khác như PHP, Ruby, Python, .NET trong việc phát triển ứng dụng web, NodeJS hiện vẫn đang ở giai đoạn phát triển và chưa thể vượt qua các ngôn ngữ khác. Tuy nhiên, với NodeJS, vẫn có khả năng xây dựng được các ứng dụng như mong đợi.

2.1.3. ExpressJS

ExpressJS là một framework được xây dựng trên nền tảng của NodeJS. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. ExpressJS hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

Tổng hợp một số chức năng chính của ExpressJS như sau:

- Thiết lập các lớp trung gian để trả về các yêu cầu HTTP.
- Định nghĩa bộ định tuyến cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số.

❖ **Ưu điểm:**

- *Hiệu suất cao:* Nhờ sử dụng DOM ảo, ReactJS có thể cải thiện hiệu suất của ứng dụng bằng cách chỉ hiển thị dữ liệu lại các thành phần (component) đã thay đổi, giúp tránh việc hiển thị dữ liệu lại toàn bộ trang mỗi khi có sự thay đổi.

❖ **Nhược điểm:**

- ReactJS chỉ phục vụ cho tầng View. React chỉ là một thư viện hiển thị, không phải là một framework MVC như các framework khác. Nó chỉ là một phần của Facebook giúp hiển thị dữ liệu ra phần giao diện. Do đó, React không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng không hỗ trợ 2-way binding hay Ajax.

2.1.4. ReactJS

React (còn được gọi là Reactjs hoặc React.js) là một thư viện JavaScript được phát triển bởi sự hợp tác giữa Facebook và Instagram. Nó cho phép các nhà phát triển web tạo ra giao diện người dùng một cách nhanh chóng. Phần giao diện của ReactJS thường được tạo ra bằng cách sử dụng chủ yếu các thành phần (components) chứa các thành phần cụ thể hoặc các thẻ HTML. Một trong những đặc điểm độc đáo của Reactjs là khả năng hiển thị dữ liệu không chỉ có thể thực hiện ở phía máy chủ mà còn ở phía máy khách.

Nó cũng sử dụng khái niệm là DOM ảo (Virtual DOM). DOM ảo tạo ra một bản sao cấu trúc dữ liệu của ứng dụng trong bộ nhớ. Sau đó, trong mỗi vòng lặp, nó kiểm tra những thay đổi và sau đó cập nhật lại những thay đổi đó trên DOM của trình duyệt một cách hiệu quả. Điều này cho phép chúng ta viết mã như thể toàn bộ trang được hiển thị lại, dù thực tế là ReactJS chỉ hiển thị dữ liệu lại những thành phần hoặc các thành phần con mà thực sự đã thay đổi.



Hình 2.2 ReactJS

❖ **Ưu điểm:**

- *Hiệu suất cao:* Nhờ sử dụng DOM ảo, ReactJS có thể cải thiện hiệu suất của ứng dụng bằng cách chỉ hiển thị dữ liệu lại các thành phần (component) đã thay đổi, giúp tránh việc hiển thị dữ liệu lại toàn bộ trang mỗi khi có sự thay đổi.
- *Tính tái sử dụng cao:* ReactJS sử dụng thành phần một cách cấu trúc hóa giao diện người dùng, giúp việc tái sử dụng đoạn mã trở nên dễ dàng hơn. Các thành phần có thể được tái sử dụng trong nhiều phần của ứng dụng mà không cần viết lại đoạn mã.
- *Cộng đồng lớn và hỗ trợ tốt:* ReactJS có một cộng đồng lớn và đầy đủ tài liệu, điều này giúp cho việc học và giải quyết vấn đề trở nên dễ dàng hơn. Ngoài ra, có nhiều công cụ hỗ trợ và thư viện bổ sung được phát triển bởi cộng đồng.
- *Tích hợp dễ dàng:* ReactJS có khả năng kết hợp với các thư viện và framework khác một cách linh hoạt, cho phép phát triển ứng dụng theo các nhu cầu cụ thể của dự án.
- *Cập nhật và duy trì dễ dàng:* Cách cấu trúc của ReactJS giúp cho việc cập nhật và duy trì ứng dụng trở nên đơn giản hơn. Các thành phần độc

lập giúp việc thêm, sửa đổi hoặc xóa bớt các tính năng trở nên dễ dàng hơn mà không ảnh hưởng đến các phần khác của ứng dụng.

❖ **Nhược điểm:**

- ReactJS chỉ phục vụ cho tầng View. React chỉ là một thư viện hiển thị, không phải là một framework MVC như các framework khác. Nó chỉ là một phần của Facebook giúp hiển thị dữ liệu ra phần giao diện. Do đó, React không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng không hỗ trợ 2-way binding hay Ajax.
- Tích hợp ReactJS vào các framework MVC truyền thống đòi hỏi phải cấu hình lại.
- React khá nặng so với các framework khác. React có kích thước tương tự với Angular (khoảng 35kb so với 39kb của Angular). Trong khi đó, Angular là một framework hoàn chỉnh.
- Khó tiếp cận cho người mới học Web.

2.1.5. Redux

Redux là một thư viện Javascript giúp quản lý trạng thái của ứng dụng. Nó được xây dựng dựa trên ý tưởng của kiến trúc Flux được Facebook giới thiệu, vì vậy thường được sử dụng cùng với React. Trong Redux, trạng thái của ứng dụng được lưu trữ trong Store. Mỗi thành phần có thể truy cập vào bất kỳ trạng thái nào từ Store khi cần.

❖ **Cơ chế hoạt động :**

- *State (Trạng thái):* Ứng dụng được quản lý bởi một trạng thái duy nhất (single source of truth). Trạng thái này được lưu trữ trong Redux Store.
- *Action:* Mọi thay đổi trạng thái trong ứng dụng đều được thực hiện thông qua các Action, là các đối tượng chứa thông tin về hành động và dữ liệu đi kèm.
- *Reducer:* Các Reducer xử lý các Action và cập nhật trạng thái mới của ứng dụng dựa trên các Action đó.
- *Store:* Redux Store lưu trữ trạng thái của ứng dụng và quản lý việc gửi đi và xử lý các Action.

❖ **Ưu điểm:**

- *Quản lý trạng thái dễ dàng*: Redux giúp tổ chức và quản lý trạng thái của ứng dụng một cách cấu trúc và có trật tự. Các trạng thái được lưu trữ trong một cây trạng thái duy nhất, giúp cho việc theo dõi và hiểu cấu trúc dữ liệu dễ dàng hơn.
- *Dễ kiểm tra*: Redux khuyến khích việc viết các Reducer và Action không phụ thuộc vào môi trường. Điều này làm cho việc kiểm tra và đảm bảo chất lượng của ứng dụng dễ dàng hơn.
- *Công cụ phát triển mạnh mẽ*: Redux có rất nhiều công cụ phát triển mạnh mẽ như Redux DevTools, giúp theo dõi các hành động, thay đổi trạng thái và thực hiện gỡ lỗi một cách hiệu quả.
- *Tích hợp dễ dàng*: Redux có thể tích hợp với nhiều framework và thư viện JavaScript khác, không chỉ giới hạn trong ReactJS. Điều này giúp tái sử dụng mã và tận dụng sức mạnh của Redux trong nhiều dự án.

❖ **Nhược điểm:**

- *Khối lượng mã tăng*: Redux có thể đòi hỏi viết thêm một số mã để cấu hình và quản lý trạng thái. Điều này có thể làm tăng khối lượng mã so với việc quản lý trạng thái trực tiếp trong ReactJS.
- *Gấp đôi công việc*: Redux yêu cầu bạn định nghĩa các Action và Reducer cho mỗi trạng thái. Điều này có thể tạo ra một số công việc trùng lặp và làm tăng độ phức tạp của mã.

2.1.6. MongoDB

❖ **Giới thiệu NoSQL:**

NoSQL ra đời như một giải pháp thay thế cho hạn chế của hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) trong các ứng dụng có yêu cầu về tốc độ, khả năng mở rộng và tính linh hoạt cao. Thay vì tuân theo mô hình dữ liệu cấu trúc của RDBMS, NoSQL sử dụng một cấu trúc dữ liệu linh hoạt hơn, thường dựa trên JSON và các cặp key-value.

NoSQL cho phép người quản trị cơ sở dữ liệu mở rộng hệ thống mà không cần phải quan tâm đến những vấn đề như tạo khóa ngoại, khóa chính hay kiểm tra ràng buộc dữ liệu. Thay vào đó, nó tập trung vào việc cung cấp hiệu suất cao và khả năng mở rộng dữ liệu một cách linh hoạt.

Tính toàn vẹn dữ liệu và giao dịch (transaction) được coi là những yếu tố không cần thiết trong môi trường NoSQL, khi mà sự chú trọng được đặt vào việc đạt được hiệu suất nhanh và khả năng mở rộng. Thay vào đó, NoSQL thường chấp nhận một số mất mát dữ liệu nhỏ trong quá trình xử lý để đảm bảo hiệu suất tốt nhất. Các công ty công nghệ lớn như Facebook và Google đã chọn sử dụng các hệ thống NoSQL như Cassandra và BigTable để xây dựng các ứng dụng quy mô lớn của mình. Điều này thể hiện sự phát triển và ứng dụng rộng rãi của NoSQL trong cộng đồng công nghệ hiện đại.

❖ **MongoDB:**

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, thuộc loại NoSQL, và được hàng triệu người sử dụng. MongoDB sử dụng mô hình cơ sở dữ liệu hướng tài liệu, trong đó dữ liệu được lưu trữ dưới dạng các tài liệu JSON thay vì bảng như trong cơ sở dữ liệu quan hệ, giúp tăng tốc độ truy vấn.

Trong cơ sở dữ liệu quan hệ, dữ liệu thường được tổ chức vào các bảng, trong khi đó MongoDB sử dụng khái niệm "collection" thay vì "bảng". Mỗi collection trong MongoDB tương ứng với một bảng trong hệ quản trị cơ sở dữ liệu quan hệ. Thay vì dùng "row" như trong RDBMS, MongoDB sử dụng "document" để đại diện cho dữ liệu.

Các collection trong MongoDB được cấu trúc linh hoạt, không yêu cầu các dữ liệu tuân theo cấu trúc nhất định. Thông tin liên quan thường được lưu trữ cùng nhau trong một document, giúp truy cập và truy vấn dữ liệu nhanh chóng thông qua ngôn ngữ truy vấn của MongoDB.



Hình 2.3 Hệ quản trị cơ sở dữ liệu MongoDB

Các khái niệm chính MongoDB:

- Database (Cơ sở dữ liệu): Database là một không gian lưu trữ vật lý cho các collection. Mỗi database có danh sách riêng của các tệp hệ thống file. Một máy chủ MongoDB thường chứa nhiều database.
- Collection (Bộ sưu tập): Collection là một nhóm các documents trong MongoDB, tương đương với một bảng trong hệ thống quản lý cơ sở dữ liệu quan hệ. Mỗi collection thuộc về một database duy nhất. Collections không có schema cố định, có nghĩa là các documents trong collection có thể có các trường khác nhau. Thông thường, các documents trong một collection có mục đích hoặc liên quan đến nhau.
- Document (Tài liệu): Một document là một tập hợp các cặp key-value. Các document có schema linh hoạt, có nghĩa là các document trong cùng một collection không bắt buộc phải có cùng cấu trúc trường hoặc các trường giống nhau. Các trường thông thường trong các document của collection có thể chứa các loại dữ liệu khác nhau.

❖ **Ưu điểm:**

- Vì MongoDB lưu trữ dữ liệu dưới dạng file JSON, mỗi collection có thể chứa các document có kích thước và cấu trúc khác nhau, tạo ra sự linh hoạt trong lưu trữ dữ liệu.
- Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau và không cần sử dụng join như trong RDBMS. Điều này khiến cho việc thêm, xóa hoặc sửa dữ liệu trở nên đơn giản hơn và không cần mất thời gian kiểm tra ràng buộc dữ liệu.
- MongoDB dễ dàng mở rộng theo chiều ngang bằng cách thêm các node vào cluster. Một cluster trong MongoDB là một nhóm các node chứa dữ liệu giao tiếp với nhau.
- Trường dữ liệu "id" luôn được tự động đánh chỉ mục, giúp tăng hiệu suất truy vấn thông tin.
- Khi có một truy vấn dữ liệu, bản ghi được lưu trữ tạm thời trong bộ nhớ RAM, giúp tăng tốc độ truy vấn lần sau mà không cần phải đọc từ ổ cứng.
- Hiệu suất của MongoDB cao hơn đáng kể so với các hệ quản trị cơ sở dữ liệu quan hệ. Với lượng dữ liệu đủ lớn, tốc độ thêm của MongoDB có thể nhanh hơn đến 100 lần so với MySQL.

❖ **Nhược điểm:**

- MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với MongoDB thì phải hết sức cẩn thận.
- Tổn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.
- Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điều này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện.

2.2. Bcrypt và JSON Web Token trong bảo mật thông tin

2.2.1. Mã hóa mật khẩu người dùng bằng thuật toán Bcrypt

Trong thời đại hiện nay, bảo mật thông tin là vô cùng quan trọng. Nếu mật khẩu người dùng bị rò rỉ, những kẻ tấn công có thể sử dụng tài khoản đăng nhập để giả mạo người dùng thực hiện nhiều hành vi vi phạm. Ví dụ, trong cuộc điều tra do Keeper thực hiện, đã có 87% người từ 18 đến 30 tuổi đã phải tái sử dụng lại mật khẩu. Điều này có nghĩa là, nếu lỡ mật khẩu từ dịch vụ bạn đang sử dụng bị rò rỉ ra ngoài, sự ảnh hưởng không chỉ giới hạn đối với dịch vụ của bạn, mà những kẻ tấn công còn có thể truy cập trái phép vào bất kỳ dịch vụ nào được sử dụng bởi người dùng đó.

Cho dù chúng ta có cố gắng để thực hiện các biện pháp phòng ngừa nhưng cũng không thể đảm bảo cơ sở dữ liệu của mình là không thể xâm nhập. Vì vậy, thay vì việc nghĩ cách làm sao để tăng tính an toàn cho cơ sở dữ liệu, chúng ta có thể mã hóa mật khẩu của mình để tội phạm mạng không thể giải mã được nó. Để có thể mã hóa mật khẩu, chúng ta cần băm mật khẩu trước khi đưa nó vào cơ sở dữ liệu.

BCrypt là một thuật toán băm mật mã được sử dụng rộng rãi để lưu trữ mật khẩu một cách an toàn. Nó sử dụng thuật toán mã hóa Blowfish, được thiết kế bởi Bruce Schneier vào năm 1993. BCrypt đặc biệt phổ biến trong cộng đồng nhà phát triển web nhờ vào tính bảo mật mạnh mẽ của nó và hiệu suất trong việc bảo vệ dữ liệu nhạy cảm của người dùng.

BCrypt là thuật toán mà em sử dụng để mã hóa. Thuật toán BCrypt là một thuật toán hash mật khẩu được thiết kế để làm việc chậm hơn và tăng khả năng chống lại các cuộc tấn công brute-force. Nó được sử dụng phổ biến trong việc lưu trữ mật khẩu trong các hệ thống bảo mật.

2.2.2. Sử dụng JSON Web Token để giao tiếp an toàn

JSON Web Token (JWT) là một tiêu chuẩn mở (RFC 7519) xác định một cách nhỏ gọn và khép kín để truyền thông tin an toàn giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó được ký điện tử. JWT có thể được ký bằng cách sử dụng bí mật (với thuật toán HMAC) hoặc cặp khóa công khai / riêng tư bằng RSA hoặc ECDSA.

JSON Web Tokens bao gồm 3 phần được phân tách bằng dấu chấm (.) đó là Header, Payload, Signature. Do đó, JWT có cấu trúc như sau: xxxxx.yyyyy.zzzzz

Header: thường bao gồm hai phần: loại token (mã thông báo), là JWT và thuật toán ký đang được sử dụng, chẳng hạn như HMAC SHA256 hoặc RSA. Ví dụ:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Sau đó, JSON này được mã hóa Base64Url để tạo thành phần đầu tiên của JWT.

Payload: nó chứa các the claims (khẳng định). Claims là các tuyên bố về một thực thể (thường là người dùng) các dữ liệu bổ sung. Có 3 loại claims: registered, public và private claims. Ví dụ:

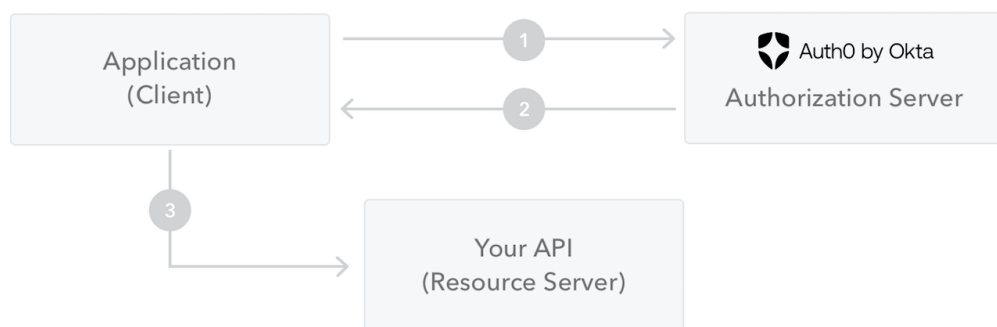
```
{
  "email": "user@gamil.com",
  "roleId": "12",
  "userId": "1"
}
```

Sau đó, JSON này được mã hóa Base64Url để tạo thành phần thứ hai của JWT.

Signature: Để tạo phần chữ ký, bạn phải lấy header được mã hóa, payload được mã hóa, một bí mật, thuật toán được chỉ định trong header và ký vào đó. Do bản thân Signature đã bao gồm cả header và payload nên Signature có thể dùng để kiểm tra tính toàn vẹn của dữ liệu khi truyền tải.

Trong quá trình xác thực, khi người dùng đăng nhập thành công bằng thông tin đăng nhập của họ, một JWT sẽ được trả về. Khi người dùng muốn truy cập vào một tài nguyên hay đường dẫn được bảo vệ, trình duyệt người dùng sẽ gửi JWT, thường là trong tiêu đề Authorization bằng lược đồ Bearer. Nội dung của tiêu đề sẽ là “Authorization: Bearer <token>”. Đó là cách JWT hoạt động.

Sơ đồ sau đây cho thấy cách JWT được lấy và sử dụng để truy cập API hoặc tài nguyên:



Hình 2.4 Sơ đồ cách hoạt động JSON Web Token

1. Ứng dụng hoặc máy khách yêu cầu ủy quyền cho máy chủ ủy quyền. Điều này được thực hiện thông qua một trong các luồng ủy quyền khác nhau.
2. Khi ủy quyền được cấp, máy chủ ủy quyền trả về mã thông báo truy cập cho ứng dụng.
3. Ứng dụng sử dụng mã thông báo truy cập để truy cập tài nguyên được bảo vệ (như API).

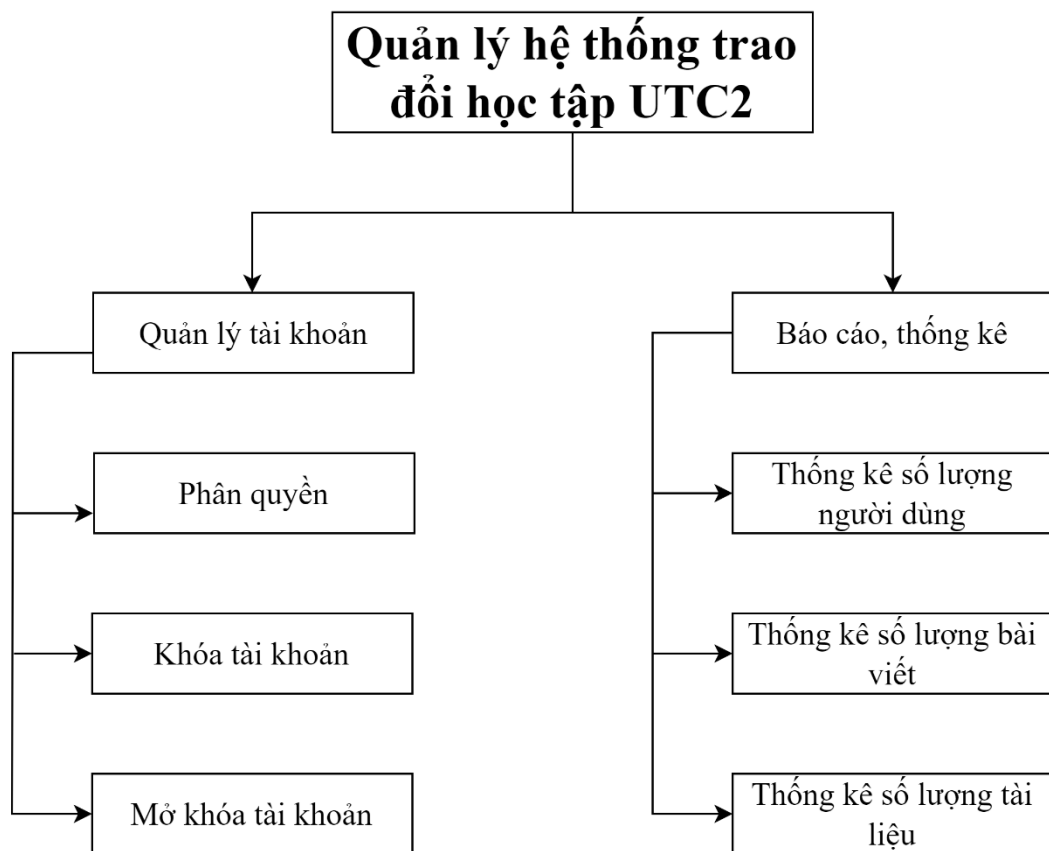
Xin lưu ý rằng với mã thông báo đã ký, tất cả thông tin chứa trong mã thông báo được hiển thị cho người dùng hoặc các bên khác, mặc dù họ không thể thay đổi nó. Điều này có nghĩa là bạn không nên đặt thông tin bí mật trong mã thông báo.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

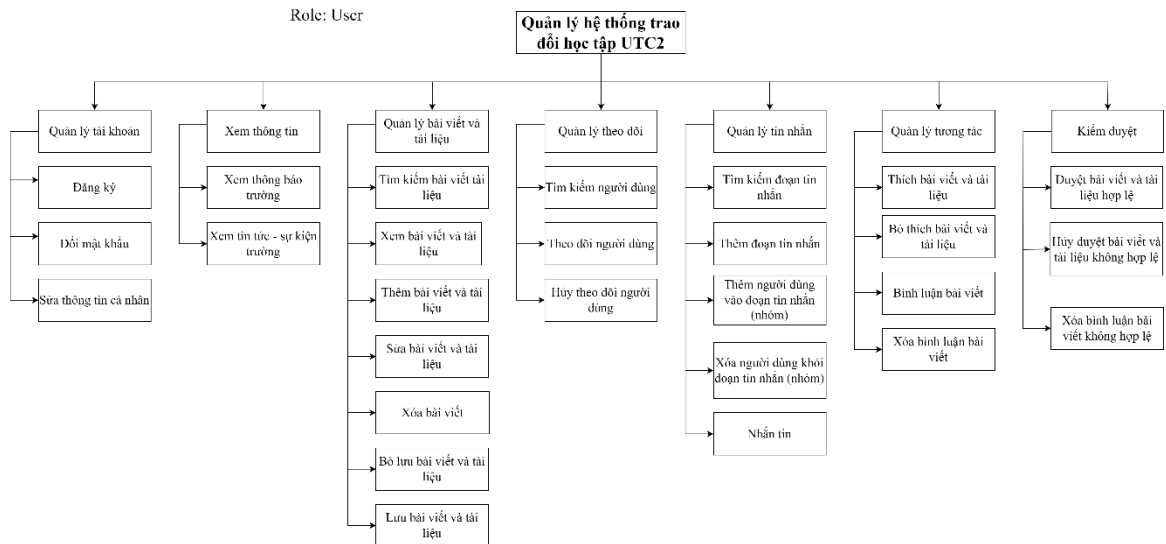
3.1. Biểu đồ phân rã chức năng

Mô hình phân rã chức năng (BFD) là công cụ biểu diễn việc phân rã có thứ bậc đơn giản các công việc cần thực hiện. Mỗi công việc được chia ra làm các công việc con, số mức chia ra phụ thuộc kích cỡ và độ phức tạp của hệ thống.

Role: Admin



Hình 3.1 Biểu đồ phân rã chức năng của hệ thống (Quản trị viên)

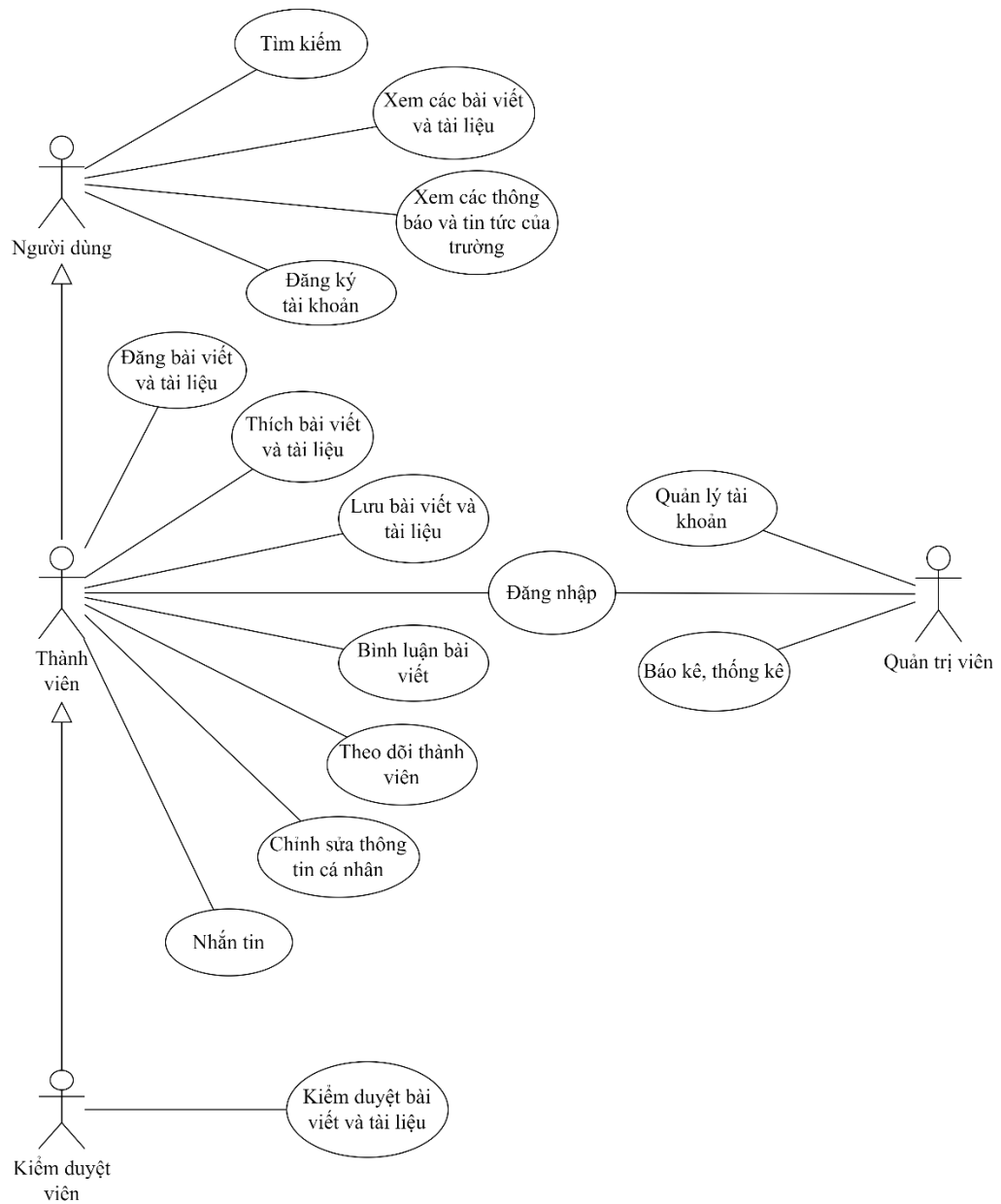


Hình 3.2 Biểu đồ phân rã chức năng của hệ thống (Người dùng)

3.1.1. Các thực thể và thuộc tính

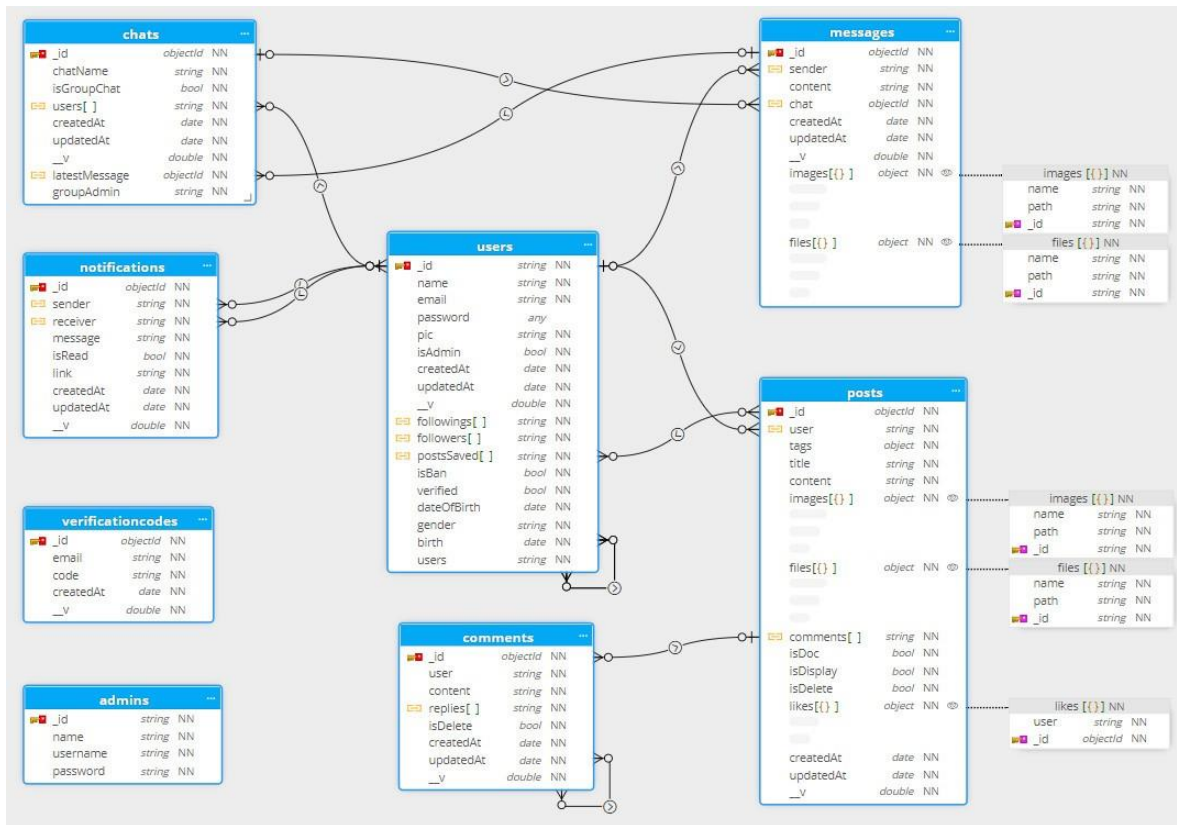
3.2. Biểu đồ ca sử dụng

Biểu đồ ca sử dụng là một biểu đồ có thể sử dụng để giải thích các quá trình, các hành động và các hành vi của người sử dụng đối với một hệ thống nào đó. Biểu đồ này bao gồm các ca sử dụng cũng như các loại người dùng khác nhau của hệ thống và thường đi kèm với một số loại biểu đồ khác. Các ca sử dụng được thể hiện bằng hình tròn hoặc hình elip, còn tác nhân thường được biểu diễn dưới dạng hình người.



Hình 3.3 Biểu đồ ca sử dụng tổng quát của hệ thống

3.3. Thiết kế cơ sở dữ liệu



Hình 3.4 ERD của hệ thống

ERD (Entity – Relationship Diagram, Sơ đồ mối quan hệ thực thể) là một dạng trực quan của cơ sở dữ liệu quan hệ. Mọi người sử dụng ERD để mô hình hóa và thiết kế cơ sở dữ liệu quan hệ.

Cơ sở dữ liệu quan hệ là một loại cơ sở dữ liệu lưu trữ và cung cấp quyền truy cập vào các điểm dữ liệu bằng cách tổ chức dữ liệu thành các hàng và cột, gọi chung tạo thành một bảng.

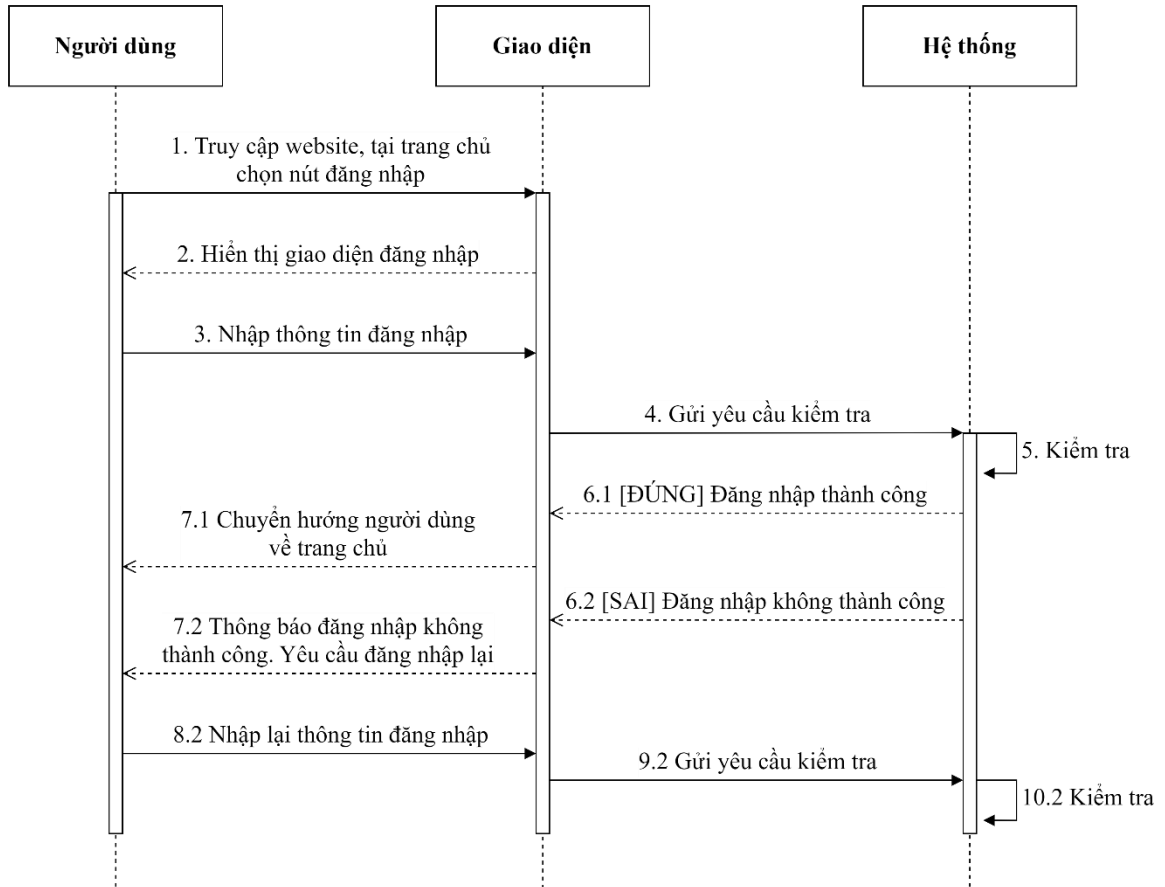
Sơ đồ ERD thường được sử dụng để thiết kế hoặc gỡ lỗi trong cơ sở dữ liệu quan hệ trong các lĩnh vực kỹ thuật phần mềm, hệ thống thông tin kinh doanh, giáo dục và nghiên cứu.

3.4. Biểu đồ tuần tự của hệ thống

Biểu đồ tuần tự là một loại sơ đồ UML đại diện cho cách thức và thứ tự các đối tượng trong một hệ thống tương tác với nhau. Biểu đồ tuần tự được sử dụng để thiết kế, ghi lại và xác nhận kiến trúc, giao diện và logic của các hệ thống bằng cách mô tả

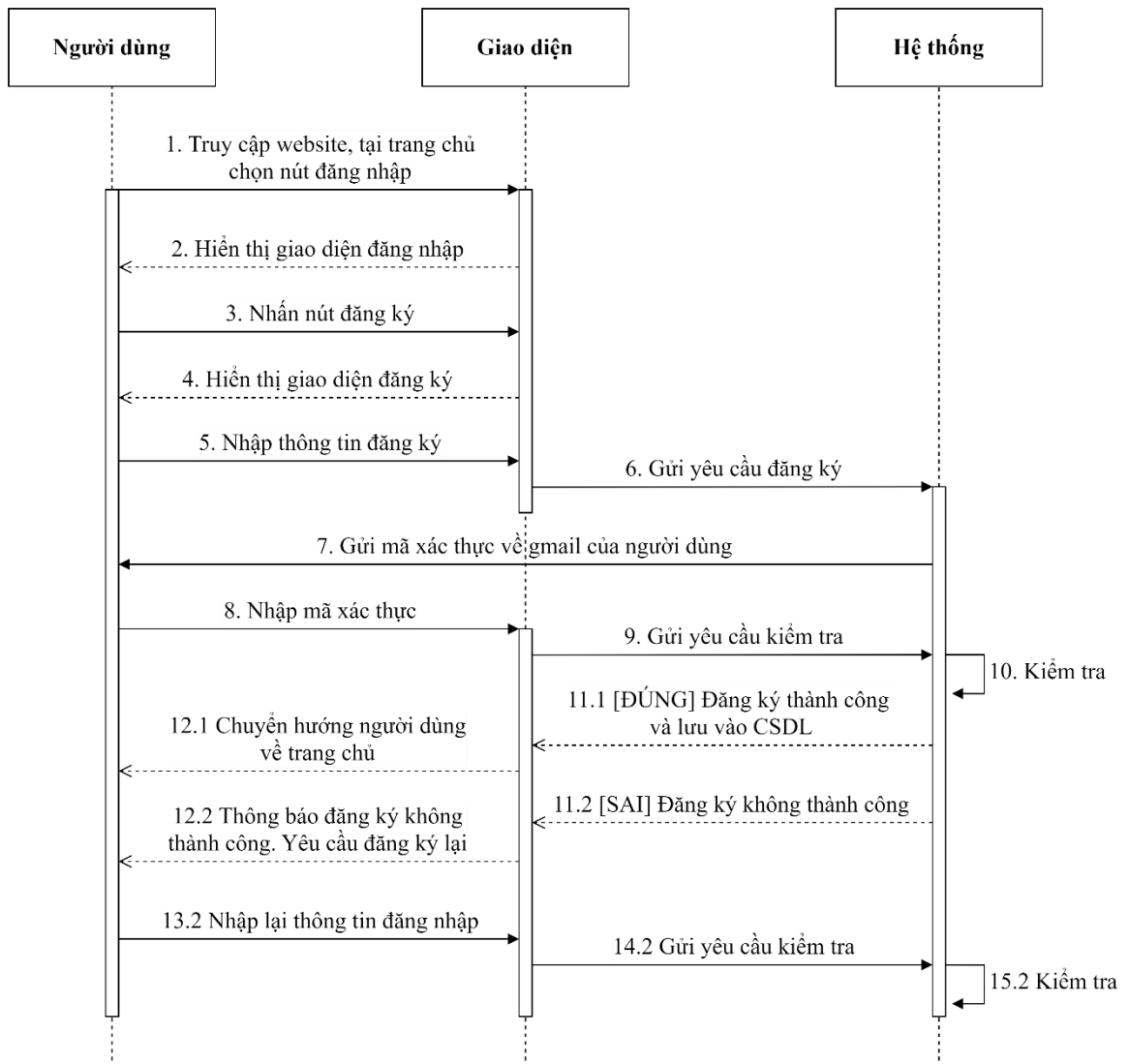
chuỗi các hành động cần được thực hiện để hoàn thành một nhiệm vụ. Nhóm sẽ trình bày một số biểu đồ tuần tự chính của hệ thống.

3.4.1. Ca sử dụng đăng nhập



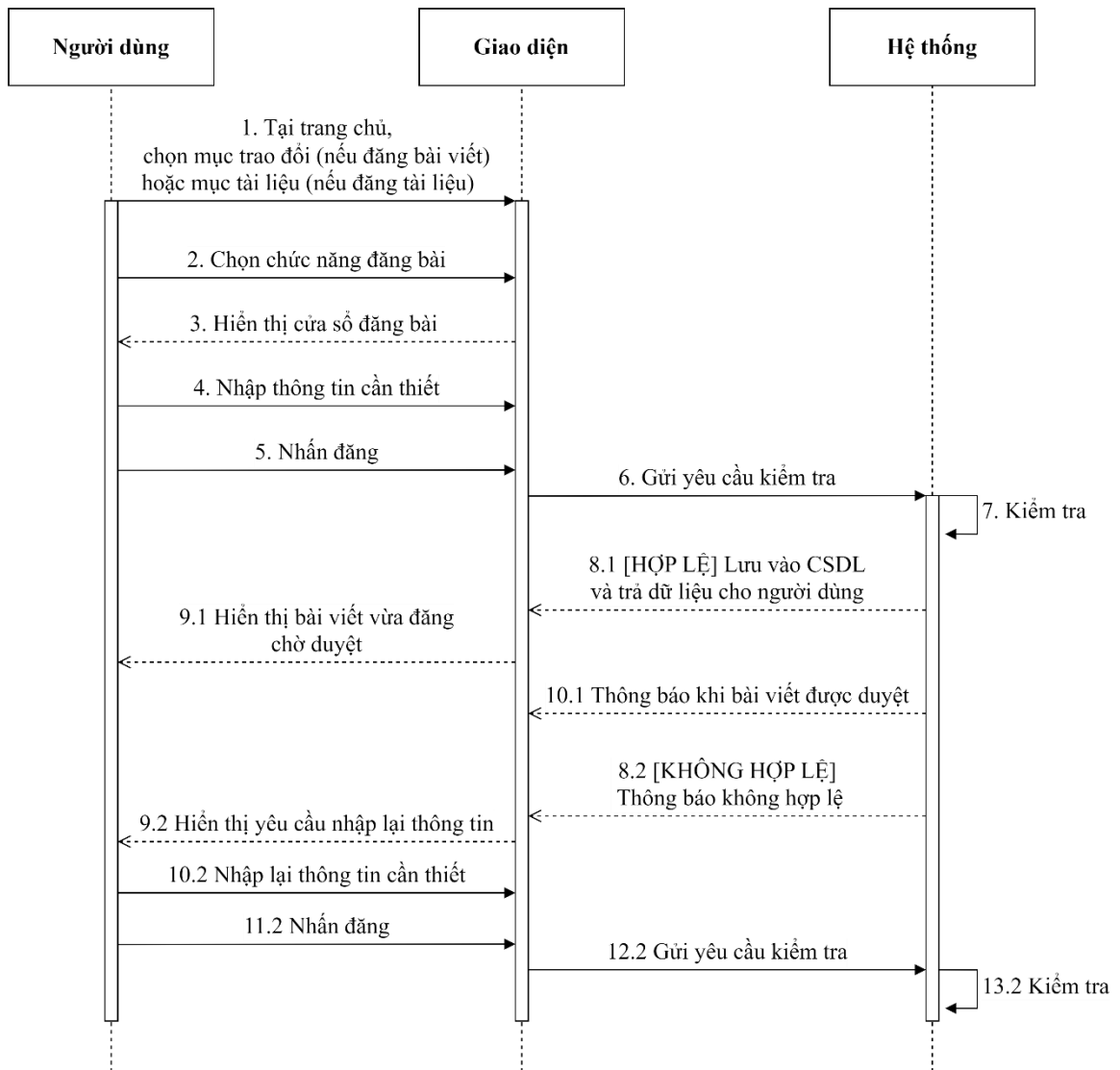
Hình 3.5 Biểu đồ tuần tự ca sử dụng đăng nhập

3.4.2. Ca sử dụng đăng ký



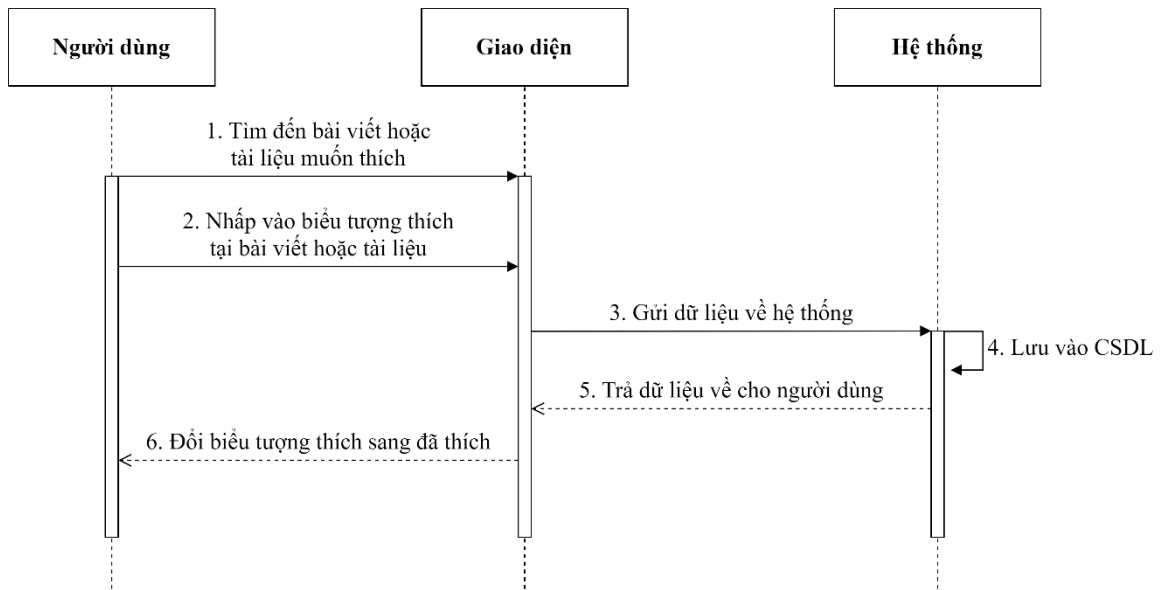
Hình 3.6 Biểu đồ tuần tự ca sử dụng đăng ký

3.4.3. Ca sử dụng đăng bài viết và tài liệu



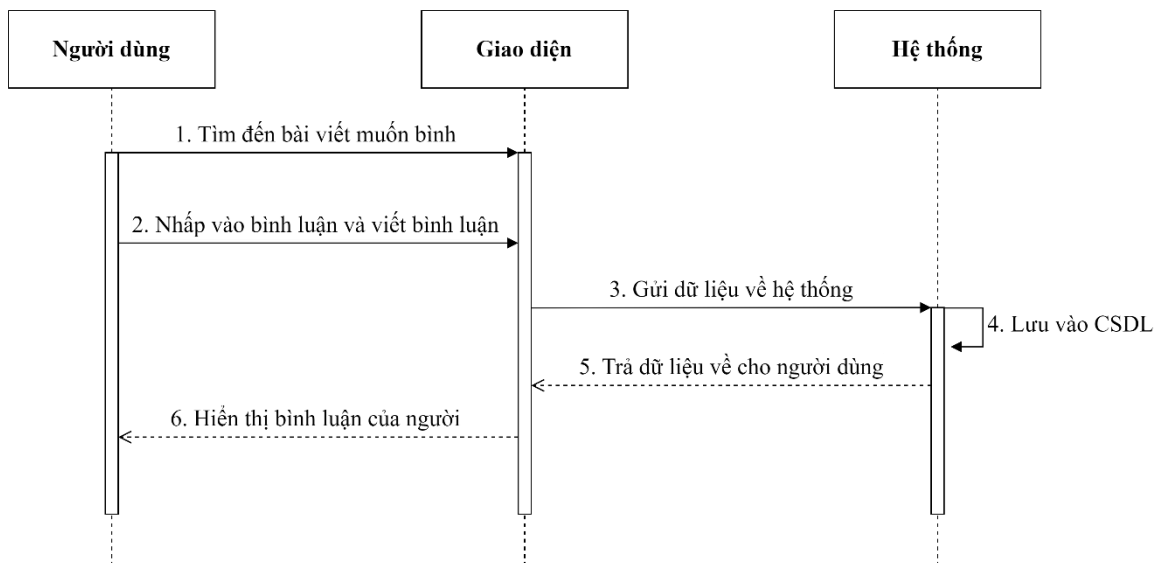
Hình 3.7 Biểu đồ tuần tự ca sử dụng đăng bài viết và tài liệu

3.4.4. Ca sử dụng thích bài viết và tài liệu



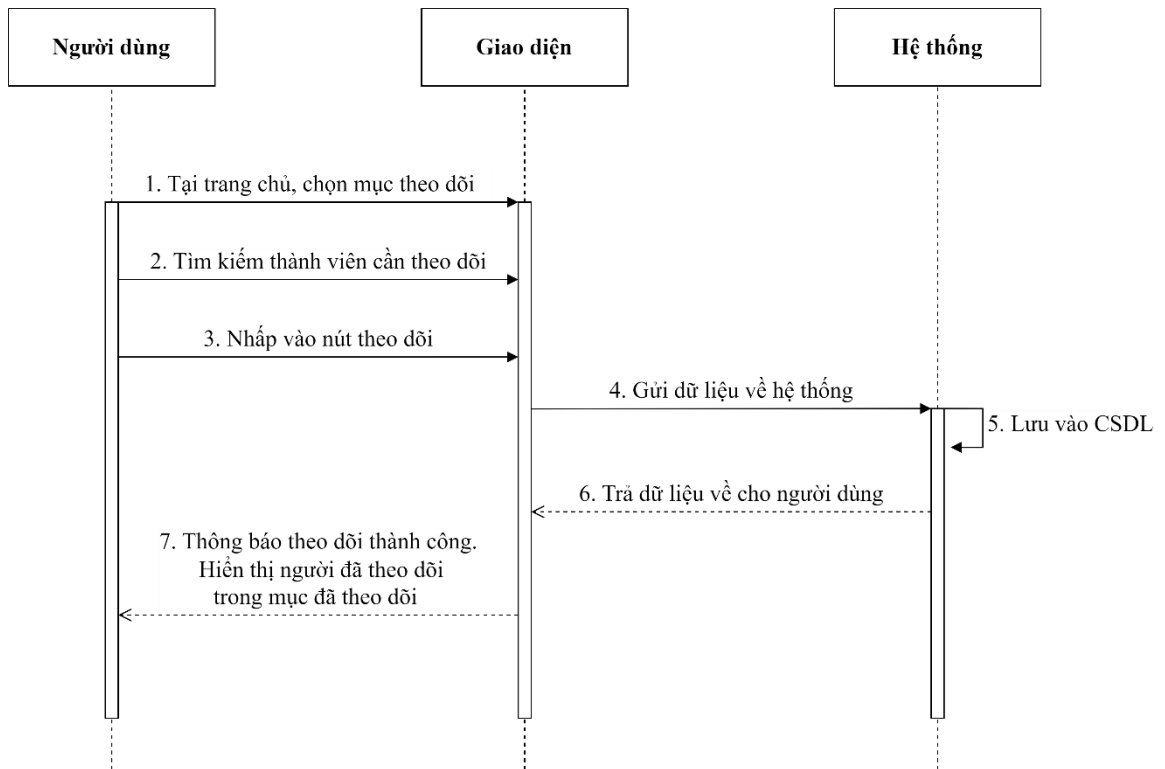
Hình 3.8 Biểu đồ tuần tự ca sử dụng thích bài viết và tài liệu

3.4.5. Ca sử dụng bình luận bài viết



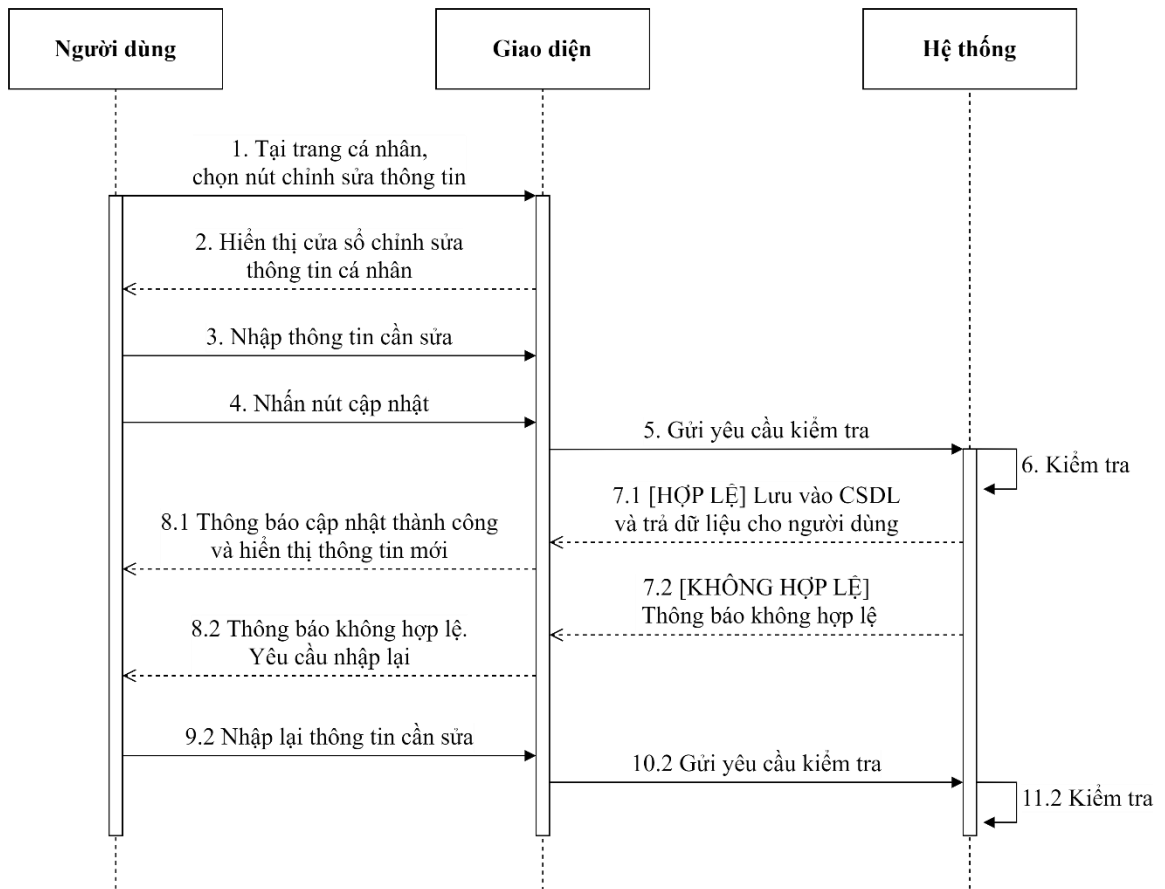
Hình 3.9 Biểu đồ tuần tự ca sử dụng bình luận bài viết

3.4.6. Ca sử dụng theo dõi thành viên



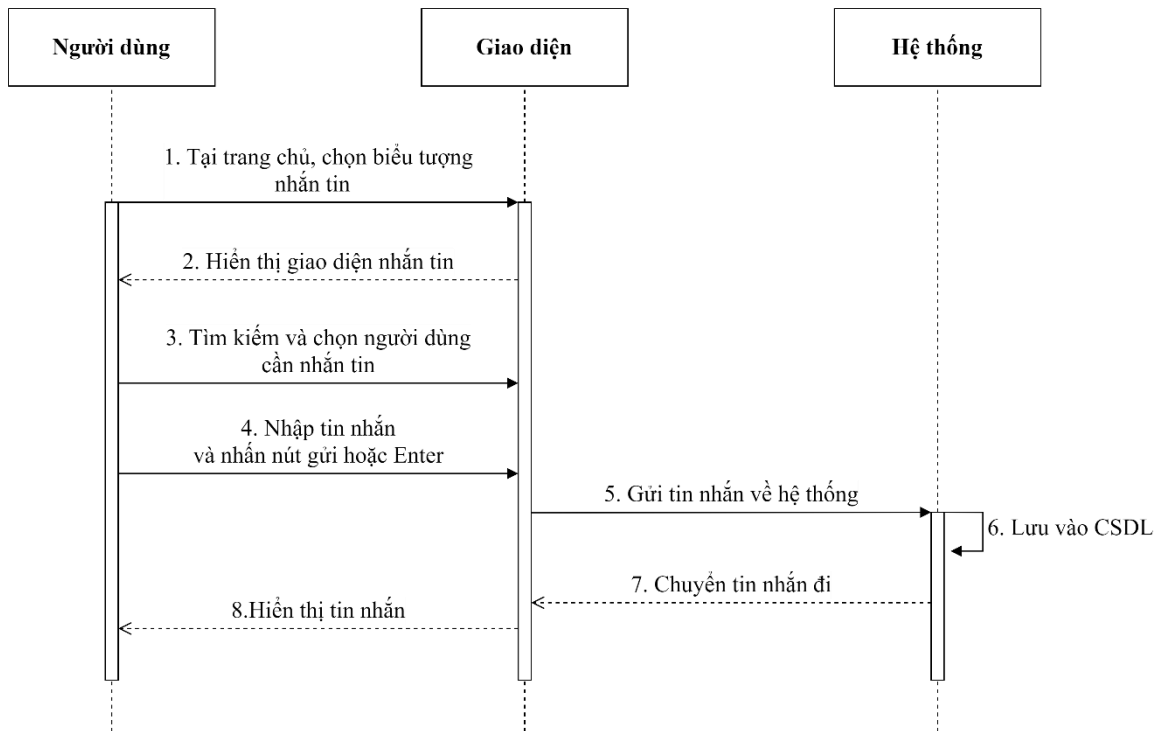
Hình 3.10 Biểu đồ tuần tự ca sử dụng theo dõi thành viên

3.4.7. Ca sử dụng chỉnh sửa thông tin cá nhân



Hình 3.11 Biểu đồ tuần tự ca sử dụng chỉnh sửa thông tin cá nhân

3.4.8. Ca sử dụng nhắn tin



Hình 3.12 Biểu đồ tuần tự ca sử dụng nhắn tin

CHƯƠNG 4. TRIỂN KHAI CHƯƠNG TRÌNH

4.1. Giao diện Website của người dùng

4.1.1. Giao diện trang chủ (thông báo trường)

4.1.2. Giao diện trang đăng nhập và đăng ký

4.1.3. Giao diện trang tin tức – sự kiện trường

4.1.4. Giao diện trang trao đổi

4.1.5. Giao diện đăng bài viết

4.1.6. Giao diện trang tài liệu

4.1.7. Giao diện đăng tài liệu

4.1.8. Giao diện trang theo dõi

4.1.9. Giao diện trang

4.1.10.

4.2. Giao diện Website của quản trị viên

4.2.1. Giao diện trang đăng nhập

4.2.2. Giao diện trang chủ (thống kê)

KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết quả đạt được

Trong quá trình tìm hiểu và hoàn thành đề tài “Xây dựng hệ thống trao đổi học tập online cho sinh viên UTC2”, nhóm đã đạt được những kết quả sau:

- ...
- Các dữ liệu được lưu trực tiếp xuống cơ sở dữ liệu, giúp cho người dùng không lo việc mất mát dữ liệu.
- Website có tính năng đăng nhập phân quyền, giúp đảm bảo hệ thống bảo đảm và an toàn hơn. Phòng ngừa được nguy cơ mất dữ liệu do lỗi từ người dùng.
- Hiểu được cách phân tích và thiết kế hệ thống nói chung và hệ thống trao đổi học tập trực tiếp cho sinh viên UTC2 nói riêng.
- Được ôn lại, rèn luyện và áp dụng các kiến thức đã học trong khi xây dựng hệ thống.
- Được biết thêm nhiều điều mới, thỏa sức sáng tạo đồng thời bản thân được nâng cao các kỹ năng tìm hiểu, xử lý vấn đề, kiên nhẫn, khả năng xây dựng và lập trình.

2. Kiến nghị

Do thời gian tìm hiểu và kiến thức còn hạn chế, hệ thống trao đổi học tập trực tuyến cho sinh viên UTC2 so với các hệ thống cùng loại hiện nay vẫn còn một số thiếu sót như:

- Hệ thống thật sự chưa có tính năng khác biệt hay sáng tạo.
- Các lỗi vẫn còn tiềm ẩn trong hệ thống chưa tìm ra được.
- Giao diện của hệ thống còn đơn giản.
- Tốc độ xử lý còn chậm.

3. Hướng phát triển

Trong tương lai nếu có điều kiện, hệ thống sẽ được phát triển theo hướng sau:

- Tìm ra và loại bỏ các lỗi vẫn còn tiềm ẩn để tăng độ chính xác, đúng đắn vốn có.

- Tìm hiểu thêm các công nghệ hỗ trợ khác để cải thiện giao diện, đồ họa của hệ thống.
- Áp dụng trí tuệ nhân tạo cùng với việc phát triển thêm nhiều tính năng cho hệ thống để nâng cao trải nghiệm cho người dùng.

PHỤ LỤC

- Đường dẫn website hệ thống trao đổi học tập trực tuyến cho sinh viên
UTC2:...

TÀI LIỆU THAM KHẢO

- [1]. **Cao Thị Nhận, Nguyễn Đình Loan Phương**, *Giáo Trình Phân Tích Thiết Kế Hệ Thống Thông Tin*, NXB Đại Học Quốc Gia, 2019.