

**TRƯỜNG ĐẠI HỌC THĂNG LONG  
KHOA TOÁN TIN**



# **KHÓA LUẬN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG DIỄN ĐÀN TOÁN HỌC**

**GIÁO VIÊN HƯỚNG DẪN**

**TS. MAI THÚY NGÀ**

**SINH VIÊN THỰC HIỆN**

**NHÂM NGỌC TUẤN ANH - A28348  
TM29**

**Ngành: Toán ứng dụng**

**HÀ NỘI – 2021**

# LỜI CẢM ƠN

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành nhất tới toàn thể các thầy cô giáo trong Khoa Toán Tin cũng như các thầy cô giảng dạy trong trường Đại học Thăng Long đã truyền đạt những kiến thức quý báu và bổ ích nhất cho em trong những năm học vừa qua.

Đặc biệt, em xin gửi lời cảm ơn chân thành và sâu sắc tới cô – Mai Thuý Nga, giảng viên khoa Toán Tin, Đại học Thăng Long. Thời gian qua nhờ vào sự hướng dẫn tận tình và sự động viên, giúp đỡ ân cần từ cô mà em mới có thành quả là xây dựng và hoàn thiện được Khoá luận tốt nghiệp như ngày hôm nay.

Em cũng xin gửi lời cảm ơn tới anh, chị và các bạn trong khoa Toán - Tin Đại học Thăng Long đã luôn giúp đỡ, chia sẻ những kinh nghiệm và chỉ dẫn cho em trong suốt quá trình học và quá trình thực hiện khóa luận.

Tuy nhiên, do năng lực còn hạn chế nên em không thể phát huy hết những ý tưởng, khả năng hỗ trợ của ngôn ngữ cùng với đó sự giúp đỡ nhiệt tình của mọi người vào đề tài. Trong quá trình xây dựng website, không thể tránh khỏi những sai sót, mong nhận được sự đóng góp và cảm thông của quý thầy cô và các bạn.

Mọi đóng góp xin gửi về địa chỉ email : [vucute9366@yahoo.com.vn](mailto:vucute9366@yahoo.com.vn).

Một lần nữa xin chân thành cảm ơn.

Sinh viên trình bày: Nhâm Ngọc Tuấn Anh

Lớp : Toán Ứng Dụng - K29

# LỜI MỞ ĐẦU

Năm 2020, chúng ta đã chứng kiến sự phát triển vũ bão của internet, mạng 4G vừa ra mắt không lâu thì mạng 5G thử nghiệm, cùng với đó là tác động lớn của Covid19. Các cách học của học sinh, sinh viên đã được bổ trợ thêm cách học online, tra cứu trên internet.

Sinh viên có thể tham gia các hội nhóm của trường (facebook, twitter), đăng bài tập và hỏi phương hướng làm. Nhưng sẽ có những hạn chế như sau:

- Các bài đăng không có liên kết với nhau, nên khi tìm lại sẽ rất khó
- Do tính chất của mạng xã hội nên dễ gây xao nhãng

Sinh viên cũng có thể gửi email cho thầy, cô hoặc nhắn tin qua zalo, facebook,... Nhưng vẫn sẽ tồn tại những hạn chế sau:

- Đòi hỏi sinh viên phải có địa chỉ email của thầy cô. Và nếu viết không chuẩn thì sẽ bị lọc vào hòm thư spam.
- Các hình thức mạng xã hội như zalo, facebook,... tuy rất phát triển nhưng không phải nơi các thầy cô thường xuyên truy cập. Nên sẽ có ít nhiều sự chậm trễ.

Nhược điểm chung là viết các câu trả lời với các công thức toán học thường rất khó khăn.

Còn diễn đàn thì sao? Nếu như nói email là mối quan hệ giữa 1 người hỏi và một người trả lời. Còn mạng xã hội là mối quan hệ giữa một người hỏi với nhiều người trả lời nhưng dễ gây xao nhãng. Diễn đàn là không gian thảo luận 1-n, giống như mạng xã hội. Nhưng nó là một nơi riêng - một ốc đảo mang tính chuyên biệt hơn, nơi mà sinh viên có thể:

- Nhập trực tiếp các công thức toán học, hoặc viết bài lên.
- Hệ thống có nhiều câu hỏi có thể tìm kiếm dễ dàng, nhanh chóng, chính xác trong cơ sở dữ liệu.
- Trả lời nhanh gọn. Truy cập mọi nơi.

Thời gian chọn đề tài “Xây dựng diễn đàn toán học” đến lúc hoàn thành trong vòng 6 tháng. Nội dung khóa luận được trình bày bao gồm 6 chương:

- Chương 1. Tổng quan hệ thống: Nghiệp vụ bài toán và chức năng hệ thống
- Chương 2. Công nghệ sử dụng: Công nghệ sử dụng trong forum
- Chương 3. Đặc tả hệ thống: Là khâu đầu tiên của quá trình phát triển phần mềm.
- Chương 4. Thiết kế Database, API: Thiết kế database và api
- Chương 5. Thiết kế chức năng: Phân tích sơ đồ lớp và sơ đồ trình tự.
- Chương 6. Kiểm thử: Giới thiệu về kiểm thử, các loại kiểm thử đã áp dụng.

# MỤC LỤC

<b>LỜI CẢM ƠN.....</b>	<b>2</b>
<b>LỜI MỞ ĐẦU.....</b>	<b>3</b>
<b>MỤC LỤC.....</b>	<b>4</b>
<b>DANH MỤC HÌNH ẢNH, SƠ ĐỒ.....</b>	<b>7</b>
<b>CHƯƠNG 1: TỔNG QUAN HỆ THỐNG.....</b>	<b>9</b>
1.1. Yêu cầu bài toán.....	9
1.1.1. Mô tả bài toán.....	9
1.1.2. Sơ đồ quy trình nghiệp vụ.....	10
1.1.3. Yêu cầu nghiệp vụ.....	10
1.2. Sơ đồ Use-Case.....	11
1.2.1. Sơ đồ chức năng hệ thống.....	11
1.2.2. Mô tả các tác nhân tham gia hệ thống.....	12
1.2.3. Các chức năng hệ thống.....	12
1.3. Ánh xạ các yêu cầu nghiệp vụ và các chức năng hệ thống.....	14
<b>CHƯƠNG 2 : CÔNG NGHỆ SỬ DỤNG.....</b>	<b>16</b>
2.1. Kiến trúc tổng thể.....	16
2.1.1. Sơ đồ kiến trúc.....	16
2.1.2. Mô tả kiến trúc.....	16
2.2. Server.....	16
2.2.1. <i>Java</i> .....	16
2.2.2. <i>Hibernate</i> .....	18
2.2.3. <i>Spring Data JPA</i> và <i>Hibernate</i> .....	19
2.2.4. <i>Spring boot</i> .....	21
2.2.5. <i>Spring Security</i> .....	22
2.2.6. <i>Postgresql</i> .....	23
2.3. Client.....	25
2.3.1. <i>CkEditor</i> .....	25
2.3.2. <i>Ant design</i> .....	26
2.3.3. <i>ReactJS</i> .....	27
2.4. Thư mục mã nguồn.....	28
2.4.1. Môi trường cài đặt.....	28
2.4.2. Cấu trúc thư mục mã nguồn.....	29
<b>CHƯƠNG 3 : ĐẶC TẢ HỆ THỐNG.....</b>	<b>31</b>
3.1. Chức năng đăng ký.....	31
3.2. Chức năng đăng nhập.....	33
3.3. Trang cá nhân.....	35
3.3.1. <i>Xem trang cá nhân</i> .....	35
3.3.2. <i>Quản lý thông tin trang cá nhân</i> .....	37
3.4. Quản lý câu hỏi.....	39
3.4.1. <i>Xem câu hỏi</i> .....	39
3.4.2. <i>Thêm câu hỏi</i> .....	42
3.4.3. <i>Sửa câu hỏi</i> .....	43
3.4.4. <i>Xóa câu hỏi</i> .....	45

3.5. Quản lý câu trả lời.....	46
3.5.1. Xem câu trả lời.....	46
3.5.2. Viết câu trả lời.....	48
3.5.3. Sửa câu trả lời.....	49
3.5.4. Xóa câu trả lời.....	50
3.6. Quản lý chủ đề.....	51
3.7. Quản lý người dùng.....	54
3.8. Chức năng đăng xuất.....	57
3.9. Chức năng quên mật khẩu.....	58
<b>CHƯƠNG 4. THIẾT KẾ DATABASE, API.....</b>	<b>60</b>
4.1. Mô tả cơ sở dữ liệu.....	60
4.2. Thiết kế database.....	60
4.2.1. Lược đồ cơ sở dữ liệu.....	60
4.2.2. Các bảng trong cơ sở dữ liệu.....	61
4.2.3. Bảng Users.....	62
4.2.4. Bảng Role.....	63
4.2.5. Bảng Tag.....	63
4.2.6. Thiết kế bảng question.....	63
4.2.7. Thiết kế bảng answer.....	64
4.2.8. Thiết kế bảng user_roles.....	64
4.2.9. Thiết kế bảng question_tag.....	65
4.3. Tổng quan API.....	65
4.3.1. Cấu trúc của một request.....	66
4.3.2. Cấu trúc của một response.....	67
4.4. Các API của hệ thống.....	68
4.4.1. API cho chức năng xác thực tài khoản.....	68
4.4.2. API cho quản lý câu hỏi.....	68
4.4.3. API cho quản lý câu trả lời.....	69
4.4.4. API cho quản lý chủ đề.....	70
4.4.4. API cho quản lý người dùng.....	71
<b>CHƯƠNG 5. THIẾT KẾ CHỨC NĂNG.....</b>	<b>72</b>
5.1. Quản lý câu hỏi.....	72
5.1.1. Sơ đồ lớp.....	72
5.1.2. Sơ đồ trình tự.....	72
5.2. Quản lý câu trả lời.....	75
5.2.1. Sơ đồ lớp.....	75
5.2.2. Sơ đồ trình tự.....	75
5.3. Quản lý chủ đề.....	78
5.3.1. Sơ đồ lớp.....	78
5.3.2. Sơ đồ trình tự.....	78
5.4. Quản lý người dùng.....	81
5.4.1. Sơ đồ lớp.....	81
5.4.2. Sơ đồ trình tự.....	81
5.5. Đăng ký.....	84
5.6. Đăng nhập.....	84

5.7. Reset Mật khẩu.....	85
<b>CHƯƠNG 6. KIỂM THỬ.....</b>	<b>86</b>
6.1. Giới thiệu chung.....	86
6.1.1. Kiểm thử phần mềm.....	86
6.1.2. Tạo sao phải kiểm thử.....	86
6.1.3. Ý nghĩa của kiểm thử.....	86
6.1.4. Mục tiêu kiểm thử.....	86
6.1.5. Nguyên tắc kiểm thử.....	87
6.1.6. Quy trình kiểm thử.....	87
6.2. Kiểm thử trong hệ thống.....	88
6.2.1. Các loại kiểm thử trong hệ thống.....	88
6.2.2. Kiểm thử trên Web Client.....	90
6.2.3. Kiểm thử trên Server.....	91
<b>KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN.....</b>	<b>94</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>95</b>

# DANH MỤC HÌNH ẢNH, SƠ ĐỒ

Hình 1.1. Sơ đồ quy trình nghiệp vụ.....	10
Hình 1.2. Sơ đồ use case của hệ thống.....	11
Hình 2.1. Sơ đồ kiến trúc hệ thống.....	16
Hình 2.1. Cách hoạt động của hibernate.....	18
Hình 2.3. Minh họa quá trình ánh xạ của Hibernate và database .....	19
Hình 2.2. Cơ chế hoạt động của Authentication.....	22
Hình 2.3. Cấu trúc thư mục mã nguồn.....	29
Hình 2.6. Cấu trúc file backend và frontend.....	30
Hình 3.1. Hình minh họa màn hình đăng ký diễn đàn.....	33
Hình 3.2. Hình minh họa màn hình đăng nhập diễn đàn.....	35
Hình 3.3. Hình minh họa màn hình thông tin cá nhân.....	36
Hình 3.4. Hình minh họa quản lý thông tin cá nhân.....	39
Hình 3.5. Hình minh họa chức năng xem câu hỏi.....	41
Hình 3.6. Hình minh họa chức năng thêm câu hỏi.....	43
Hình 3.7. Hình minh họa chức năng sửa câu hỏi.....	45
Hình 3.8. Hình minh họa chức năng xóa câu hỏi.....	46
Hình 3.9. Hình minh họa chức năng xem câu trả lời.....	47
Hình 3.10. Hình minh họa chức năng viết câu trả lời.....	49
Hình 3.11. Hình minh họa chức năng sửa câu trả lời.....	50
Hình 3.12. Hình minh họa chức năng xóa câu trả lời.....	51
Hình 3.13. Hình minh họa chức năng quản lý chủ đề.....	53
Hình 3.14. Hình minh họa chức năng quản lý người dùng.....	56
Hình 3.15. Hình minh họa chức năng đăng xuất.....	57
Hình 3.16. Hình minh họa chức năng quên mật khẩu.....	59
Hình 4.1. Lược đồ cơ sở dữ liệu.....	61
Hình 4.2. Hình mô tả cấu trúc request.....	66
Hình 5.1. Sơ đồ lớp câu hỏi.....	72
Hình 5.2. Tạo câu hỏi.....	72
Hình 5.3. Hình mô tả sửa câu hỏi.....	73
Hình 5.4. Hình mô tả chức năng xóa câu hỏi.....	73
Hình 5.5. Mô tả chức năng tìm kiếm câu hỏi.....	74
Hình 5.6. Xem danh sách câu hỏi.....	74
Hình 5.7. Mô tả sơ đồ lớp.....	75
Hình 5.8. Mô tả chức năng tạo câu trả lời.....	75
Hình 5.9. Sửa câu trả lời.....	76
Hình 5.10. Mô tả chức năng xóa câu trả lời.....	76
Hình 5.11. Mô tả chức năng tìm kiếm câu trả lời.....	77
Hình 5.12. Mô tả chức năng xóa câu trả lời.....	77
Hình 5.13. Mô tả chức năng quản lý chủ đề.....	78
Hình 5.14. Mô tả chức năng tạo chủ đề.....	78
Hình 5.15. Mô tả chức năng sửa chủ đề.....	79
Hình 5.16. Mô tả chức năng xóa chủ đề.....	79
Hình 5.17. Mô tả chức năng tìm kiếm chủ đề.....	80

Hình 5.18. Mô tả chức năng xem danh sách chủ đề.....	80
Hình 5.20. Mô tả chức năng tạo chủ đề.....	81
Hình 5.21. Mô tả chức năng sửa thông tin người dùng.....	82
Hình 5.22. Mô tả chức năng xóa người dùng.....	82
Hình 5.23. Mô tả chức năng tìm kiếm người dùng.....	83
Hình 5.24. Mô tả chức năng danh sách người dùng.....	83
Hình 5.25. Mô tả chức năng đăng ký.....	84
Hình 5.26. Mô tả chức năng đăng nhập.....	84
Hình 5.27. Mô tả chức năng Reset mật khẩu.....	85
Hình 6.1. Mô tả kiểm thử.....	86
Hình 6.2. Sơ đồ quy trình kiểm thử.....	87



# CHƯƠNG 1: TỔNG QUAN HỆ THỐNG

## 1.1. Yêu cầu bài toán

### 1.1.1. Mô tả bài toán

Khi tham gia vào diễn đàn người sử dụng sẽ lựa chọn một chủ đề trong loạt các chủ đề của diễn đàn, Ví dụ ta chọn chủ đề “lập trình”, sau khi chọn chủ đề xong, một danh sách các câu hỏi thuộc chủ đề đó sẽ xuất hiện. Người sử dụng sẽ chọn một trong các câu hỏi trong danh sách để xem nội dung, ví dụ chọn bài “Nhiệt lượng mặt trời” thì nội dung của bài này hiện ra, nếu người đọc thấy có thể trả lời được thì nhấn vào nút trả lời và nhập nội dung vào form trả lời và nhấn nút trả lời, bài trả lời sẽ nằm ngay dưới câu hỏi. Nếu người sử dụng muốn gửi một bài mới thì có thể nhấn vào nút “đặt câu hỏi”. Trên đây là phần sơ lược về hoạt động cơ bản của diễn đàn. Ngoài ra nó còn cho phép người dùng có nhiều thao tác khác nhau. Để minh họa cho chức năng chính của diễn đàn là gửi và trả lời một câu hỏi, ta đưa ra ví dụ để chúng ta có thể hình dung được khái quát diễn đàn dùng làm gì và nó có đáng đáp như thế nào.

**Đặt câu hỏi** là chức năng nổi bật nhất của forum. Câu hỏi là mục cho phép người dùng soạn thảo câu hỏi. Câu hỏi gồm 3 mục:

- Tiêu đề
- Chủ đề
- Nội Dung

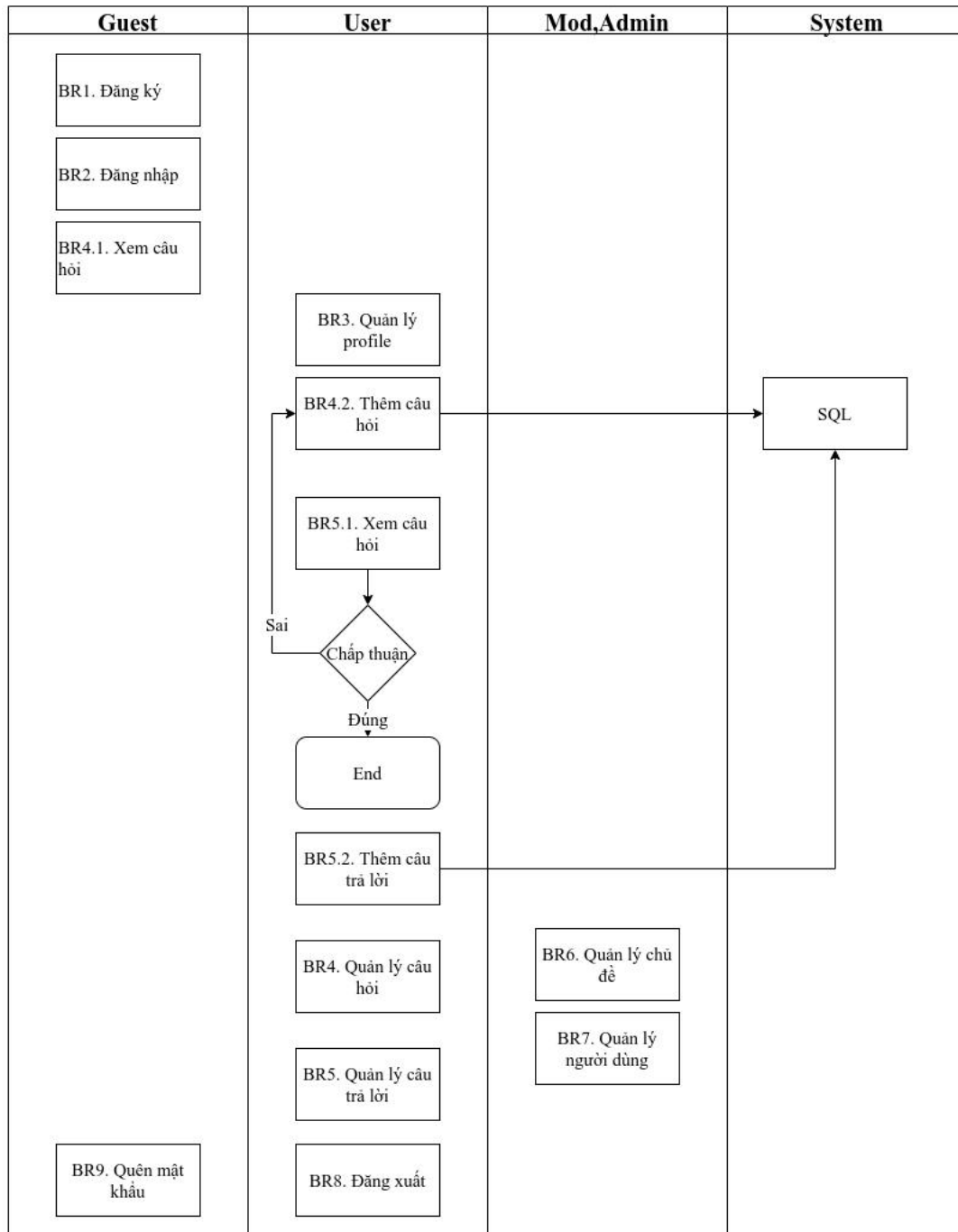
Câu hỏi đó có thể được sửa lại, xóa hoặc đóng phần trả lời tùy thuộc vào quyết định của người tạo ra nó.

Lưu ý: Người dùng muốn đặt được câu hỏi trong forum thì phải đăng nhập vào hệ thống mới có quyền đặt câu hỏi.

Đối với một câu hỏi được tạo ra, tất cả những người có quyền nhìn thấy nó trên trang chủ của mình đều có thể ẩn viết câu trả lời. Đối với những câu hỏi công khai thì mọi người đều có thể ẩn tham gia viết câu trả lời, trả lời đó có thể xóa hoặc sửa tùy theo mục đích của người tạo ra nó. Còn đối với những câu hỏi ở chế độ khóa trả lời thì chỉ có thể xem mà không nhập câu trả lời được.

Để quản lý chủ đề có trong hệ thống hệ thống, trong forum có ít nhất một moderator đóng người quản trị vai trò làm. Moderator này có quyền quản lý thông tin chủ đề có trong hệ thống.

### 1.1.2. Sơ đồ quy trình nghiệp vụ



Hình 1.1. Sơ đồ quy trình nghiệp vụ

### 1.1.3. Yêu cầu nghiệp vụ

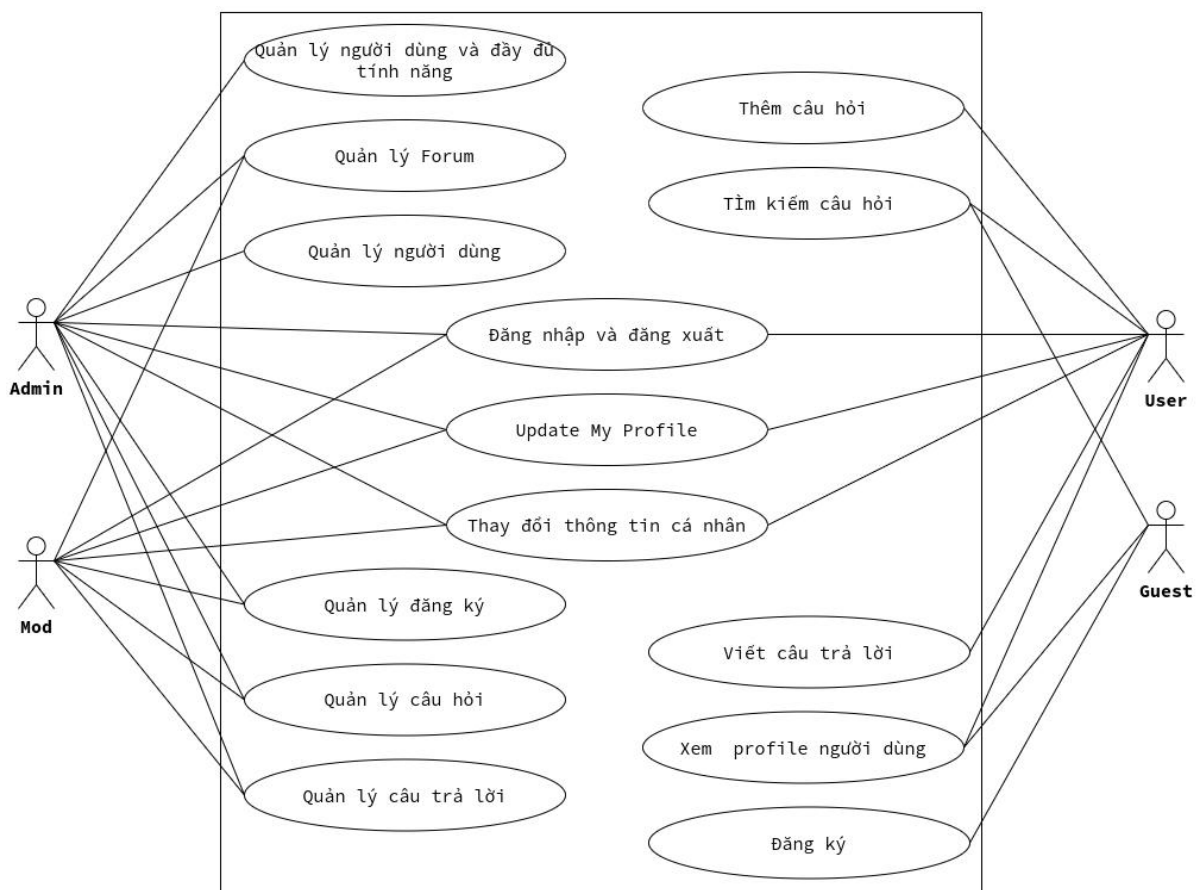
- BR1: Đăng ký
- BR2: Đăng nhập
- BR3: Trang cá nhân
  - BR3.1: Xem trang cá nhân
  - BR3.2: Quản lý thông tin cá nhân.
- BR4: Quản lý câu hỏi

- BR4.1: Xem câu hỏi
- BR4.2: Thêm câu hỏi
- BR4.3: Sửa câu hỏi
- BR4.4: Xóa câu hỏi
- BR5: Quản lý câu trả lời
  - BR5.1: Xem câu trả lời
  - BR5.2: Viết câu trả lời
  - BR5.3: Sửa câu trả lời
  - BR5.4: Xóa câu trả lời
- BR6: Thêm chủ đề
- BR7: Quản lý người dùng
- BR8: Đăng xuất
- BR9: Quên mật khẩu

## 1.2. Sơ đồ Use-Case

### 1.2.1. Sơ đồ chức năng hệ thống

Từ các yêu cầu nghiệp vụ. Chúng ta có sơ đồ use-case sau:



Hình 1.2. Sơ đồ use case của hệ thống

### ***1.2.2. Mô tả các tác nhân tham gia hệ thống***

- **Guest:** Xem mọi thông tin forum đăng tải, gồm:
  - Xem trang chủ
  - Xem các câu hỏi theo:
    - ◆ Theo người dùng
    - ◆ Theo chủ đề
  - Xem chi tiết câu hỏi được chọn.,
  - Xem câu trả lời
  - Xem danh sách người dùng.
  - Đăng ký làm thành viên.
- **User:** là sinh viên hoặc bất kỳ người nào đã đăng nhập - được cấp role "user", và họ sử dụng các chức năng:
  - Đăng nhập
  - Xem câu trả lời
  - Đặt câu trả lời
  - Quản lý câu hỏi của chính mình
  - Quản lý câu trả lời của chính mình.
- **Mod:** Viết tắt của "Moderator" .Những người có phụ trách giải đáp câu hỏi của sinh viên.
  - Quản lý toàn bộ câu hỏi có trên hệ thống
  - Quản lý toàn bộ câu trả lời có trên hệ thống
  - Quản lý các chủ đề của diễn đàn:
    - ◆ Xóa một chủ đề bất kỳ
    - ◆ Thêm vào một chủ đề mới
- **Admin:** Viết tắt của "Administrator". Chức năng của Admin là điều hành chung trong toàn bộ Forum , là thành viên cao nhất trong Forum có chức năng vừa quản lý vừa phát triển Forum .Người phân quyền cho các user. Trực tiếp quản lý dữ liệu trong database.
  - Quản lý người sử dụng:
    - ◆ Xóa người sử dụng: xóa một người có role là “user” ra khỏi hệ thống
    - ◆ Tạo người sử dụng mới

### ***1.2.3. Các chức năng hệ thống***

- **UC #01. Đăng ký:** Chức năng cho phép người dùng tạo một tài khoản để người dùng đăng nhập vào hệ thống. Người dùng truy cập vào website và nhập đầy đủ thông tin yêu cầu để tạo tài khoản cho mình.

- **UC #02. Đăng nhập:** Chức năng cho phép người dùng đăng nhập thông qua tên đăng nhập và mật khẩu đã được người sử dụng đăng ký tài khoản từ trước. Sau khi đăng nhập vào hệ thống, người dùng có thể sử dụng các chức năng của hệ thống.
- **UC #03. Trang cá nhân:**
  - **UC #3.1. Xem trang cá nhân:** Chức năng này cho phép người dùng có thể xem được trang cá nhân của một người dùng bất kỳ trong hệ thống. Trang cá nhân này gồm các thông tin như: thông tin người dùng, những câu hỏi đã đặt.
  - **UC #3.2. Quản lý thông tin trang cá nhân:** Chức năng này dùng để quản lý các thông tin của một người dùng như: tên người dùng, những câu hỏi đã đặt, những câu trả lời đã viết .
- **UC #04. Quản lý câu hỏi:**
  - **UC #4.1. Xem câu hỏi:** Chức năng cho phép người dùng có thể xem danh sách câu hỏi hoặc chi tiết 1 câu hỏi có trên hệ thống.
  - **UC #4.2. Thêm câu hỏi:** Chức năng cho phép người dùng tạo mới một câu hỏi.
  - **UC #4.3. Sửa câu hỏi:** Chức năng cho phép người dùng sửa các thành phần của câu hỏi.
  - **UC #4.4. Xóa câu hỏi:** Chức năng cho phép người dùng xóa câu trả lời.
- **UC #05. Quản lý câu trả lời:**
  - **UC #5.1. Xem câu trả lời:** Chức năng cho phép người dùng xem các câu trả lời của một câu hỏi cụ thể.
  - **UC #5.2. Viết câu trả lời:** Chức năng cho phép người dùng tạo mới một câu trả lời.
  - **UC #5.3. Sửa câu trả lời:** Chức năng cho phép người dùng sửa nội dung của một câu trả lời.
  - **UC #5.4. Xóa câu trả lời:** Chức năng cho phép người dùng xóa câu trả lời.
- **UC #06. Quản lý chủ đề (tag):** Chức năng dành cho admin và mod để quản lý chủ đề.
- **UC #07. Quản lý người dùng:** Chức năng dành cho admin và mod. Có quyền thêm, xóa và sửa thông tin người dùng.
- **UC #08. Đăng xuất:** Chức năng này cho phép người dùng muốn thoát hẳn khỏi trạng thái đăng nhập hệ thống, người dùng sau khi đã đăng xuất sẽ không thể thực hiện chức năng nào nữa, mà bắt buộc phải đăng nhập lại khi muốn sử dụng hệ thống.
- **UC #09. Quên mật khẩu:** Chức năng này cho phép người dùng cài lại mật khẩu nếu quên bằng cách nhập email của người dùng đã đăng ký và mật khẩu mới.

### 1.3. Ảnh xạ các yêu cầu nghiệp vụ và các chức năng hệ thống

UC	Mô tả	BR
UC #01: Đăng ký		
UC #01	Đăng ký	BR1
UC #02: Đăng nhập		
UC #02	Đăng nhập	BR2
UC #03: Trang cá nhân		
UC #3.1	Xem trang cá nhân	BR3.1
UC #3.2	Quản lý trang cá nhân	BR3.2
UC #04: Quản lý câu hỏi		
UC #4.1	Xem câu hỏi	BR4.1
UC #4.2	Thêm câu hỏi	BR4.2
UC #4.3	Sửa câu hỏi	BR4.3
UC #4.4	Xóa câu hỏi	BR4.4
UC #05: Câu trả lời		
UC #5.1	Xem câu trả lời	BR5.1
UC #5.2	Viết câu trả lời	BR5.2
UC #5.3	Sửa câu trả lời	BR5.3
UC #5.4	Xóa câu trả lời	BR5.4
UC #06: Quản lý chủ đề		
UC #06	Quản lý chủ đề	BR6
UC #07: Quản lý người dùng		

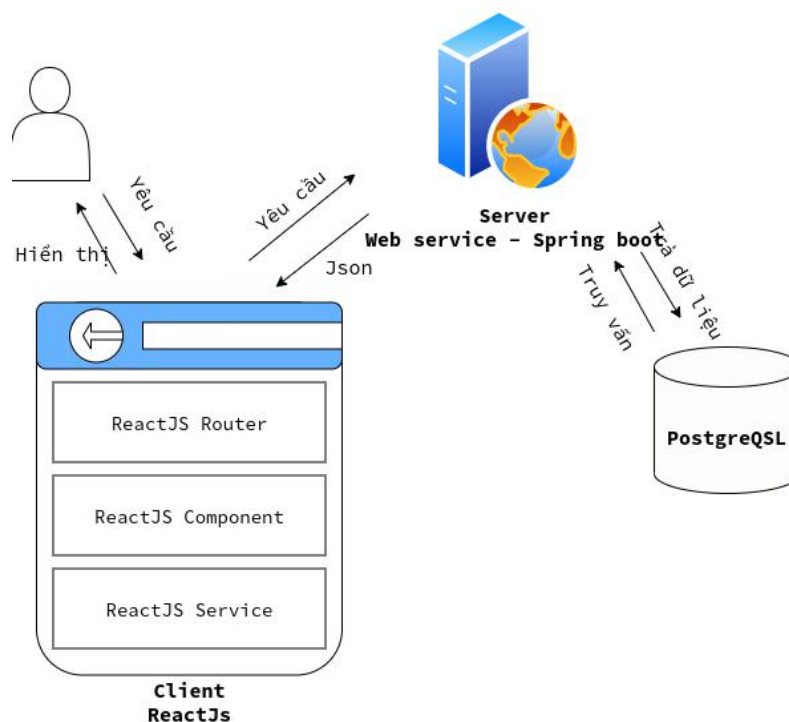
UC #07	Quản lý người dùng	BR7
UC #08: Đăng xuất		
UC #08	Đăng xuất	BR8
UC #09: Quên mật khẩu		
UC #09	Quên mật khẩu	BR9

**Kết luận chương:** Như vậy qua chương 1, chúng ta có thể thấy được một cái nhìn tổng quan hơn về khóa luận “diễn đàn toán học”. Em đã quyết định chọn đề tài này cho khóa luận tốt nghiệp, xây dựng các yêu cầu nghiệp vụ dựa trên những yêu cầu bài toán đề ra. Để từ đó, em có xây dựng và phát triển sơ đồ kiến trúc tổng thể của hệ thống gồm có 2 phần chính là: Server và Client (Web). Tiếp tục tài liệu chương 2, em sẽ đi sâu hơn vào các công nghệ được sử dụng trong từng phần kiến trúc tổng thể.

# CHƯƠNG 2 : CÔNG NGHỆ SỬ DỤNG

## 2.1. Kiến trúc tổng thể

### 2.1.1. Sơ đồ kiến trúc



Hình 2.1. Sơ đồ kiến trúc hệ thống

### 2.1.2. Mô tả kiến trúc

Hệ thống được thiết kế với 2 phần, gồm:

- Client: Được viết bằng ReactJS
  - Người dùng tương tác với ứng dụng
  - Ứng dụng trao đổi dữ liệu với server qua phương thức HTTP
- Server: Sử dụng Java Spring Boot
  - Server trao đổi dữ liệu với ReactJs bằng phương thức HTTP, server sẽ trả các dữ liệu cho client dưới dạng Json
  - Sever liên kết với database ở đây là PostgreSQL

## 2.2. Server

### 2.2.1. Java

Java là một ngôn ngữ lập trình hiện đại, bậc cao, hướng đối tượng, bảo mật và mạnh mẽ. và là một Platform.

Platform: Bất cứ môi trường phần cứng hoặc phần mềm nào mà trong đó có một chương trình chạy, thì được hiểu như là một Platform. Với môi trường runtime riêng cho mình (JRE) và API, Java được gọi là Platform.



Ngôn ngữ lập trình Java ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995. Phiên bản mới nhất của Java Standard Edition là Java SE 8. Với sự tiến bộ của Java và sự phổ biến rộng rãi của nó, nhiều cấu hình đã được xây dựng để phù hợp với nhiều loại nền tảng khác nhau. Ví dụ: J2EE cho các ứng dụng doanh nghiệp, J2ME cho các ứng dụng di động.

Ưu điểm của Java:

Cũng như nhiều ngôn ngữ lập trình khác, trình duyệt Java cũng có cho mình khá nhiều ưu điểm. Trong đó cần được kể tới như: hướng đối tượng rộng, có một nền tảng riêng biệt, có thiết kế đơn giản, khả năng bảo mật cao, nhanh và mạnh.

a, Hướng đối tượng rộng

Hướng đối tượng rộng trong Java chính là tất cả những thứ đều được mở rộng, trong đó thì Java sẽ được dùng dựa trên các mô hình là Object.

b, Java có nền tảng riêng biệt

Java có nền tảng riêng biệt, người ta nói như vậy là bởi khi nhận được một câu lệnh nào đó, thì Java sẽ tự động thực hiện biên tập câu lệnh đó sang những Bite Code ở dạng độc lập. Trong đó, Bite Code độc lập này sẽ được hỗ trợ bởi các dịch bằng Virtual Machine với bất cứ phần mềm, ứng dụng nào có sử dụng tới nó.

c, Thiết kế mẫu khá đơn giản

Không giống như nhiều ngôn ngữ lập trình khác, Java có thiết kế mẫu khá là đơn giản bởi thế mà những nhà lập trình viên không cần phải mất quá nhiều thời gian theo học. Muốn học tốt, thành thạo về Java thì mỗi người chỉ mất từ 1 đến 3 năm là đã có thể thành công.

d, Tính bảo mật cao

Tính bảo mật cao, chính là một ưu điểm của Java so với các trình duyệt khác. Trong đó, khả năng của Java là phát hiện được những thành phần có chứa các virus độc hại, rồi sau đó nó cũng có thể “tiêu diệt” được virus đó. Để thực hiện được điều này, những nhà lập trình viên ra Java đã phát triển cho nó những thuật toán ở mức độ cao nhất.

Ứng dụng trên website được nhiều quốc gia yêu thích sử dụng cho các lĩnh vực khác nhau như: tài chính ngân hàng, bảo hiểm, kiểm toán... Tính bảo mật ở trên phần mềm Java rất cao thế cho nên khi sử dụng Java trên những phần mềm này sẽ được viết riêng cho mỗi tổ chức phi chính phủ.

Vì Java có tính bảo mật cao, nên đó là cơ sở để người dùng tin tưởng và sử dụng rộng rãi hơn.

e, Nhanh và mạnh

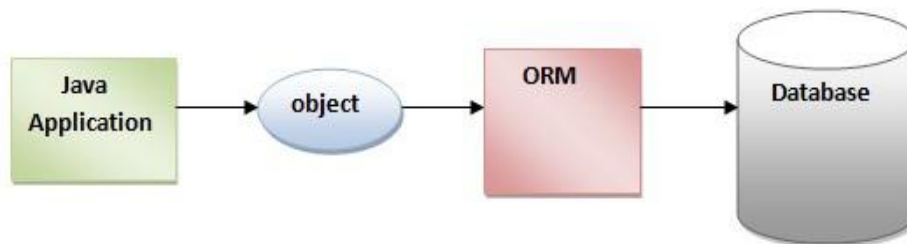
Đối với ưu điểm này, Java là một trình duyệt có được khả năng xử lý những tình huống bị xảy ra trên máy chủ rất nhanh. Bên cạnh đó, nó cũng có được khả năng truyền dẫn về internet với tốc độ cao, không kém gì những ứng dụng khác.

### 2.2.2. *Hibernate*

Hibernate ra đời năm 2001 bởi nhà sáng lập Gavin King như một sự thay thế cho EJB2 kiểu thực thể bean.

Hibernate framework là một giải pháp ORM (Object Relational Mapping) mã nguồn mở, gọn nhẹ. Hibernate giúp đơn giản hoá sự phát triển của ứng dụng Java để tương tác với cơ sở dữ liệu.

Tool ORM giúp đơn giản hoá việc tạo ra dữ liệu, thao tác dữ liệu và truy cập dữ liệu. Đó là một kỹ thuật lập trình để ánh xạ đối tượng vào dữ liệu được lưu trữ trong cơ sở dữ liệu.



Hình 2.1. Cách hoạt động của hibernate

Lợi ích của Hibernate Framework:

- Mã nguồn mở và nhẹ: Hibernate Framework là mã nguồn mở có giấy phép LGPL và nhẹ.
- Hiệu suất nhanh: Hiệu suất của Hibernate Framework là nhanh bởi vì bộ nhớ cache được sử dụng trong nội bộ Hibernate Framework. Có hai loại bộ nhớ cache trong Hibernate Framework, gồm bộ nhớ cache cấp một và bộ nhớ cache cấp hai. Bộ nhớ cache cấp một được bật bằng lệnh mặc định.
- Truy vấn cơ sở dữ liệu độc lập: HQL (Hibernate Query Language) là phiên bản hướng đối tượng của SQL. Nó tạo ra các truy vấn cơ sở dữ liệu độc lập. Vì vậy, bạn không cần phải viết các truy vấn cơ sở dữ liệu cụ thể. Trước Hibernate, nếu dự án có cơ sở dữ liệu bị thay đổi, chúng ta cần phải thay đổi truy vấn SQL dẫn đến sự cố bảo trì.
- Tạo bảng tự động: Hibernate framework cung cấp phương tiện để tạo ra các bảng cơ sở dữ liệu tự động. Vì vậy, không cần phải tạo ra các bảng trong cơ sở dữ liệu bằng tay.
- Đơn giản lệnh join phức tạp: Có thể lấy dữ liệu từ nhiều bảng một cách dễ dàng với Hibernate framework.

- Cung cấp thông kê truy vấn và trạng thái cơ sở dữ liệu: Hibernate hỗ trợ bộ nhớ cache truy vấn và cung cấp số liệu thống kê về truy vấn và trạng thái cơ sở dữ liệu.

### 2.2.3. Spring Data JPA và Hibernate

Điều đầu tiên, Spring Data JPA là một module nhỏ trong một project lớn gọi là Spring Data project. Mục đích của Spring Data project là giảm thiểu các đoạn code lặp đi lặp lại liên quan đến phần thao tác với các hệ thống quản trị data khi phát triển các ứng dụng có sử dụng Spring framework.

Để đạt được mục đích giảm thiểu code, Spring Data định nghĩa một interface chính tên là Repository nằm trong module Spring Data Common, module này sẽ được sử dụng cho tất cả các module còn lại trong Spring Data project. Nội dung của interface này đơn giản như sau:

```
package org.springframework.data.repository;

import org.springframework.stereotype.Indexed;

@Indexed
public interface Repository<T, ID> {

}
```

Interface này sử dụng 2 generic type:

- T là domain class mà repository sẽ quản lý
- ID là kiểu dữ liệu của ID của domain class mà repository quản lý.

Kết quả sẽ là:

```

<terminated> Application (6) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_131.jdk/Contents/Home/bin/java (Jan 27, 2018, 12:04:16 PM)
Jan 27, 2018 12:04:16 PM org.hibernate.cfg.Environment <clinit>
INFO: HHH000206: hibernate.properties not found
Jan 27, 2018 12:04:16 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
Sat Jan 27 12:04:16 ICT 2018 WARN: Establishing SSL connection without server's identity verification is not recommended
Jan 27, 2018 12:04:16 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
Sat Jan 27 12:04:17 ICT 2018 WARN: Establishing SSL connection without server's identity verification is not recommended
Hibernate: select student0_.id as id1_0_0_, student0_.name as name2_0_0_ from student student0_ where student0_.id=?
Khanh

```

Hình 2.3. Minh họa quá trình ánh xạ của Hibernate và database .

### Kết hợp giữa Java JPA và Hibernate

Ví dụ 1:

Class Task dưới được ánh xạ và bảng Task. Tuy nhiên, có sự bất phù hợp trong tên cột. Chúng ta có thể dùng một số annotation để thực hiện ánh xạ:

- `@Table(name = "Task")` - đánh dấu class này sẽ mapping với bảng trong CSDL, bảng này có tên là Task
- `@Id` - đánh dấu trường sẽ là khóa chính
- `@GeneratedValue` - giá trị sẽ tự sinh ra
- `@Column(name = "description")` - đánh dấu trường mapping với cột trong bảng, cột có tên description

Object:

```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue; import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.Table;
@Entity @Table(name = 'Task' )
public class Task {
    @Id
    @GeneratedValue
    private int id;

    @Column(name = 'description' )
    private String desc;

    @Column(name = 'target_date' )
    private Date targetDate;

    @Column(name = 'is_done' )
    private boolean isDone;
}
```

Table: CREATE TABLE task ( id INTEGER GENERATED BY DEFAULT AS IDENTITY, description VARCHAR(255), is\_done BOOLEAN, target\_date TIMESTAMP, PRIMARY KEY (id) )

Ví dụ 2:

Mối quan hệ giữa các object được thể hiện khác so với mối quan hệ giữa các bảng. Mỗi Employee có thể có nhiều Task. Mỗi Task có thể nhiều Employee cùng làm. Đây là mối quan hệ nhiều – nhiều. Chúng ta sử dụng annotation `@ManyToMany` để thiết lập mối quan hệ này.

Object:

```
public class Employee { //Some other code
```

```

@ManyToMany
private List<Task> tasks;

}

public class Task { //Some other code

    @ManyToMany(mappedBy = 'tasks')
    private List<Employee> employees;
}

Table: CREATE TABLE employee ( id BIGINT NOT NULL, OTHER_COLUMNS ) CREATE TABLE
employee_tasks ( employees_id BIGINT NOT NULL, tasks_id INTEGER NOT NULL ) CREATE TABLE
task ( id INTEGER GENERATED BY DEFAULT AS IDENTITY, OTHER_COLUMNS )

```

#### 2.2.4. *Spring boot*

Spring Boot là một dự án phát triển bởi JAVA (ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng. Để phát triển một ứng dụng web cơ bản Hello World sử dụng Spring framework bạn sẽ cần ít nhất 5 công đoạn sau:

- 1) Tạo một project sử dụng Maven với các dependency cần thiết của Spring MVC và Servlet API.
- 2) Một tập tin web.xml để khai báo DispatcherServlet của Spring MVC.
- 3) Một tập tin cấu hình của Spring MVC.
- 4) Một class Controller trả về một trang “Hello World” khi có request đến.
- 5) Cuối cùng là phải có một web server dùng để triển khai ứng dụng lên chạy

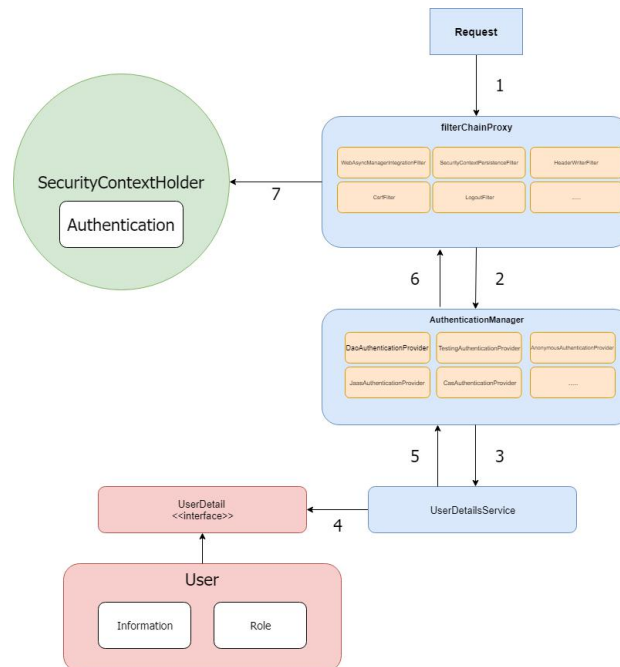
*Sử dụng Spring Boot sẽ mang đến những lợi ích nổi bật như sau:*

- Giúp tạo được ứng dụng độc lập dựa trên Spring, có thể tự chạy được `java -jar`
- Có ít cấu hình, có khả năng tự động cấu hình lại Spring khi cần, từ đó giúp các thành viên có thể tiết kiệm thời gian viết code và tăng thêm năng suất.
- Giữ đầy đủ các tính năng của Spring Framework
- Spring boot không yêu cầu cấu hình XML và không sinh code cấu hình
- Không cần phải triển khai file WAR mà thực hiện nhúng trực tiếp các ứng dụng server
- Giúp cung cấp nhiều plugin

### 2.2.5. Spring Security

Spring security là một framework của spring được tạo ra nhằm phục vụ việc Authentication(Xác thực) và Authorization (Phân quyền) cho các ứng dụng java. Chủ yếu sẽ là các ứng dụng web và rest service.

Cơ chế hoạt động của Authentication (Xác thực) trong Spring Security.

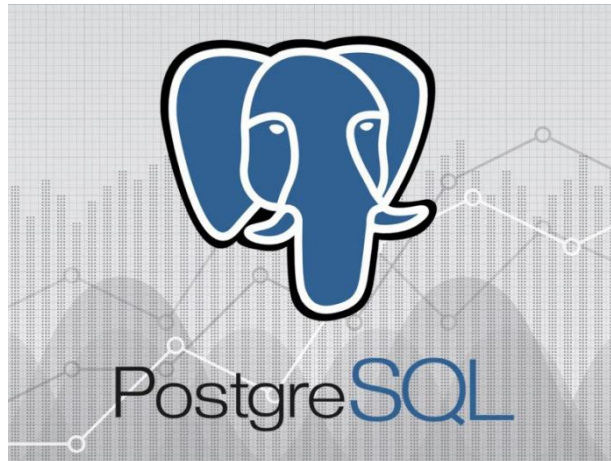


Hình 2.2. Cơ chế hoạt động của Authentication

- Mỗi request vào hệ thống sẽ được trải qua một tập các Filter, tập các filter này được quản lý bởi một **springSecurityFilterChain**, cụ thể ở đây là **DelegatingFilterProxy**. Bản chất filter Proxy này giữ một tập các filter đã được định nghĩa trước và lần lượt cho request đi qua filter một
- Với mỗi Filter, sẽ có một **authenticationProvider** phù hợp tương ứng. Ví dụ khi filter lọc được username và password từ request đầu vào, nó sẽ tạo ra một object **Authentication** (**UsernamePasswordAuthenticationToken**) và bắn sang cho AuthenticationManager xử lý. Authentication bản chất cũng giống thằng filter, nó quản lý một tập các AuthenticationProvider, mỗi AuthenticationProvider sẽ chịu trách nhiệm cho một phương cách xác thực của hệ thống. Trong list **AuthentionProvider** sẽ có một thằng chịu trách nhiệm xử lý object **Authentication** kia. Cụ thể m sẽ có demo trong các bài post sau.
- **AuthentionProvider** sẽ kết hợp với **userDetailsService** của hệ thống để lôi ra các thông tin liên quan đến **User** bao gồm các thông tin cơ bản và thông tin mang tính phân quyền của user. Các thông tin này được gói gọn lại trong một cài đặt cụ thể của UserDetail.

- Với dữ liệu có được từ **userDetailsService**, **AuthenticationProvider** tiến hành xác thực với dữ liệu đầu vào từ request. Nếu việc Xác thực thành công, Một Object Authentication Sẽ được lưu vào trong **SecurityContextHolder** của spring. Việc xác thực kết thúc.

#### 2.2.6. *Postgresql*



PostgreSQL đã tự đưa ra tuyên bố là "Cơ sở dữ liệu mã nguồn mở tiên tiến nhất thế giới".

#### **PostgreSQL là gì?**

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay.

PostgreSQL được phát triển dựa trên POSTGRES 4.2 tại phòng khoa học máy tính Berkeley, Đại học California.

PostgreSQL được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, PostgreSQL sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.

PostgreSQL là một phần mềm mã nguồn mở miễn phí. Mã nguồn của phần mềm khả dụng theo license của PostgreSQL, một license nguồn mở tự do. Theo đó, bạn sẽ được tự do sử dụng, sửa đổi và phân phối PostgreSQL dưới mọi hình thức.

PostgreSQL không yêu cầu quá nhiều công tác bảo trì bởi có tính ổn định cao. Do đó, nếu bạn phát triển các ứng dụng dựa trên PostgreSQL, chi phí sở hữu sẽ thấp hơn so với các hệ thống quản trị dữ liệu khác.

Công việc của PostgreSQL:

- Thêm dữ liệu vào cơ sở dữ liệu.
- Sửa dữ liệu trong cơ sở dữ liệu.
- Xóa dữ liệu khỏi cơ sở dữ liệu.



- Truy vấn dữ liệu từ cơ sở dữ liệu.
- Tạo cơ sở dữ liệu, tạo bảng...

Vì sao sử dụng PostgreSQL?

PostgreSQL sở hữu một hệ tính năng đa dạng giúp hỗ trợ các nhà phát triển xây dựng app, các nhà quản trị bảo vệ toàn vẹn dữ liệu, và tạo ra một môi trường chịu lỗi **fault-tolerant** giúp bạn quản lý dữ liệu bất kể tập dữ liệu lớn hay nhỏ. Bên cạnh hệ thống nguồn mở và miễn phí, PostgreSQL cũng có khả năng mở rộng tuyệt vời. Ví dụ, bạn có thể định nghĩa các kiểu dữ liệu riêng của bạn, xây dựng các hàm tùy chỉnh, hay viết mã từ các ngôn ngữ lập trình khác nhau mà không cần biên dịch lại cơ sở dữ liệu! PostgreSQL tuân theo tiêu chuẩn SQL nhưng không mâu thuẫn với các tính năng truyền thống hay có thể dẫn đến các quyết định kiến trúc gây hại. Nhiều tính năng theo tiêu chuẩn SQL được hỗ trợ, tuy nhiên đôi khi có thể có cú pháp hoặc hàm hơi khác một chút.

Các yếu tố giúp PostgreSQL nổi bật:

- Tốc độ: PostgreSQL rất nhanh. Những nhà phát triển cho rằng PostgreSQL là cơ sở dữ liệu nhanh nhất mà bạn có thể có.
- Dễ sử dụng: PostgreSQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn .
- Giá thành: PostgreSQL là miễn phí cho hầu hết các việc sử dụng trong một tổ chức.
- Năng lực: Nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Bạn có thể truy cập PostgreSQL tương tác với sử dụng một vài giao diện để bạn có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web...
- Kết nối và bảo mật: PostgreSQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó bạn có thể chia sẻ dữ liệu của bạn với bất kỳ ai, bất kỳ nơi nào. Nhưng PostgreSQL kiểm soát quyền truy cập cho nên người mà không nên nhìn thấy dữ liệu của bạn thì không thể nhìn được.
- Sự hỗ trợ: Luôn có một cộng đồng nhiệt tình sẵn sàng để trợ giúp về các vấn đề có thể gặp phải khi làm việc với PostgreSQL.
- Tính linh động: Là hệ thống quản lý cơ sở dữ liệu quan hệ đối tượng, PostgreSQL cho phép thêm vào các tính năng tùy chỉnh được phát triển bằng các ngôn ngữ chương trình khác nhau như C/C , Java,...



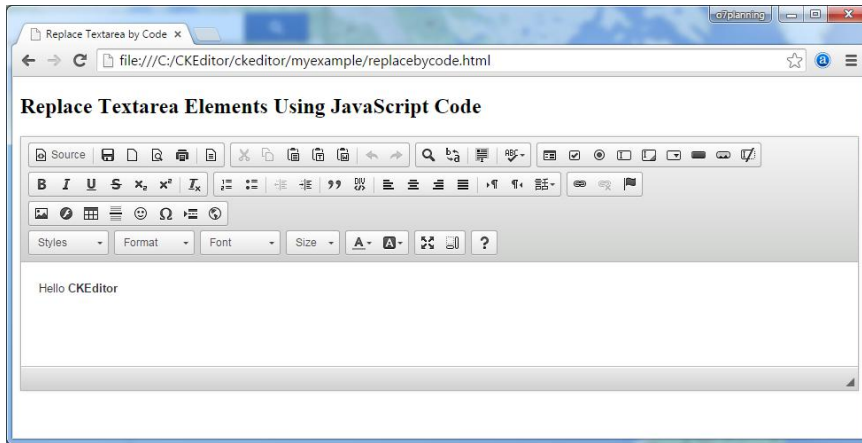
## Một số cú pháp cơ bản trong PostgreSQL

- Tạo bảng
  - `CREATE TABLE table_name ( column_name1 data_type(size), column_name2 data_type(size), column_name3 data_type(size), .... );`
  - Trong đó
    - `table_name` là tên của bảng mà bạn muốn tạo.
    - `column_name1` là tên cột thứ nhất của bảng. (Tương tự: `column_name2` và `column_name3` lần lượt là tên cột thứ hai và cột thứ ba).
    - `data_type` là kiểu dữ liệu của cột.
    - `size` là kích thước tối đa của kiểu dữ liệu của cột.
- Thêm dữ liệu vào bảng
  - `INSERT INTO table_name (column1,column2,column3)`
  - `VALUES (value1,value2,value3);`
  - Trong đó
    - `column1, column2, column3` lần lượt là tên của cột 1, cột 2, cột 3.
    - `value1, value2, value3` lần lượt là giá trị của cột 1, cột 2, cột 3.
- Cú pháp xóa
  - Để xóa database ta sử dụng cú pháp
  - `DROP DATABASE db_name;`
  - Để xóa bảng ta sử dụng cú pháp
  - `DROP TABLE tb_name;`
- Cú pháp truy xuất dữ liệu
  - `SELECT * FROM name_table;`
  - Cú pháp này sẽ truy xuất tất cả dữ liệu của bảng có tên là `name_table`.
  - `SELECT name_column1, name_column2 FROM name_table;`
  - Cú pháp này sẽ truy xuất tất cả dữ liệu của hai cột có tên là `name_column1` và `name_column2` trong bảng có tên là `name_table`.

### 2.3. Client

#### 2.3.1. CkEditor

Với đặc tính của forum là hỗ trợ người dùng soạn thảo công thức toán học một cách nhanh gọn nhất. Nên trình soạn thảo để người dùng có thể viết ra công thức toán học một cách nhanh gọn dễ dàng là vô cùng quan trọng. Nên Ckeditor kết hợp cùng



ReactJS là sự lựa chọn tốt.

Vậy Ckeditor là gì ?

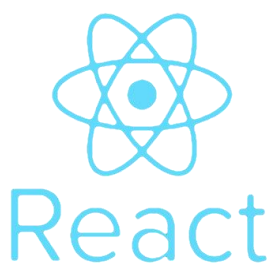
CKEditor (còn gọi là FCKeditor) là một trình soạn thảo mã nguồn mở theo kiểu WYSIWYG (tay làm - mắt thấy) của CKSource. Chương

trình này có thể tích hợp vào các website mà không cần cài đặt. Phiên bản đầu tiên được phát hành năm 2003 và đến nay được rất nhiều người sử dụng.[3]

Cũng giống các trình soạn thảo dành cho web khác, CKEditor sử dụng JavaScript là nền tảng, riêng việc tương tác với server thì CKEditor sử dụng các ngôn ngữ sau: Active-FoxPro, ASP, ASP.NET, ColdFusion, Java, JavaScript, Lasso, Perl, PHP và Python.[4]

CKEditor tương thích với hầu hết các trình duyệt Internet, gồm có: Internet Explorer 6.0+ (Windows), Firefox 2.0+, Safari 3.0+, Google Chrome (Windows), Opera 9.50+, and Camino 1.0+ (Apple).[4]

### 2.3.2. Ant design



Ant là tập hợp các components của React được xây dựng theo chuẩn thiết kế của Ant UED Team. Tương tự như chuẩn Material Design, Ant cung cấp hầu hết các component thông dụng trong ứng dụng web hiện đại, như Layout, Button, Icon, DatePicker, v.v... Bên cạnh đó Ant cũng có những component riêng thú vị, như LocaleProvider cho phép bạn thay đổi ngôn ngữ trên toàn ứng dụng.

Ant hiện đang có hơn 25k star trên Github.

Có thể coi Ant Design cho React là tập hợp của hầu hết các thư viện về React. Nó đáp ứng được hầu hết các yêu cầu của project của bạn mà bạn không phải cài thêm bất cứ thư viện nào nữa. Dưới đây là danh sách các component mà nó cung cấp:

- General: Button, Icon
- Layout: Grid, Layout
- Navigation: Affix, Breadcrumb, Dropdown, Menu, Pagination, Steps
- Data Entry: AutoComplete, Checkbox, Cascader, DatePicker, Form, InputNumber, Input, Mention, Rate, Radio, Switch, Slider, Select, TreeSelect, Transfer, TimePicker, Upload
- Data Display: Avatar, Badge, Collapse, Carousel, Card, Calendar, List, Popover, Tree, Tooltip, Timeline, Tag, Tabs, Table
- Feedback: Alert, Drawer, Modal, Message, Notification, Progress, Popconfirm, Spin, Skeleton
- Other: Anchor, BackTop, Divider, LocaleProvider

### 2.3.3. *ReactJS*

Reactjs là một thư viện **Javascript** mã nguồn mở hỗ trợ xây dựng các thành phần giao diện nhanh gọn và tiện lợi. Bình thường các lập trình viên sẽ nhúng javascript vào code HTML thông qua các attribute như AngularJS nhưng với Reactjs làm việc như một thư viện cho phép **nhúng HTML vào javascript thông qua JSX**. Qua đó bạn có thể dễ dàng lồng các đoạn HTML vào trong JSX làm cho các component dễ hiểu và dễ sử dụng hơn.

Trong Reactjs thường dùng javascript để thiết kế bố cục cho trang web, nhưng nhược điểm là cấu trúc khá là khó. Thay vào đó sử dụng JSX và nhúng các đoạn HTML vào javascript, ta thấy cú pháp dễ hiểu hơn và JSX cũng có thể tối ưu code khi biên soạn. Vừa dễ cho người lập trình mà vừa tiện cho việc biên dịch.

*Thành phần cơ bản của Reactjs là gì?*

Thành phần cơ bản của React được gọi là **components**. Syntax để viết HTML sử dụng Javascript để render. Bạn có thể tạo ra một component bằng các gọi phương thức `createClass` của đối tượng React, điểm bắt đầu khi tiếp cận với thư viện này. Có thể lồng nhiều component vào nhau thông qua lệnh `return` của phương thức `render`.

Trong một chương trình thì có rất nhiều các component, để đơn giản việc quản lý các component đó người ta sử dụng **redux**, **redux giống như 1 cái kho chứa các component** và khi dùng component nào thì chỉ cần gọi nó ra.

Virtual DOM không được tạo ra bởi Reactjs nhưng lại được sử dụng rất nhiều. Đây là một chuẩn của W3C được dùng để truy xuất code HTML hoặc XML. Các **Virtual DOM** sẽ được tạo ra khi chạy chương trình, đó là nơi chứa các component. Sử dụng DOM sẽ tiết kiệm được hiệu suất làm việc, khi có thay đổi gì Reactjs đều tính toán trước và việc còn lại chỉ là thực hiện chúng lên DOM.

## Lợi ích khi sử dụng ReactJs

Làm gì cũng phải có nguyên nhân, người dùng sử dụng Reactjs ngày càng nhiều thì Reactjs phải mang lại những lợi ích tốt.

- ReactJS giúp cho việc viết các đoạn code Javascript sẽ trở nên dễ dàng hơn vì nó sử dụng một cú pháp đặc biệt đó chính là cú pháp JSX. Thông qua JSX cho phép nhúng code HTML và Javascript.
- ReactJS cho phép Developer phá vỡ những cấu tạo UI phức tạp thành những component độc lập. Dev sẽ không phải lo lắng về tổng thể ứng dụng web, giờ đây Developer dễ dàng chia nhỏ các cấu trúc UI/UX phức tạp thành từng component đơn giản hơn.
- Đi kèm với ReactJS là rất nhiều các công cụ phát triển giúp cho việc debug code một cách dễ dàng hơn.
- Một trong những ưu điểm nữa của ReactJS đó là sự thân thiện với SEO. Hầu như các JS Frameworks không thân thiện với các tìm kiếm mặc dù đã được cải thiện nhiều nhưng dưới sự hỗ trợ của các render dữ liệu trả về dưới dạng web page giúp cho SEO chuẩn hơn.

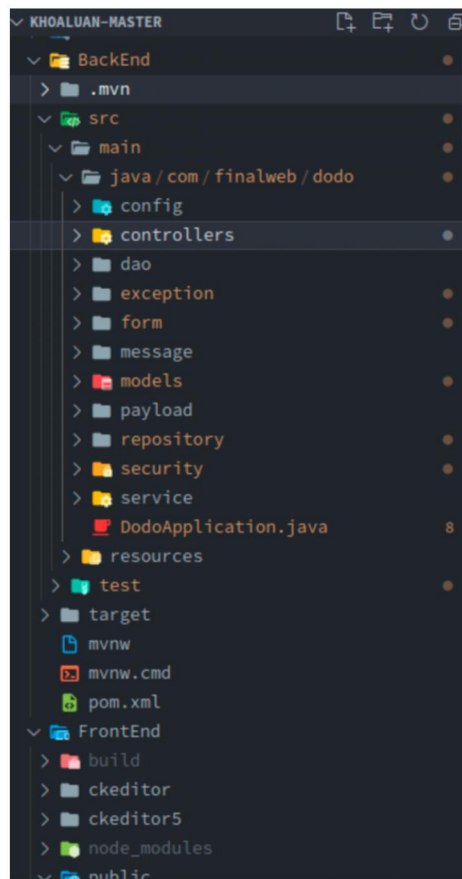
## 2.4. Thư mục mã nguồn

### 2.4.1. Môi trường cài đặt

Trong quá trình cài đặt, các giải pháp công nghệ được ứng dụng trong hệ thống bao gồm:

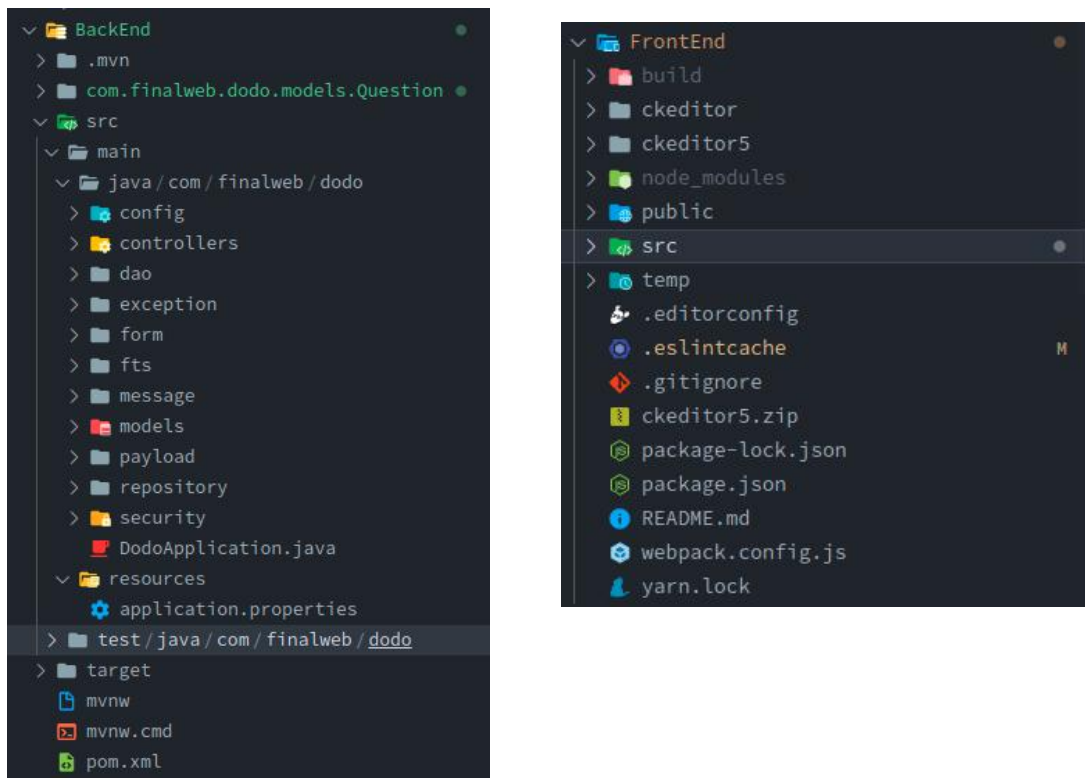
Yêu cầu	Công nghệ
Hệ điều hành	Linux
Môi trường ứng dụng	Java 11
IDE (Môi trường phát triển)	Visual Code
Quản lý source code	GitHub
Hệ quản trị CSDL	Postgresql
Công cụ quản lý tiến độ dự án	Trello
Công cụ tạo báo cáo thuyết trình	Google Slide
Công cụ xử lý văn bản	Google Doc

### 2.4.2. Cấu trúc thư mục mã nguồn



Hình 2.3. Cấu trúc thư mục mã nguồn

- FrontEnd: File chứa toàn bộ code Frontend
- BankEnd: File chứa toàn bộ code Backend



Hình 2.6. Cấu trúc file backend và frontend

**Kết luận chương:** Qua chương 2, em đã liệt kê và mô tả tổng quan các công nghệ được sử dụng trong hệ thống. Hiện tại tài liệu mới chỉ dừng lại các công nghệ đã và đang sử dụng trong hệ thống. Trong tương lai gần nhóm sẽ tiếp tục tìm hiểu các công nghệ mới để phát triển hệ thống một cách tốt hơn. Chương 3 sau đây em sẽ nói về thiết kế API, cơ sở dữ liệu và cài đặt thiết bị nhúng.

## CHƯƠNG 3 : ĐẶC TẢ HỆ THỐNG

Là khâu kỹ thuật đầu tiên của quá trình phát triển phần mềm. Thiếu nó không thể tiếp tục quá trình phối hợp của nhà phát triển và khách hàng. Nó quyết định chất lượng phần mềm đạt được với chi phí dự kiến và thời hạn cho trước.

### 3.1. Chức năng đăng ký

UC #01		ĐĂNG KÝ	Độ phức tạp: Trung Bình				
Mô tả		Chức năng cho phép để tạo một tài khoản cho việc đăng nhập vào hệ thống. Guest truy cập vào trang đăng ký của forum và nhập đầy đủ thông tin yêu cầu để tạo tài khoản.					
Tác nhân		Guest					
Tiền điều kiện		Người dùng truy cập vào trang đăng ký					
Hậu điều kiện	Thành công	Người dùng tạo tài khoản forum thành công.					
	Lỗi	Người dùng phải nhập lại thông tin đăng ký					
ĐẶC TẢ CHỨC NĂNG							
Luồng sự kiện chính							
<p><b>Luồng chính:</b></p> <p>Chức năng bắt đầu khi người dùng muốn đăng ký một tài khoản trên hệ thống thông qua username, password.</p> <ul style="list-style-type: none"><li>● Người dùng chọn chức năng đăng ký</li><li>● Hệ thống sẽ hiển thị giao diện đăng ký gồm những thông tin bắt buộc sau đây:</li></ul> <table><tr><td>Email</td><td>Password</td></tr><tr><td>Username</td><td>Check Password</td></tr></table> <ul style="list-style-type: none"><li>● Người dùng bấm ratio để xác nhận không phải là robot.</li><li>● Người dùng nhập đầy đủ thông tin yêu cầu, sau đó bấm nút đăng ký để hoàn thành chu trình đăng ký.</li></ul>				Email	Password	Username	Check Password
Email	Password						
Username	Check Password						
Luồng sự kiện phát sinh							

**Nhập thiếu thông tin tài khoản:**

- Nút đăng ký (registration button) sẽ bị lock không bấm được nếu bất kỳ trường input nào bị thiếu.
- Hệ thống sẽ thông báo đỏ ở dưới.

**Nhập sai định dạng password, username, email**

- Khi email hoặc username hoặc password bị sai định dạng.
- Hệ thống sẽ có thông báo đỏ và nút đăng ký sẽ bị lock.

**Mật khẩu và nhập lại mật khẩu không trùng nhau**

- Hai trường "Mật khẩu" và "Nhập lại mật khẩu" không trùng khớp nhau.
- Hệ thống sẽ hiển thị thông báo “mật khẩu không khớp”, người dùng cần nhập lại chính xác thông tin để có thể đăng ký tài khoản.

**Bỏ chọn “I am not robot”**

Khi người dùng ấn đăng ký, nếu bỏ chọn “I am not robot”, thì nút đăng ký sẽ bị khóa.

**Thông tin tài khoản đã tồn tại trong hệ thống**

- Khi người dùng ấn đăng ký, để đảm bảo tính duy nhất của mỗi tài khoản trên hệ thống, thông tin email hoặc username gửi lên sẽ được kiểm tra trên hệ thống để xem đã tồn tại người dùng này chưa.
- Nếu email hoặc username đã tồn tại trên hệ thống thì ứng dụng sẽ thông báo “Email hoặc username đã tồn tại” và người dùng cần nhập lại email hoặc username để có thể đăng ký tài khoản.

**Không thể đăng ký tài khoản do lỗi hệ thống**

Khi người dùng ấn đăng ký, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, trang web sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.

**Giao diện minh họa**



Đăng ký

User name

Email

Password

Confirm Password

☐

Tôi đã đọc [điều khoản dịch vụ](#)

Đăng ký

Hình 3.1. Hình minh họa màn hình đăng ký diễn đàn

3.2. Chức năng đăng nhập

UC #02		ĐĂNG NHẬP	Độ phức tạp: Trung Bình
Mô tả		Chức năng này dùng để cho người dùng đăng nhập vào hệ thống. Guest truy cập vào trang đăng nhập của website và nhập đầy đủ thông tin yêu cầu để đăng nhập. Sau khi đăng nhập vào hệ thống, người dùng có thể sử dụng chức năng của forum.	
Tác nhân		Guest	
Tiền điều kiện		Người dùng truy cập vào trang đăng nhập	
Hậu điều kiện	Thành công	Người đã đăng nhập sẽ được cấp role. tùy theo role sẽ có các quyền sử dụng các chức năng khác nhau của forum.	
	Lỗi	Người dùng phải nhập lại thông tin đăng nhập hoặc hủy thao tác.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			

**Luồng chính:**

Chức năng bắt đầu khi người dùng muốn đăng nhập một tài khoản trên hệ thống thông qua username, password.

- Sau khi người dùng chọn chức năng đăng nhập.
- Hệ thống sẽ hiển thị giao diện đăng ký gồm những thông tin bắt buộc sau đây:

Username	Check robot
Password	

- Người dùng bấm ratio để xác nhận không phải là robot.
- Người dùng nhập đầy đủ thông tin yêu cầu, sau đó bấm nút đăng nhập để hoàn thành chu trình đăng nhập.
- Hệ thống sẽ đăng nhập vào tài khoản nếu thông tin có trong cơ sở dữ liệu, nếu không có thông tin trong cơ sở dữ liệu thì hiển thị thông báo thông tin không chính xác.

**Luồng sự kiện phát sinh****Nhập thiếu thông tin:**

- Khi một trường thông tin nào đó bị điền thiếu.
- Hệ thống sẽ báo đỏ kèm thông báo nhỏ ở dưới.

**Nhập username hoặc password sai định dạng.**

- Khi trường input username bị sai định dạng.
- Hệ thống sẽ có thông báo đỏ thông báo và nút đăng nhập sẽ bị lock .

**Mật khẩu hoặc username sai**

- Khi hệ thống kiểm tra thông tin nhập vào của người dùng thấy mật khẩu và tên đăng nhập không trùng khớp. Hoặc tên đăng nhập hiện chưa tồn tại.
- Hệ thống sẽ đưa ra thông báo yêu cầu người dùng nhập chính xác tên đăng nhập với mật khẩu tương ứng.

**Bỏ chọn “I am not robot”**

Khi người dùng ấn đăng ký, nếu bỏ chọn “I am not robot”, thì nút đăng ký sẽ bị khóa.

**Không thể đăng nhập tài khoản do lỗi hệ thống**

Khi người dùng ấn đăng nhập, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, ứng dụng sẽ hiển thị thông báo cho người dùng biết là nhập sai tài khoản, mật khẩu hoặc gợi ý cách giải quyết sự cố nếu có.

Giao diện minh họa

Đăng nhập

🔍

username

🔒

password

👁️

☐ I am not robot

Quên mật khẩu

Đăng nhập

hoặc [Đăng ký ngay!](#)

Hình 3.2. Hình minh họa màn hình đăng nhập diễn đàn

### 3.3. Trang cá nhân

#### 3.3.1. Xem trang cá nhân

UC #3.1		XEM TRANG CÁ NHÂN	Độ phức tạp: Dễ
Mô tả		Chức năng này để xem thông tin trang cá nhân của bất cứ thành viên khác nào trong hệ thống mà người dùng đó có thể tìm kiếm được, hoặc được cho phép hiển thị với người dùng đó.	
Tác nhân		Guest, User, Admin, Moderator	
Tiền điều kiện		Không	
Hậu điều kiện	Thành công	Người dùng có thể tìm kiếm và xem được các thông tin trong trang cá nhân của một thành viên khác.	
	Lỗi	Không xem được trang cá nhân hoặc không có bất kỳ dữ liệu gì được hiển thị.	
ĐẶC TẢ CHỨC NĂNG			

### Luồng sự kiện chính

#### Luồng chính:

Chức năng này cho phép người dùng có thể xem trang cá nhân của một người bất kỳ. Trang cá nhân sẽ hiển thị những thông tin bao gồm:

- Email đã được mã hóa.
- Role của người đó.

#### Luồng con: Lịch sử câu hỏi

Chức năng cho phép người dùng xem danh sách câu hỏi của người chủ trang cá nhân.

### Luồng sự kiện phát sinh

#### Không thể xem trang cá nhân do lỗi hệ thống.

Khi người dùng thực hiện xem trang cá nhân, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, forum sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.

### Giao diện minh họa

## Thông tin cá nhân

[Thông tin cơ bản](#)

[Lịch sử câu hỏi](#)

### User Info

UserName: admin12anh

Email: adm\*\*\*\*@dodo.com

Hình 3.3. Hình minh họa màn hình thông tin cá nhân

### 3.3.2. Quản lý thông tin trang cá nhân

UC #3.2		QUẢN LÝ THÔNG TIN TRANG CÁ NHÂN	Độ phức tạp: Trung Bình				
Mô tả		Chức năng này dùng để quản lý các thông tin của một người dùng như: username và password, lịch sử đặt câu hỏi và lịch sử câu trả lời.					
Tác nhân		User, Moderator, Admin					
Tiền điều kiện		Người dùng đã đăng nhập					
Hậu điều kiện	Thành công	Khi đã cập nhật thông tin thành công thông tin cá nhân của công người dùng sẽ được lưu.					
	Lỗi	Người dùng cần thực hiện lại hoặc hủy thao tác thay đổi các thông tin cá nhân.					
ĐẶC TẢ CHỨC NĂNG							
Luồng sự kiện chính							
<p><b>Luồng chính:</b></p> <p>Chức năng này được thực hiện khi người dùng muốn xem các thông tin về bản thân hoặc muốn sửa đổi một số thông tin về bản thân:</p> <ul style="list-style-type: none"><li>● Hệ thống hiển thị các thông tin cá nhân của người dùng.</li><li>● Người dùng có thể sửa đổi một số thông tin cá nhân như: mật khẩu, email, ...</li></ul> <p><b>Luồng con: Chỉnh sửa thông tin cá nhân</b></p> <p>Chức năng này bắt đầu khi một người dùng muốn chỉnh sửa thông tin cá nhân của mình:</p> <ul style="list-style-type: none"><li>● Sau khi đăng nhập, người dùng mở menu phải và chọn Trang cá nhân để vào trang cá nhân của mình.</li><li>● Người dùng có thể chỉnh sửa username, email và password của mình .</li><li>● Hệ thống sẽ hiển thị giao diện chỉnh sửa gồm những thông tin sau đây:</li></ul> <table><tr><td>Username</td><td>Password</td></tr><tr><td>Email</td><td></td></tr></table>				Username	Password	Email	
Username	Password						
Email							

- Người dùng nhập đầy đủ thông tin cần chỉnh sửa, sau đó bấm nút lưu để hoàn thành chu trình chỉnh sửa.
- Nếu chỉnh sửa password, người dùng sẽ bị logout và phải đăng nhập lại bằng mật khẩu mới.

#### **Luồng con: Quản lý danh sách câu hỏi**

Chức năng bắt đầu khi một người dùng muốn quản lý câu hỏi mình đã đặt.

#### **Luồng con: Quản lý danh sách câu trả lời**

Chức năng này bắt đầu khi một người dùng muốn quản lý danh sách câu trả lời mình đã viết.

### **Luồng sự kiện phát sinh**

#### **Nhập thiếu thông tin:**

Nút lưu (submit) sẽ bị lock không bấm được, trường input nào bị thiếu sẽ báo đỏ kèm thông báo nhỏ ở dưới.

#### **Nhập sai định dạng username hoặc mật khẩu**

Khi có một trường input nào bị sai định dạng, thì sẽ có thông báo đỏ thông báo và nút submit sẽ bị lock

#### **Không thể chỉnh sửa thông tin tài khoản do lỗi hệ thống**

Khi người dùng ấn submit, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, forum sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.

#### **Không thể quản lý được câu hỏi hoặc câu trả lời do lỗi hệ thống**

Khi người yêu cầu quyền quản lý câu trả lời hoặc câu hỏi, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, forum sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.

#### **Chọn “Thoát” sau khi điền thông tin muốn sửa đổi**

Thông tin người dùng giữ nguyên, không thay đổi gì.

### **Giao diện minh họa**

# admin12anh

Thông tin cơ bản
Lịch sử câu hỏi
Lịch sử trả lời

---

Thông tin

Tên

admin12anh

Email

adm\*\*\*\*@dodo.com

Điểm

17

Chỉnh sửa thông tin cá nhân

Vai trò

ROLE\_MOD

ROLE\_ADMIN

Hình 3.4. Hình minh họa quản lý thông tin cá nhân

### 3.4. Quản lý câu hỏi

#### 3.4.1. Xem câu hỏi

UC #4.1		XEM CÂU HỎI	Độ phức tạp: Trung Bình
Mô tả		Chức năng này cho phép một người dùng có thể xem mọi câu hỏi có trên hệ thống.	
Tác nhân		User, Moderator, Admin,Guest	
Tiền điều kiện		Không có	
Hậu điều kiện	Thành công	Khi truy cập diễn đàn thành công. Danh sách câu hỏi mà người dùng đã đặt đã sẽ xuất hiện.	
	Lỗi	Người dùng không xem được câu hỏi do lỗi hệ thống.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			
Luồng chính: Chức năng này bắt đầu khi một người dùng truy cập vào diễn đàn: <ul style="list-style-type: none"><li>• Danh sách câu hỏi được hiện thì ở trang chủ.</li><li>• Lượng câu hỏi có trong hệ thống. Được phân trang 25 câu hỏi một trang. Sắp xếp theo thời gian được tạo.</li></ul>			

**Luồng con: Xem chi tiết câu hỏi**

Chức năng này cho phép người dùng có thể xem chi tiết về một câu hỏi, các nội dung có thể bao gồm:

- Xem đầy đủ nội dung câu hỏi
- Xem các câu trả lời của câu hỏi đó nếu có.
- Viết câu trả lời cho câu hỏi nếu đó là câu hỏi mở.

**Luồng con: Xem danh sách câu hỏi của người dùng**

Chức năng này cho phép người dùng có thể xem danh sách câu hỏi của người dùng, các nội dung có thể bao gồm:

- Xem đầy đủ nội dung câu hỏi
- Xem các câu trả lời của câu hỏi đó nếu có.
- Viết câu trả lời cho câu hỏi nếu đó là câu hỏi mở.

**Luồng sự kiện phát sinh****Không xem chi tiết câu hỏi trong các danh sách.**

Khi người dùng không thể xem, hoặc danh sách không còn hiển thị câu hỏi, thì có nghĩa là câu hỏi đã được xóa.

**Giao diện minh họa**



# yêu toán học

Đặt câu hỏi

2:47:57 PM at 4/4/2021

Nội dung:

$\frac{2}{3}$

toán

văn



admin12anh

0 câu trả lời



No Data

< 0 > 5 / page ▾

Viết bình luận

Hình 3.5. Hình minh họa chức năng xem câu hỏi

### 3.4.2. Thêm câu hỏi

UC #4.2		THÊM CÂU HỎI	Độ phức tạp: Trung Bình				
Mô tả		Chức năng cho phép một người dùng đặt câu hỏi .					
Tác nhân		User, Moderator, Admin					
Tiền điều kiện		Người dùng đã đăng nhập hệ thống.					
Hậu điều kiện	Thành công	Người dùng đặt câu hỏi thành công. Câu hỏi mới được lưu vào hệ thống					
	Lỗi	Người dùng không đặt được câu hỏi.					
ĐẶC TẢ CHỨC NĂNG							
Luồng sự kiện chính							
<p><b>Luồng chính:</b></p> <p>Chức năng này bắt đầu khi một người dùng chọn chức năng đặt câu hỏi:</p> <ul style="list-style-type: none"><li>Tại Trang chủ, nhấn tab "Đặt câu hỏi"</li><li>Form đặt câu hỏi gồm :</li></ul> <table><tr><td>Chủ đề</td><td>Nội dung</td></tr><tr><td>Tiêu đề</td><td></td></tr></table> <ul style="list-style-type: none"><li>Người dùng điền đầy đủ thông tin cần thiết và ấn submit. Câu hỏi của người dùng sẽ được tạo ra và lưu trữ trên hệ thống. Sau đó người dùng sẽ được đưa về trang chủ và hiện ra thông báo đặt câu hỏi thành công.</li></ul>				Chủ đề	Nội dung	Tiêu đề	
Chủ đề	Nội dung						
Tiêu đề							
Luồng sự kiện phát sinh							
<p><b>Người dùng để trống thanh tiêu đề hoặc thanh nội dung</b></p> <ul style="list-style-type: none"><li>Người dùng đã để trống thanh tiêu đề hoặc thanh nội dung hoặc cả hai</li><li>Nút submit sẽ bị lock. Hệ thống sẽ có thông báo.</li></ul> <p><b>Không đặt được câu hỏi được do lỗi hệ thống.</b></p> <p>Khi người dùng thực hiện tạo câu hỏi, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, ứng dụng sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.</p>							

**Giao diện minh họa**

Đặt câu hỏi công khai

**Tiêu đề câu hỏi**

Mô tả ngắn gọn nội dung câu hỏi của bạn

vi dụ như. 1+1= , nhiệt lượng của mặt trời,...

**Nội dung**

Viết toàn bộ thông tin về câu hỏi để mọi người có thể trả lời câu hỏi của bạn

I ✓

**Chủ đề**

Thêm chủ đề để mô tả câu hỏi của bạn

vi dụ như. đại số tuyến tính , giải tích,...

Submit

Hình 3.6. Hình minh họa chức năng thêm câu hỏi

3.4.3. Sửa câu hỏi

UC #4.3		SỬA CÂU HỎI	Độ phức tạp: Trung Bình
Mô tả		Chức năng này cho phép một người dùng sửa câu hỏi.	
Tác nhân		User, Moderator, Admin	
Tiền điều kiện		Người dùng đã đăng nhập thành công.	
Hậu điều kiện	Thành công	Sửa câu hỏi thành công.	
	Lỗi	Người dùng cần thực hiện lại hoặc hủy thao tác.	

<b>ĐẶC TẢ CHỨC NĂNG</b>
<b>Luồng sự kiện chính</b>
<p><b>Luồng chính:</b></p> <p>Chức năng này bắt đầu khi một người dùng muốn sửa câu hỏi:</p> <ul style="list-style-type: none"> <li>● Người dùng bấm vào trang cá nhân của mình.</li> <li>● Mở tab lịch sử câu hỏi</li> <li>● Ấn nút Edit của câu hỏi muốn chỉnh sửa.</li> <li>● Thực hiện chỉnh sửa câu hỏi:</li> <li>● Người dùng chỉnh sửa trực tiếp nội dung, chủ đề hoặc tiêu đề của câu hỏi.</li> <li>● Người dùng có thể khóa bình luận của câu hỏi</li> <li>● Bấm nút submit để hoàn thành chu trình chỉnh sửa câu hỏi.</li> <li>● Hệ thống cập nhật lại thông tin của câu hỏi. Sau đó người dùng sẽ quay lại màn hình trước khi sửa câu hỏi và hiện ra thông báo xác nhận sửa câu hỏi thành công.</li> </ul>
<b>Luồng sự kiện phát sinh</b>
<p><b>Không thể sửa câu hỏi do người dùng nhập thiếu thông tin.</b></p> <ul style="list-style-type: none"> <li>● Khi người dùng nhập thiếu một số trường thông tin bắt buộc.</li> <li>● Forum sẽ thông báo để người dùng biết và nhập cho đầy đủ.</li> </ul> <p><b>Không thể sửa câu hỏi do lỗi hệ thống</b></p> <p>Khi người dùng sửa câu hỏi, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, ứng dụng sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.</p>
<b>Giao diện minh họa</b>

Sửa câu hỏi

title: xxx

content: ✓

$x = \sqrt{3}$  toán học

status: Pending

Submit

Hình 3.7. Hình minh họa chức năng sửa câu hỏi

#### 3.4.4. Xóa câu hỏi

UC #4.4		XÓA CÂU HỎI	Độ phức tạp: Trung Bình
Mô tả		Chức năng này cho phép một người dùng xóa câu hỏi.	
Tác nhân		User, Moderator, Admin	
Tiền điều kiện		Người dùng đã đăng nhập thành công.	
Hậu điều kiện	Thành công	Xóa câu hỏi thành công.	
	Lỗi	Người dùng cần thực hiện lại hoặc hủy thao tác.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			
<b>Luồng chính:</b> Chức năng này bắt đầu khi một người dùng muốn xóa câu hỏi: <ul style="list-style-type: none"><li>Sau khi người dùng đã đăng nhập thành công.</li><li>Mở trang cá nhân</li><li>Mở tab lịch sử câu hỏi</li></ul>			

- Nhấn button delete của câu hỏi muốn xóa
- Hiện thị thông báo thành công nếu xóa thành công.

#### **Luồng con: Xóa câu trả lời**

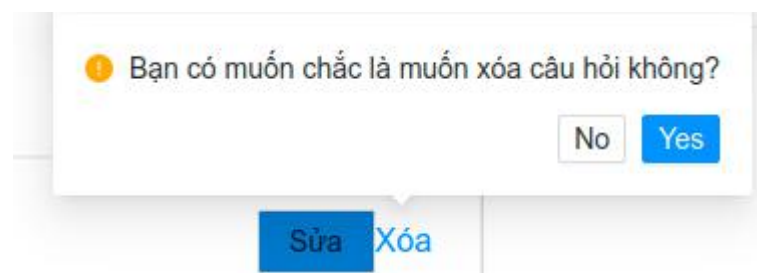
Sau khi xóa câu hỏi, tất cả câu trả lời có liên quan tới câu hỏi đó sẽ bị xóa cùng.

#### **Luồng sự kiện phát sinh**

##### **Không thể xóa câu hỏi do lỗi hệ thống**

Khi người dùng xóa câu hỏi, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, ứng dụng sẽ hiện thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.

#### **Giao diện minh họa**



Hình 3.8. Hình minh họa chức năng xóa câu hỏi

### **3.5. Quản lý câu trả lời**

#### **3.5.1. Xem câu trả lời**

UC #5.1		XEM CÂU TRẢ LỜI	Độ phức tạp: Trung Bình
Mô tả		Chức năng này cho phép một người dùng có thể xem các câu trả lời có trong các câu hỏi trên hệ thống.	
Tác nhân		User, Moderator, Admin, Guest	
Tiền điều kiện		Không có	
Hậu điều kiện	Thành công	Xem chi tiết câu hỏi. Nếu câu hỏi có câu trả lời thì list câu trả lời sẽ được hiển thị.	
	Lỗi	Người dùng không xem được câu trả lời do lỗi hệ thống.	
ĐẶC TẢ CHỨC NĂNG			

## Luồng sự kiện chính

### Luồng chính:

Chức năng này bắt đầu khi một người dùng bấm vào xem chi tiết một câu hỏi:

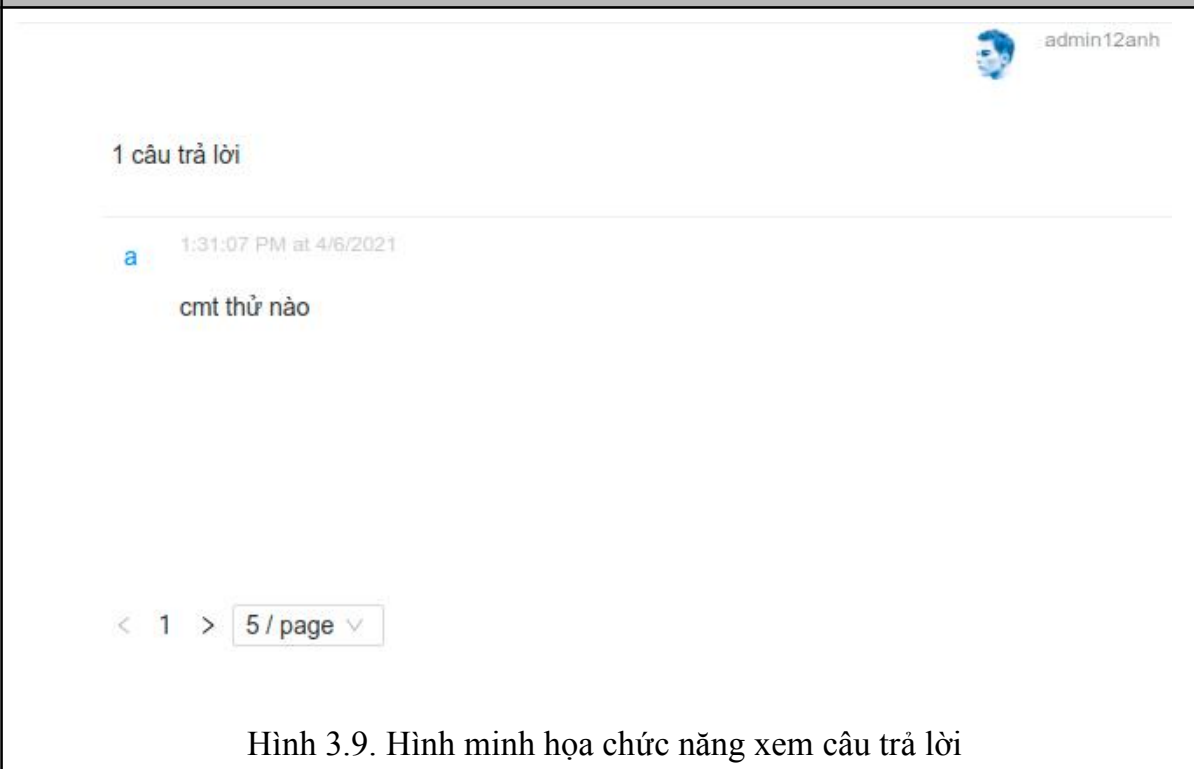
- Danh sách câu trả lời của câu hỏi đó (nếu có) sẽ được hiển thị.
- Lượng câu trả lời được phân trang, 5 câu trả lời một trang. Sắp xếp theo thời gian được tạo.

## Luồng sự kiện phát sinh

### Không xem được list câu trả lời do lỗi hệ thống.

Khi người dùng không thể xem, hoặc danh sách không còn hiển thị câu trả lời, thì có nghĩa là câu trả lời đó đã được xoá.

## Giao diện minh họa



Hình 3.9. Hình minh họa chức năng xem câu trả lời

### 3.5.2. Viết câu trả lời


UC #5.2		VIẾT CÂU TRẢ LỜI	Độ phức tạp: Trung Bình
Mô tả		Chức năng này cho phép một người dùng đã đăng nhập vào hệ thống, có thể viết câu trả lời .	
Tác nhân		User, Moderator, Admin	
Tiền điều kiện		Người dùng đã đăng nhập hệ thống.	
Hậu điều kiện	Thành công	Người dùng viết câu trả lời thành công. Câu trả lời mới được lưu vào hệ thống	
	Lỗi	Người dùng không viết được câu trả lời.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			
<p><b>Luồng chính:</b></p> <p>Chức năng này bắt đầu khi một người dùng chọn chức năng viết câu trả lời:</p> <ul style="list-style-type: none"><li>● Form viết câu trả lời</li><li>● Người dùng có thể thêm công thức toán học. Sau khi bấm biểu tượng nhập biểu thức toán học. Một hộp công cụ sẽ hiện ra và người dùng sẽ viết công thức mà mình muốn.</li><li>● Người dùng không được để trống ô nội dung.</li><li>● Ấn submit</li></ul>			
Luồng sự kiện phát sinh			
<p><b>Không thể viết câu trả lời do người dùng đã để trống thanh nội dung.</b></p> <ul style="list-style-type: none"><li>● Khi người dùng viết câu trả lời nếu nhập thiếu thanh nội dung</li><li>● Forum sẽ thông báo để người dùng biết và nhập cho đầy đủ.</li></ul> <p><b>Không viết được câu trả lời được do lỗi hệ thống.</b></p> <p>Khi người dùng thực hiện viết câu trả lời, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, ứng dụng sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.</p> <p><b>Không ấn submit</b></p>			



Trong luồng “viết câu trả lời”, nếu người dùng không ấn submit, hệ thống sẽ không có thay đổi gì về dữ liệu trên hệ thống, câu trả lời sẽ không được tạo mới.

**Giao diện minh họa**

Viết bình luận



Hình 3.10. Hình minh họa chức năng viết câu trả lời

### 3.5.3. Sửa câu trả lời

UC #5.3		SỬA CÂU TRẢ LỜI	Độ phức tạp: Trung Bình
Mô tả		Chức năng này cho phép một người dùng có thể sửa một câu trả lời.	
Tác nhân		User, Moderator, Admin	
Tiền điều kiện		Người dùng đã đăng nhập thành công.	
Hậu điều kiện	Thành công	Người dùng có thể sửa câu trả lời.	
	Lỗi	Người dùng cần thực hiện lại hoặc hủy thao tác.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			
<b>Luồng chính:</b> Chức năng này cho phép người dùng sửa câu trả lời mà họ đã viết. <ul style="list-style-type: none"><li>● Người dùng truy cập trang cá nhân của mình.</li><li>● Bấm vào phần danh sách câu trả lời</li></ul>			

- Bấm button chỉnh sửa của câu trả lời muốn chỉnh sửa.
- Nhập phần nội dung của câu trả lời.
- Sau khi sửa xong, hệ thống cập nhật lại thông tin bình luận. Sau đó người dùng sẽ quay lại màn hình trước khi sửa câu trả lời và hiện ra thông báo xác nhận sửa chuyển đi thành công.

### Luồng sự kiện phát sinh

#### Không thể sửa được câu trả lời do lỗi hệ thống

Khi người dùng ra lệnh lưu lại câu trả lời đã chỉnh sửa, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, forum sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.

### Giao diện minh họa

Sửa bình luận




hi chào

Submit

Hình 3.11. Hình minh họa chức năng sửa câu trả lời

#### 3.5.4. Xóa câu trả lời

UC #5.4		XÓA CÂU TRẢ LỜI	Độ phức tạp: Trung Bình
<b>Mô tả</b>		Chức năng này cho phép một người dùng có thể xóa câu trả lời.	
<b>Tác nhân</b>		User, Moderator, Admin	
<b>Tiền điều kiện</b>		Người dùng đã đăng nhập thành công.	
<b>Hậu điều kiện</b>	<b>Thành công</b>	Người dùng có thể xóa câu trả lời.	
	<b>Lỗi</b>	Người dùng cần thực hiện lại hoặc hủy thao tác.	

ĐẶC TẢ CHỨC NĂNG
<b>Luồng sự kiện chính</b>
<p><b>Luồng chính:</b></p> <p>Chức năng này cho phép người dùng xóa câu trả lời mà họ có trên hệ thống.</p> <ul style="list-style-type: none"> <li>● Người dùng truy cập trang cá nhân của mình.</li> <li>● Bấm vào phần danh sách câu trả lời</li> <li>● Bấm button delete của câu trả lời muốn xóa.</li> <li>● Ứng dụng sẽ quay lại màn hình trước khi xóa câu trả lời và hiện ra thông báo xác nhận xóa câu trả lời thành công.</li> </ul>
<b>Luồng sự kiện phát sinh</b>
<p><b>Không thể xóa được câu trả lời do lỗi hệ thống</b></p> <p>Khi người dùng ra lệnh xóa câu trả lời, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, forum sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.</p>
<b>Giao diện minh họa</b>
 <p>The screenshot shows a forum post with the text "hi chào 1122" and a date "4/7/2021". To the right of the post is a blue "Edit" button and a red "Xóa" (Delete) button.</p>

Hình 3.12. Hình minh họa chức năng xóa câu trả lời

### 3.6. Quản lý chủ đề

UC #07	QUẢN LÝ CHỦ ĐỀ	Độ phức tạp: Khó
<b>Mô tả</b>	Chức năng này cho phép quản lý chủ đề có trong hệ thống hệ thống.	
<b>Tác nhân</b>	Moderator, Admin	
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập thành công và có role là Admin hoặc Moderator.	
<b>Hậu điều</b>	<b>Thành</b>	Người dùng có thể thêm chủ đề vào hệ thống.

kiện	công	
	Lỗi	Người dùng cần thực hiện lại hoặc hủy thao tác.
<b>ĐẶC TẢ CHỨC NĂNG</b>		
<b>Luồng sự kiện chính</b>		
<p><b>Luồng chính:</b></p> <p>Chức năng này cho phép Admin hoặc Moderator có thể thêm xóa hoặc sửa chủ đề hệ thống.</p> <p>Khi người dùng đăng nhập vào hệ thống. Hệ thống sẽ kiểm tra vai trò của người dùng đó.</p> <p>Tính năng thêm chủ đề sẽ được mở, nếu role của người đó là Admin hoặc Mod</p> <ul style="list-style-type: none"> <li>● Người sử dụng có thể lựa chọn “Tạo mới” để tạo mới tài khoản cho Node, luồng con “Tạo mới Node” được thực hiện.</li> <li>● Người sử dụng có thể lựa chọn “Sửa” để sửa thông tin chủ đề, luồng con “Sửa thông tin chủ đề” được thực hiện.</li> <li>● Người sử dụng có thể lựa chọn “Xóa” chủ đề, luồng con “Xóa chủ đề” được thực hiện.</li> </ul> <p><b>Luồng con: Thêm mới chủ đề</b></p> <p>Để có thể đáp ứng được nhu cầu trao đổi thông tin về nhiều lĩnh vực khác nhau của người sử dụng. Chương trình cho phép người quản lý thêm chủ đề cho diễn đàn.</p> <ul style="list-style-type: none"> <li>● <i>Ví dụ:</i> diễn đàn đã có các chủ đề trao đổi về mạng, Internet-web, lập trình. Nếu người quản thấy cảm thấy nhu cầu của người sử dụng cần trao đổi các vấn đề về Cơ sở dữ liệu thì người quản lý có thể thêm chủ đề này vào cơ sở dữ liệu.</li> <li>● Admin hoặc Mod sẽ bấm vào nút thêm tag để thêm chủ đề.</li> <li>● Khi tất cả các trường input được điền đầy đủ. Nút submit sẽ được unlock.</li> <li>● Admin hoặc Moderator bấm submit.Chủ đề sẽ được tạo ra và lưu trữ trên hệ thống. Sau đó người dùng sẽ quay lại màn hình trước khi tạo chủ đề và hiện ra thông báo xác nhận tạo chủ đề thành công.</li> </ul> <p><b>Luồng con: Sửa thông tin chủ đề</b></p> <p>Trong danh sách chủ đề, Admin hoặc Moderator bấm button "Edit" để sửa.</p> <ul style="list-style-type: none"> <li>● Người dùng chỉnh sửa tên hoặc mô tả của chủ đề</li> <li>● Người dùng bấm submit</li> </ul> <p><b>Luồng con: Xóa chủ đề</b></p> <p>Trong danh sách chủ đề, Admin hoặc Moderator bấm button "Delete" để xóa.</p>		

- Người dùng chỉnh sửa tên hoặc mô tả của chủ đề
- Người dùng bấm submit

## Luồng sự kiện phát sinh

### Thiếu thông tin khi thêm mới chủ đề

- Trong luồng con “Thêm mới chủ đề”, tất cả thông tin đều bắt buộc phải điền nếu sử dụng điền thiếu, hệ thống sẽ thông báo cần phải điền đầy đủ các trường.
- Người sử dụng điền đầy đủ thông tin mới thực hiện được thao tác Lưu.

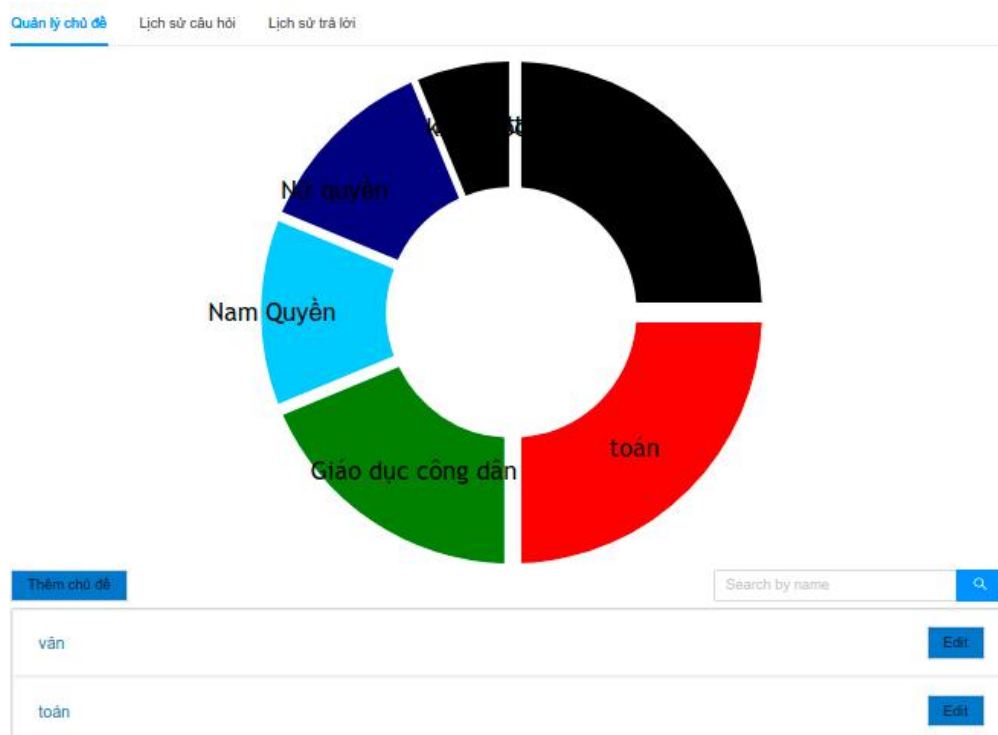
### Thiếu thông tin khi sửa chủ đề

- Trong luồng con “Sửa chủ đề”, tất cả thông tin đều bắt buộc phải điền nếu sử dụng điền thiếu, hệ thống sẽ thông báo cần phải điền đầy đủ các trường.
- Người sử dụng điền đầy đủ thông tin mới thực hiện được thao tác Lưu.

### Lỗi không xóa được chủ đề do lỗi hệ thống

Khi Admin, Moderator xóa chủ đề, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, forum sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có.

## Giao diện minh họa



Hình 3.13. Hình minh họa chức năng quản lý chủ đề

### 3.7. Quản lý người dùng

UC #07		QUẢN LÝ NGƯỜI DÙNG	Độ phức tạp: Trung Bình				
Mô tả		Chức năng này cho phép thêm người dùng vào hệ thống.					
Tác nhân		Moderator, Admin					
Tiền điều kiện		Người dùng đã đăng nhập thành công và có role là Admin hoặc Moderator.					
Hậu điều kiện	Thành công	Người dùng có thể thêm người vào hệ thống.					
	Lỗi	Người dùng cần thực hiện lại hoặc hủy thao tác.					
ĐẶC TẢ CHỨC NĂNG							
Luồng sự kiện chính							
<p><b>Luồng chính:</b></p> <p>Chức năng này cho phép Admin hoặc Moderator có thể thêm nhanh người dùng vào hệ thống.</p> <ul style="list-style-type: none"><li>● Khi người dùng đăng nhập vào hệ thống. Hệ thống sẽ kiểm tra vai trò (role) của người dùng đó.</li><li>● Nếu vai trò của người đó là Admin hoặc Moderator.</li><li>● Tính năng thêm người dùng sẽ được mở</li><li>● Tính năng quản lý câu hỏi được mở</li><li>● Tính năng quản lý câu trả lời được mở</li><li>● Header của trang forum sẽ có thêm menu "modbroad".</li></ul> <p><b>Luồng con: Thêm người dùng nhanh</b></p> <p>Chức năng này cho phép người dùng có thể thêm nhiều người khác (Chỉ hoạt động với Admin và Moderator).</p> <ul style="list-style-type: none"><li>● Tại menu danh sách người dùng.</li><li>● Admin hoặc Moderator bấm button thêm người dùng.</li><li>● 1 form thêm người dùng sẽ xuất hiện như form đăng ký gồm:</li></ul> <table><tr><td>mật khẩu</td><td>Tên đăng nhập</td></tr><tr><td>email</td><td></td></tr></table>				mật khẩu	Tên đăng nhập	email	
mật khẩu	Tên đăng nhập						
email							

- Admin hoặc Moderator phải điền đầy đủ thông tin yêu cầu.
- Khi tất cả các thông tin của các thanh input được điền đầy đủ, chính xác. Nút submit sẽ được unlock.
- Hệ thống tạo mới tài khoản

#### **Luồng con: Sửa thông tin người dùng**

- Chức năng thực hiện trên pgAdmin4.
- Admin chọn vào bảng users.
- Hệ thống hiển thị danh sách toàn bộ người dùng có trong hệ thống.
- Admin chọn người muốn sửa thông tin
- Admin ấn cập nhật

#### **Luồng con: Xóa câu hỏi**

Chức năng này cho phép người dùng có thể xóa câu hỏi của bất cứ người nào, việc này sẽ xóa hoàn toàn câu hỏi đó ra khỏi hệ thống, đồng thời xóa những tương tác với chính là những câu trả lời của câu hỏi đó.

#### **Luồng con: Xóa câu trả lời**

Chức năng này cho phép người dùng có thể xóa câu trả lời của bất cứ người nào.

### **Luồng sự kiện phát sinh**

#### **Không thể thêm người dùng do lỗi hệ thống**

Khi Admin, Moderator lưu người dùng, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, ứng dụng sẽ hiển thị thông báo và gợi ý cách giải quyết sự cố nếu có.

#### **Điền thiếu hoặc sai định dạng thông tin người dùng trong luồng “thêm mới người dùng”**

- Tính năng thêm mới người dùng được cài đặt giống chức năng đăng ký
- Tất cả các trường thông tin là bắt buộc không được bỏ trống và phải đúng định dạng quy định.
- Người sử dụng điền đầy đủ thông tin mới thực hiện được thao tác Lưu.

#### **Không thể xóa câu hỏi do lỗi hệ thống**

Khi Admin, Moderator dùng ấn chức năng delete câu hỏi, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, forum sẽ hiển thị thông báo và gợi ý cách giải quyết sự cố nếu có.

#### **Không thể xóa câu trả lời do lỗi hệ thống**

Khi người dùng ấn delete, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, forum sẽ hiển thị thông báo và gợi ý cách giải quyết sự cố nếu có.

### **Giao diện minh họa**

[Quản lý chủ đề](#)[Quản lý câu hỏi](#)[Quản lý câu trả lời](#)[Quản lý người dùng](#)

Tìm kiếm

admin

cuongbn1234		Xóa
0		
tuananh		Xóa
1		
dbuser511		Xóa
0		
admin1		Xóa
0		
admin321		Xóa
0		
dbuser51		Xóa
0		

Hình 3.14. Hình minh họa chức năng quản lý người dùng



### 3.8. Chức năng đăng xuất

UC #08		ĐĂNG XUẤT	Độ phức tạp: Trung Bình
Mô tả		Chức năng cho phép người dùng muốn thoát hẳn khỏi trạng thái đăng nhập hệ thống, người dùng sau khi đã đăng xuất sẽ không thể thực hiện chức năng nào nữa, mà bắt buộc phải đăng nhập lại khi muốn sử dụng hệ thống.	
Tác nhân		User, Moderator, Admin	
Tiền điều kiện		Người dùng đã đăng nhập vào hệ thống.	
Hậu điều kiện	Thành công	Sau khi đăng xuất thành công hệ thống sẽ điều hướng về trang chủ và người sử dụng không thể tiếp tục thực hiện các chức năng cho phép.	
	Lỗi	Vẫn sử dụng được các chức năng quản lý có trong hệ thống hoặc hủy bỏ thao tác.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			
Luồng chính:			
<ul style="list-style-type: none"><li>Chức năng này cho phép người dùng đăng xuất khỏi tài khoản đang đăng nhập trong ứng dụng.</li><li>Sau khi đăng xuất, người dùng sẽ không thể sử dụng các chức năng bên trong forum cho tới khi đăng nhập trở lại.</li></ul>			
Giao diện minh họa			
<div><div>Menu</div><div><div><div></div><div>admin12346anh</div></div><div><div></div><div>Mod Mode</div></div><div><div></div><div>Đăng xuất</div></div></div></div>			

Hình 3.15. Hình minh họa chức năng đăng xuất

Hình 3.15. Hình minh họa chức năng đăng xuất

### 3.9. Chức năng quên mật khẩu

UC #09		QUÊN MẬT KHẨU	Độ phức tạp: Trung Bình
Mô tả		Chức năng này cho phép người dùng muốn cài lại mật khẩu khi quên.	
Tác nhân		Guest	
Tiền điều kiện		Không	
Hậu điều kiện	Thành công	Người dùng cài mới được mật khẩu của mình.	
	Lỗi	Người dùng cần thực hiện lại hoặc hủy thao tác.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính			
<p><b>Luồng chính:</b></p> <p>Chức năng bắt đầu khi người dùng muốn reset mật khẩu một tài khoản có trên hệ thống thông qua email.</p> <ul style="list-style-type: none"><li>● Người dùng chọn chức năng quên mật khẩu.</li><li>● Hệ thống sẽ hiển thị giao diện đăng ký gồm những thông tin bắt buộc sau đây:</li><li>● Email</li><li>● Password</li><li>● Check Password</li><li>● Người dùng nhập đầy đủ thông tin yêu cầu, sau đó bấm nút đăng ký để hoàn thành chu trình đăng ký.</li></ul>			
Luồng sự kiện phát sinh			
<p><b>Không thể reset mật khẩu do lỗi hệ thống</b></p> <p>Khi người dùng ấn reset, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, forum sẽ hiển thị thông báo và gợi ý cách giải quyết sự cố nếu có.</p>			
Giao diện minh họa			

Cài lại mật khẩu

---

[Reset Password](#)

[Đăng nhập](#)

hoặc [Đăng ký ngay!](#)

Hình 3.16. Hình minh họa chức năng quên mật khẩu

**Kết luận chương:** Thông qua chương 3, ta có thể hình dung được chi tiết cách các chức năng hoạt động. Từ đó sang chương 4, ta có thể dễ dàng thiết kế được Database, và API.

# CHƯƠNG 4. THIẾT KẾ DATABASE, API

*Trong phần này, em sẽ giới thiệu các bảng cơ sở dữ liệu dùng trong hệ thống và danh sách các API.*

## 4.1. Mô tả cơ sở dữ liệu

Các thực thể chính:

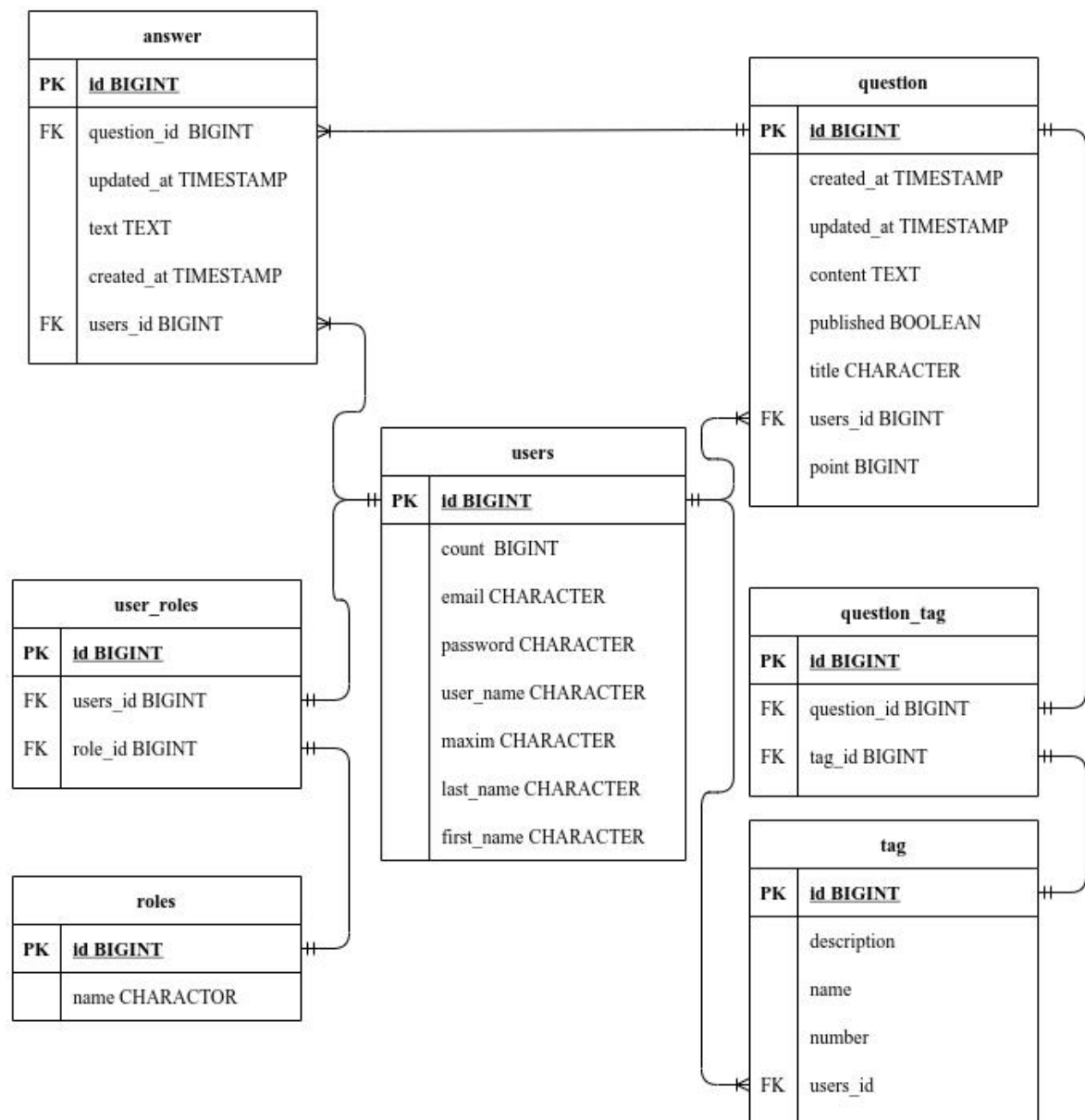
- Người dùng
- Câu hỏi
- Câu trả lời
- Chủ đề

Ví dụ mối quan hệ giữa các thực thể với nhau:

- Một người dùng có một hoặc nhiều câu hỏi.
- Một câu hỏi chỉ có thể đến từ một người.
- Một người có thể có nhiều câu trả lời.
- Mỗi câu trả lời chỉ thuộc về duy nhất một người và một câu hỏi cụ thể nào đó
- Mỗi câu hỏi có thể có 0 hoặc 1 hoặc nhiều chủ đề khác nhau
- Một chủ đề có thể liên quan tới nhiều câu hỏi.

## 4.2. Thiết kế database

### 4.2.1. Lược đồ cơ sở dữ liệu



Hình 4.1. Lược đồ cơ sở dữ liệu

#### 4.2.2. Các bảng trong cơ sở dữ liệu

STT	Tên bảng	Ý nghĩa
1	users	Lưu trữ thông tin người dùng
2	roles	Lưu trữ dữ liệu về vai trò (role) trong hệ thống
3	question	Lưu trữ dữ liệu về câu hỏi
4	answer	Lưu trữ dữ liệu về câu trả lời
5	tag	Lưu trữ dữ liệu về chủ đề

STT	Tên bảng	Ý nghĩa
6	user_roles	Lưu trữ mối quan hệ của người dùng và vai trò của họ trong hệ thống
7	question_tag	Lưu trữ mối quan hệ của câu hỏi và tag trong hệ thống

#### 4.2.3. Bảng Users

STT	Tên trường	Kiểu	Allow Null	Ghi chú
1	id	Integer	No	ID-Khóa chính
2	email	Varchar	No	Email người dùng đăng ký sử dụng hệ thống
3	username	Varchar	No	Username người dùng đăng ký sử dụng hệ thống
4	password	Varchar	No	Mật khẩu đã được encrypted
5	count	Integer	No	Điểm số của người dùng
6	maxim	Varchar	YES	Câu châm ngôn cá nhân
7	last_name	Varchar	YES	Họ của người dùng
8	first_name	Varchar	YES	Tên của người dùng

#### 4.2.4. Bảng Role

STT	Tên trường	Kiểu	Allow Null	Ghi chú
1	id	Integer	No	ID-Khóa chính
2	name	Varchar	No	Tên của vai trò

#### 4.2.5. Bảng Tag

STT	Tên trường	Kiểu	Allow Null	Ghi chú
1	id	Integer	No	ID-Khóa chính
2	name	Varchar	No	Tên của chủ đề
3	description	Varchar	No	Mô tả chủ đề

#### 4.2.6. Thiết kế bảng question

STT	Tên trường	Kiểu	Allow Null	Ghi chú
1	id	Integer	No	ID-Khóa chính
2	title	Varchar	No	Tiêu đề của câu hỏi
3	tag	Table	Yes	Danh sách chủ đề
4	content	Varchar	No	Nội dung câu hỏi
5	users_id	Integer	No	Khoá ngoại
6	published	Boolean	No	Khóa câu hỏi
7	point	Long	Yes	Điểm số (Nếu có)
8	createdAt	Date	No	Thời gian tạo

STT	Tên trường	Kiểu	Allow Null	Ghi chú
9	updatedAt	Date	No	Thời gian chỉnh sửa lần cuối

#### 4.2.7. Thiết kế bảng answer

STT	Tên trường	Kiểu	Allow Null	Ghi chú
1	Id	Long	No	ID-Khóa chính
2	text	Varchar	No	Nội dung câu trả lời
3	users_id	Long	No	Khoá ngoại
4	question_id	Long	No	Khoá ngoại
5	createdAt	Date	No	Thời gian tạo
6	updatedAt	Date	No	Thời gian chỉnh sửa lần cuối

#### 4.2.8. Thiết kế bảng user\_roles

STT	Tên trường	Kiểu	Allow Null	Ghi chú
1	id	Integer	No	ID-Khóa chính
2	users_id	Long	No	Khoá ngoại
3	role_id	Long	No	Khoá ngoại



#### 4.2.9. Thiết kế bảng *question\_tag*

STT	Tên trường	Kiểu	Allow Null	Ghi chú
1	id	Long	No	ID-Khóa chính
2	tag_id	Long	No	Khoá ngoại
3	question_id	Long	No	Khoá ngoại

#### 4.3. Tổng quan API

Hệ thống API của chương trình sẽ được cung cấp bởi Spring Boot. Các APIs sẽ được thực hiện theo phong cách RESTful API.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

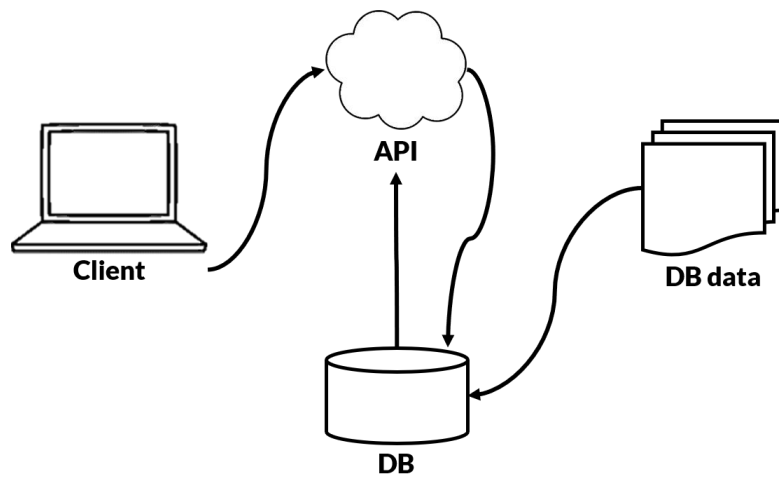
##### Diễn giải các thành phần:

- **API (Application Programming Interface)** là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như **JSON** hay XML.
- **REST (REpresentational State Transfer)** là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.
- **RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.
- Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.

**Ví dụ về RESTFUL.** Với một tài nguyên A, API của tài nguyên A sẽ bao gồm:

- GET /api/A/: lấy mọi thực thể của tài nguyên A
- GET /api/A/id: lấy thực thể với khóa chính là id của tài nguyên A
- POST /api/A/: tạo một thực thể tại tài nguyên A
- PUT /api/A/id: ghi đè/ sửa thực thể với khóa chính là id của tài nguyên A
- DELETE /api/A/id: xóa thực thể với khóa chính là id của tài nguyên A

#### **4.3.1. Cấu trúc của một request**



Trong project này, service được triển khai theo REST API. Cung cấp 4 method chính là GET, PUT, POST, DELETE.

Hình 4.2. Hình mô tả cấu trúc request

Verb	Cách dùng	Lưu ý
GET	Chỉ lấy về dữ liệu. Không thay đổi bất kỳ dữ liệu nào với GET request	Bình thường các request dạng này đều cần đi kèm theo một token được đặt trong key “Authorization” tại Header của request để cho server biết thông tin về ai là người đang gửi request này.
POST	Tạo mới dữ liệu	
PUT	Cập nhập dữ liệu đang tồn tại trong hệ thống	Ngoài header phải có token như request ở trên (trừ một vài API đặc biệt), ở những request này cần đi kèm theo body định dạng JSON hoặc multipart (dành cho gửi file) để đính kèm thông tin giúp cho server có thể xử lý thông tin.
DELETE	Xoá dữ liệu trong hệ thống	

#### 4.3.2. Cấu trúc của một response

Response sẽ bao gồm 2 thành phần chủ chốt:

- **STATUS CODE:** đây là mã của response trả về, nhằm client có thể định dạng xem request vừa rồi có thành công hay không một cách nhanh chóng. Sau đây là những code sử dụng

Mã	Tên	Mô tả
200	OK	Thông báo request đã được xử lý thành công, response sử dụng được.
400	Bad request	Thông báo request được gửi đến là không hợp lý và sẽ không xử lý.
401	Unauthorized	Thông báo request chưa được đính kèm thông tin người gửi request cần thiết (ví dụ: chưa login)
403	Forbidden	Thông báo request này bị cấm và không được gửi.
404	Not Found	Thông báo request của người dùng không thể tìm thấy.
406	Not Acceptable	Thông báo request của người dùng gửi thông tin không hợp lý và server sẽ không chấp nhận
408	Request timeout	Thông báo request không xử lý được vì mất quá lâu
500	Interval Server Error	Thông báo lỗi của server, không thể xử lý request

- RESPONSE: response được trả về sẽ đi kèm 2 trường chính: “message”, “data”, gồm có message là để mô tả chung về response và data là dữ liệu được trả về.

#### 4.4. Các API của hệ thống

##### 4.4.1. API cho chức năng xác thực tài khoản

Phương thức	Đường dẫn	Mục tiêu
POST	http://localhost:3000/api/auth/signup	Tạo mới tài khoản Tham số bắt buộc: <ul style="list-style-type: none"> <li>● Username</li> <li>● Email</li> <li>● Password</li> </ul>
POST	http://localhost:3000/api/auth/signin	Đăng nhập Tham số bắt buộc: <ul style="list-style-type: none"> <li>● username</li> <li>● password</li> </ul>
PUT	http://localhost:3000/api/auth/forgot	Quên lại mật khẩu Tham số bắt buộc: <ul style="list-style-type: none"> <li>● email</li> <li>● new password</li> </ul>

##### 4.4.2. API cho quản lý câu hỏi

Phương thức	Đường dẫn	Mục tiêu
GET	http://localhost:3000/questions/	Lấy toàn bộ câu hỏi có trên hệ thống Tham số truy vấn có thể có: <ul style="list-style-type: none"> <li>● Page (int): trang bao nhiêu</li> <li>● Size (int): mỗi trang có bao nhiêu thực thể</li> </ul>
POST	http://localhost:3000/questions/	Tạo mới câu hỏi Tham số bắt buộc: <ul style="list-style-type: none"> <li>● Tiêu đề</li> </ul>

		<ul style="list-style-type: none"> <li>Nội dung</li> </ul>
GET	http://localhost:3000/questions/:id	Lấy thông tin câu hỏi theo id
PUT	http://localhost:3000/questions/:id	Chỉnh sửa câu hỏi có trên hệ thống Tham số bắt buộc: <ul style="list-style-type: none"> <li>id: id của câu hỏi</li> </ul> Tham số có thể có: <ul style="list-style-type: none"> <li>Tiêu đề</li> <li>Nội dung</li> <li>Chủ đề</li> <li>Trạng thái câu hỏi</li> </ul>
DELETE	http://localhost:3000/questions/:id	Xóa câu hỏi theo id có trên hệ thống Tham số bắt buộc: <ul style="list-style-type: none"> <li>id: id của câu hỏi</li> </ul>

#### 4.4.3. API cho quản lý câu trả lời

Phương thức	Đường dẫn	Mục tiêu
POST	http://localhost:3000/api/questions/:questionid/answers/	Tạo mới câu trả lời trên hệ thống Tham số bắt buộc: <ul style="list-style-type: none"> <li><u>questionid</u>: id của câu hỏi</li> <li>Id của người viết câu trả lời</li> </ul>
GET	http://localhost:3000/api/question/{id}/answers	Lấy thông tin danh sách câu trả lời của 1 câu hỏi Tham số truy vấn có thể có: <ul style="list-style-type: none"> <li>id của câu hỏi</li> <li>Page (int): trang bao nhiêu</li> <li>Size (int): mỗi trang có bao nhiêu thực thể</li> </ul>
DELETE	http://localhost:3000/api/answers/:id	Xóa câu trả lời theo id Tham số truy vấn bắt buộc: <ul style="list-style-type: none"> <li>id của câu trả lời</li> </ul>

#### 4.4.4. API cho quản lý chủ đề

Phương thức	Đường dẫn	Mục tiêu
GET	http://localhost:3000/api/tags	Lấy toàn bộ thông tin các tags có trong hệ thống
POST	http://localhost:3000/tags/	Tạo mới tag
GET	http://localhost:3000/tags/:id	Lấy thông tin có tag theo id
DELETE	http://localhost:3000/tags/:id	Xóa thông tin của tag theo id

#### 4.4.4. API cho quản lý người dùng

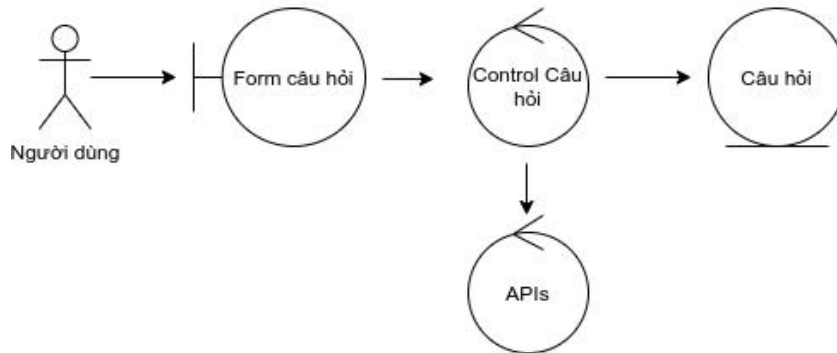
Phương thức	Đường dẫn	Mục tiêu
GET	http://localhost:3000/api/users	Hiển thị danh sách người dùng Tham số truy vấn có thể có: <ul style="list-style-type: none"><li>● Page (int): trang bao nhiêu</li><li>● Size (int): mỗi trang có bao nhiêu thực thể</li></ul>
GET	http://localhost:3000/api/users/:id	Hiển thị người dùng Tham gia bắt buộc: <ul style="list-style-type: none"><li>● Id: id người dùng</li></ul>
POST	http://localhost:3000/api/users	Thêm người dùng Tham số bắt buộc: <ul style="list-style-type: none"><li>● Username</li><li>● Email</li><li>● Password</li></ul>
PUT	http://localhost:3000/api/users/id	Cập nhật người dùng Tham gia bắt buộc: <ul style="list-style-type: none"><li>● Id: id người dùng</li></ul>
DELETE	http://localhost:3000/api/users/id	Xóa người dùng Tham gia bắt buộc: id người dùng

**Kết luận chương:** Qua chương 4, em đã đưa ra sơ đồ tổng quan Use Case các tác nhân cũng như chức năng của Forum, liệt kê danh sách một số các API được sử dụng trong hệ thống. Các API này sẽ xử lý luồng request, response mà bên phía Client gọi yêu cầu. Bảng cơ sở dữ liệu quan hệ cộng thêm mô tả bên phía data của Forum.

# CHƯƠNG 5. THIẾT KẾ CHỨC NĂNG

## 5.1. Quản lý câu hỏi

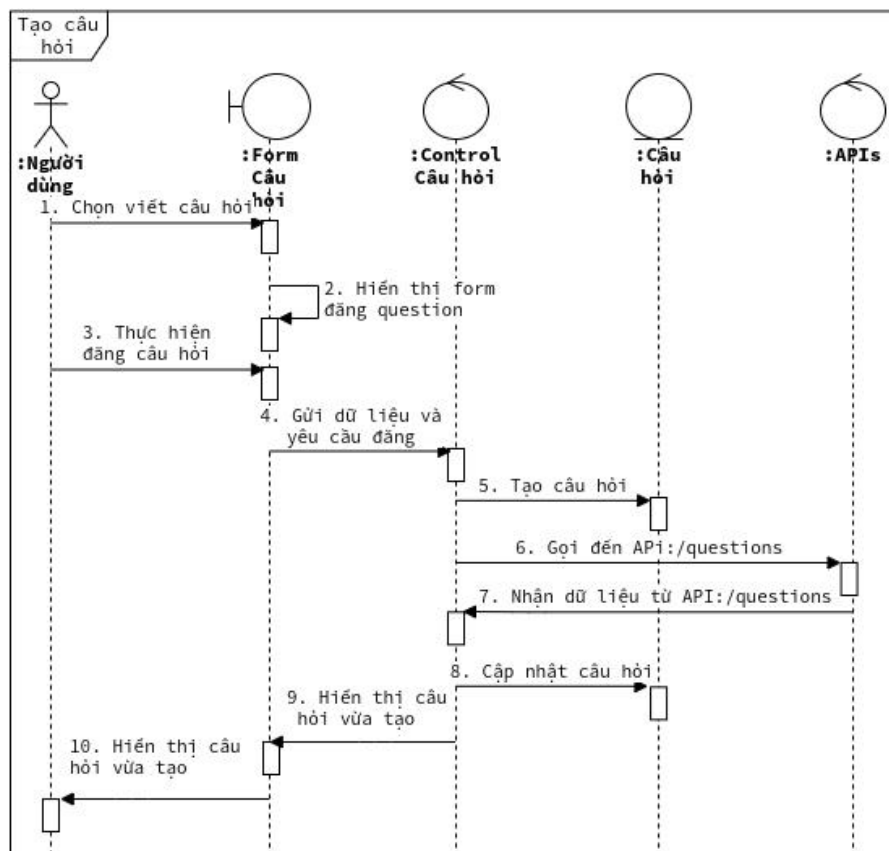
### 5.1.1. Sơ đồ lớp



Hình 5.1. Sơ đồ lớp câu hỏi

### 5.1.2. Sơ đồ trình tự

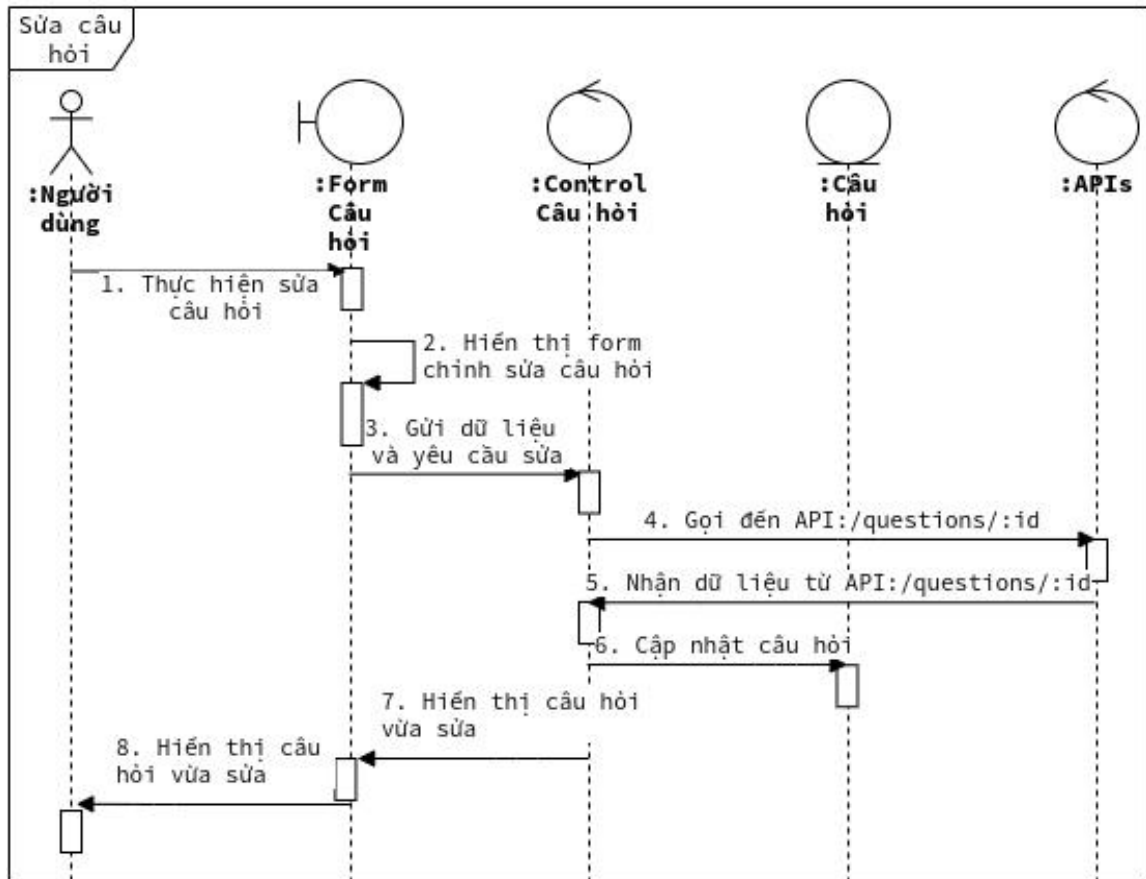
#### 5.1.2.1. Tạo câu hỏi



Hình 5.2. Tạo câu hỏi

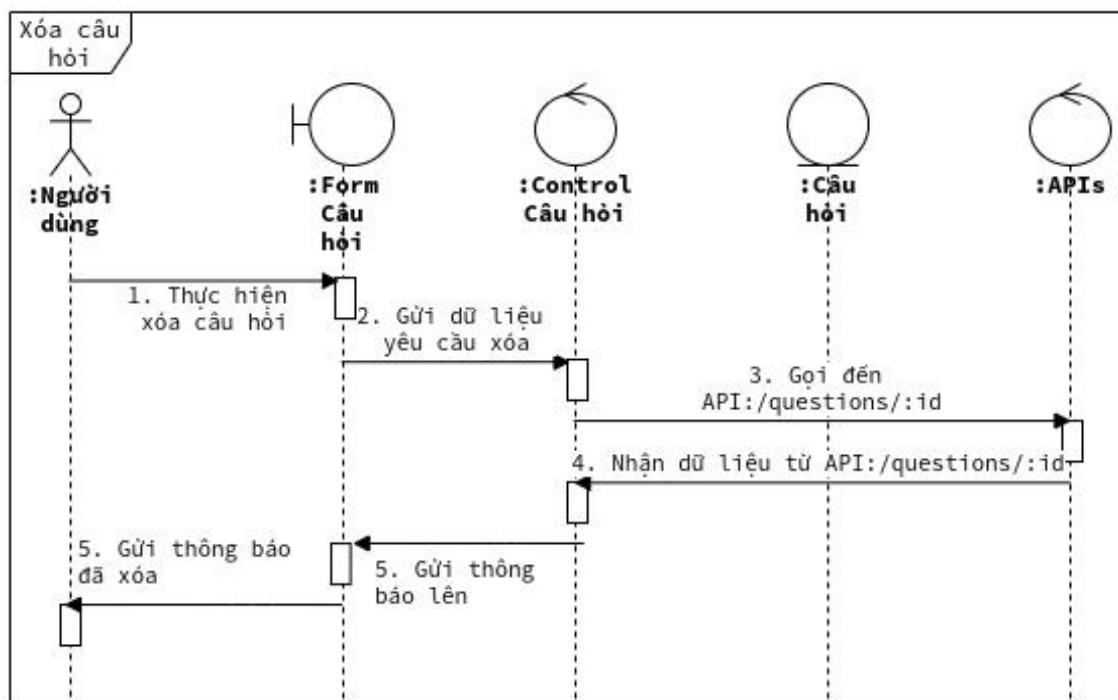


### 5.1.2.2. Sửa câu hỏi



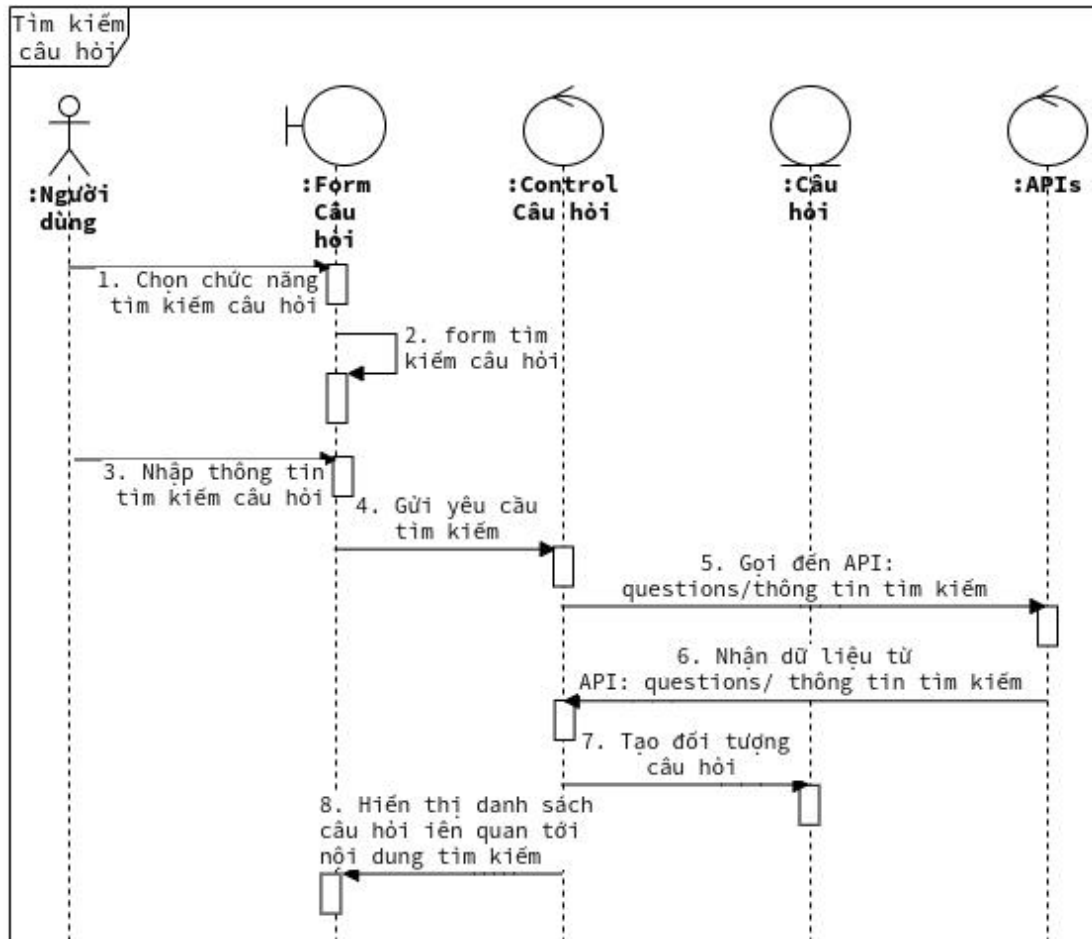
Hình 5.3. Hình mô tả sửa câu hỏi

### 5.1.2.3. Xóa câu hỏi



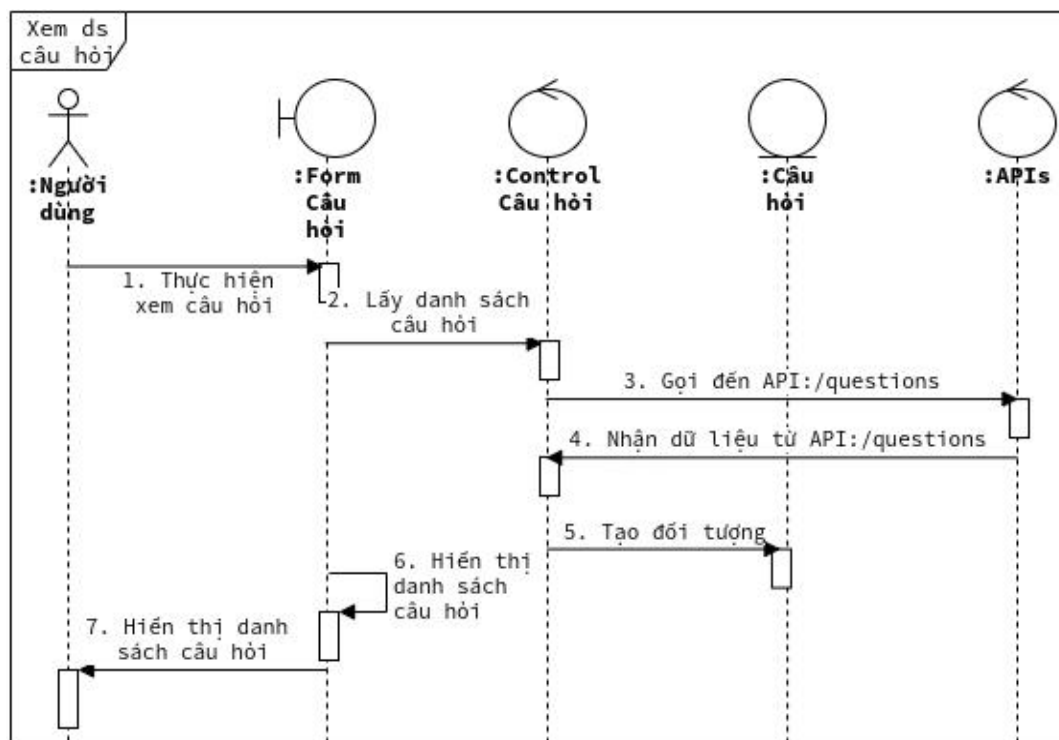
Hình 5.4. Hình mô tả chức năng xóa câu hỏi

#### 5.2.2.4. Tìm kiếm câu hỏi



Hình 5.5. Mô tả chức năng tìm kiếm câu hỏi

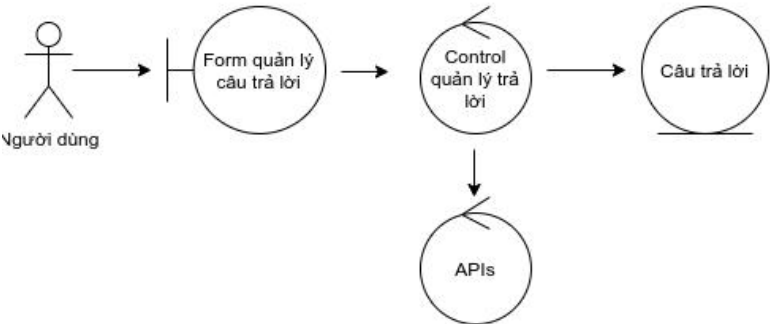
#### 5.1.2.5. Xem danh sách câu hỏi



Hình 5.6. Xem danh sách câu hỏi

5.2. Quản lý câu trả lời

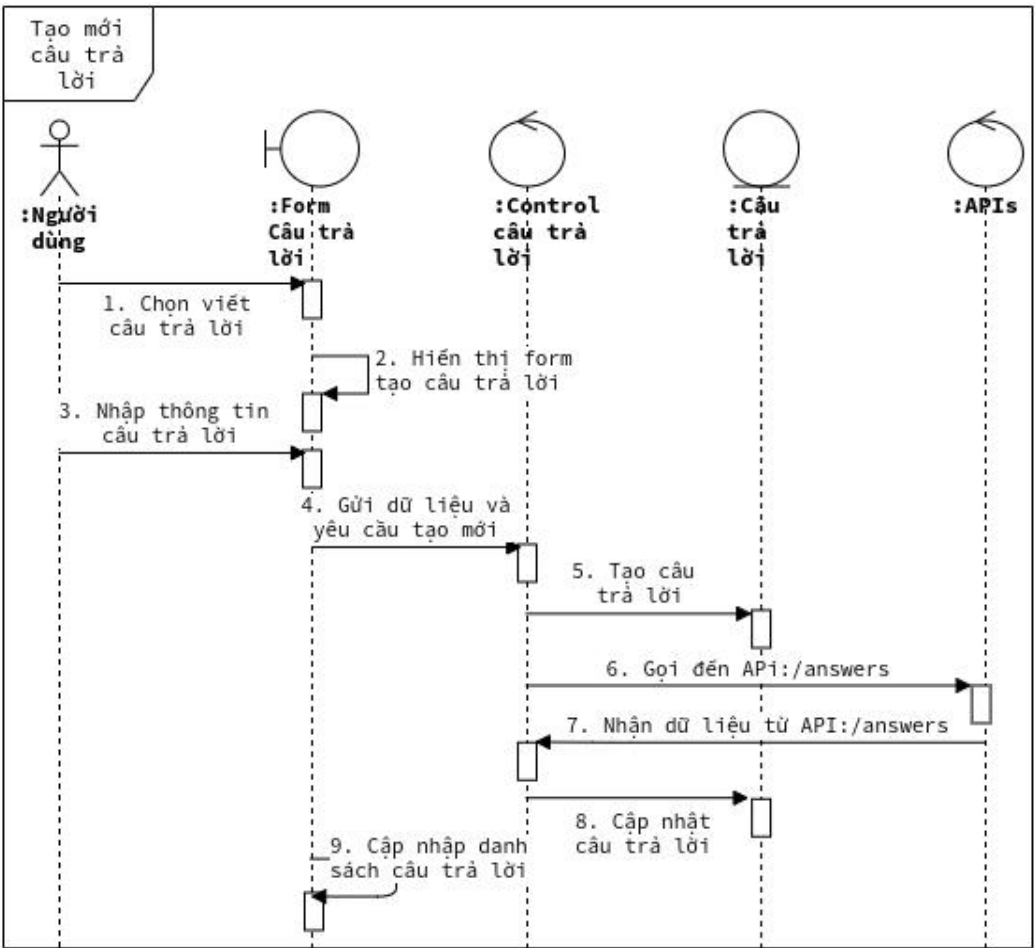
5.2.1. Sơ đồ lớp



Hình 5.7. Mô tả sơ đồ lớp

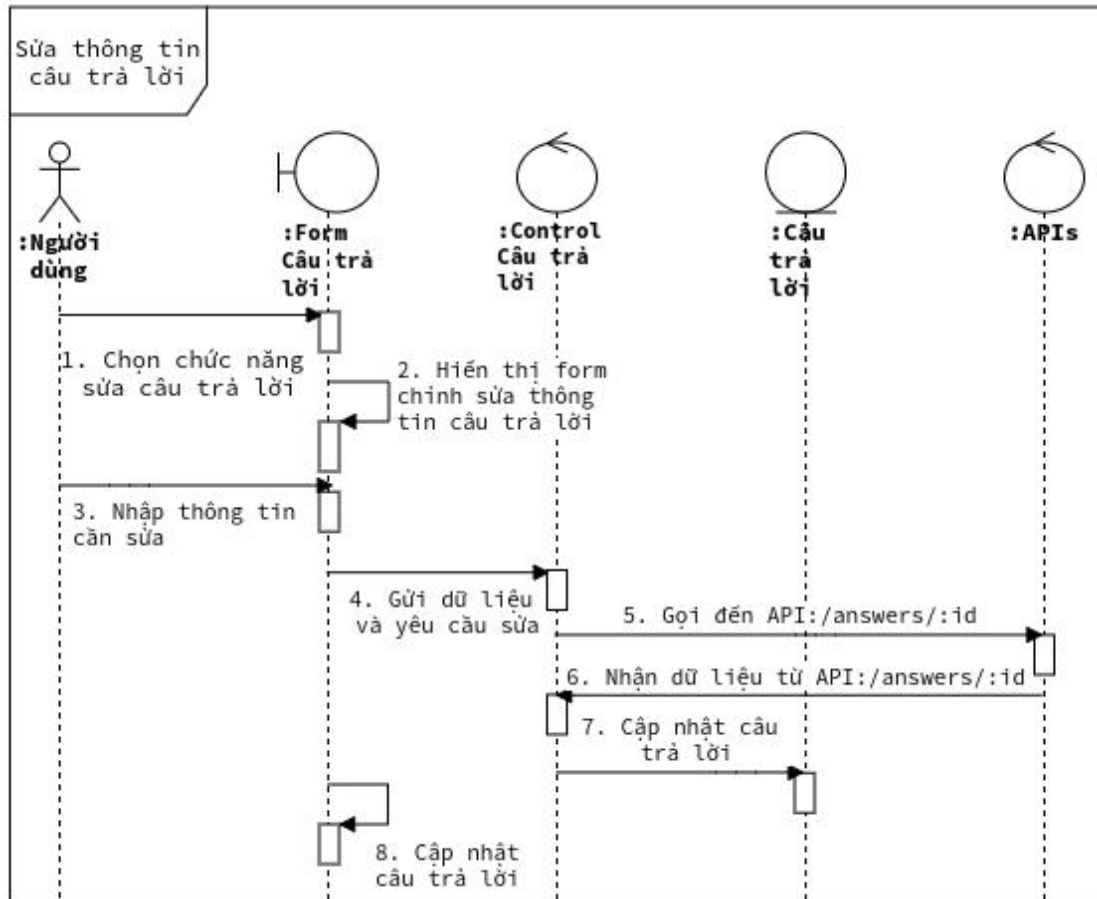
5.2.2. Sơ đồ trình tự

5.2.2.1. Tạo câu trả lời



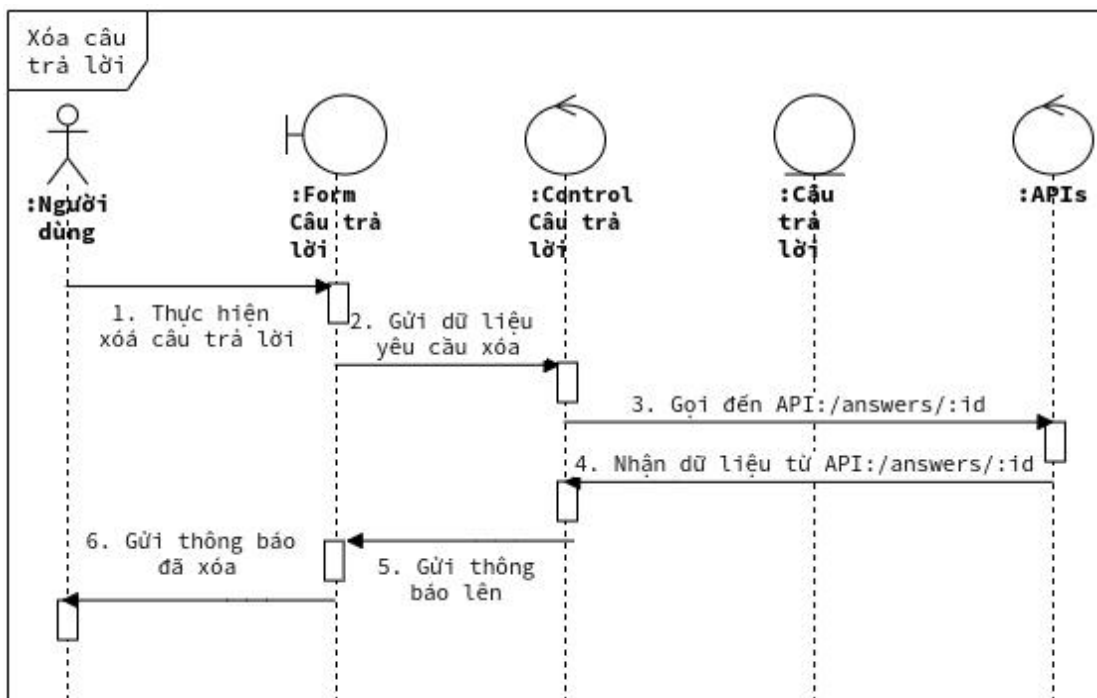
Hình 5.8. Mô tả chức năng tạo câu trả lời

### 5.2.2.2. Sửa câu trả lời



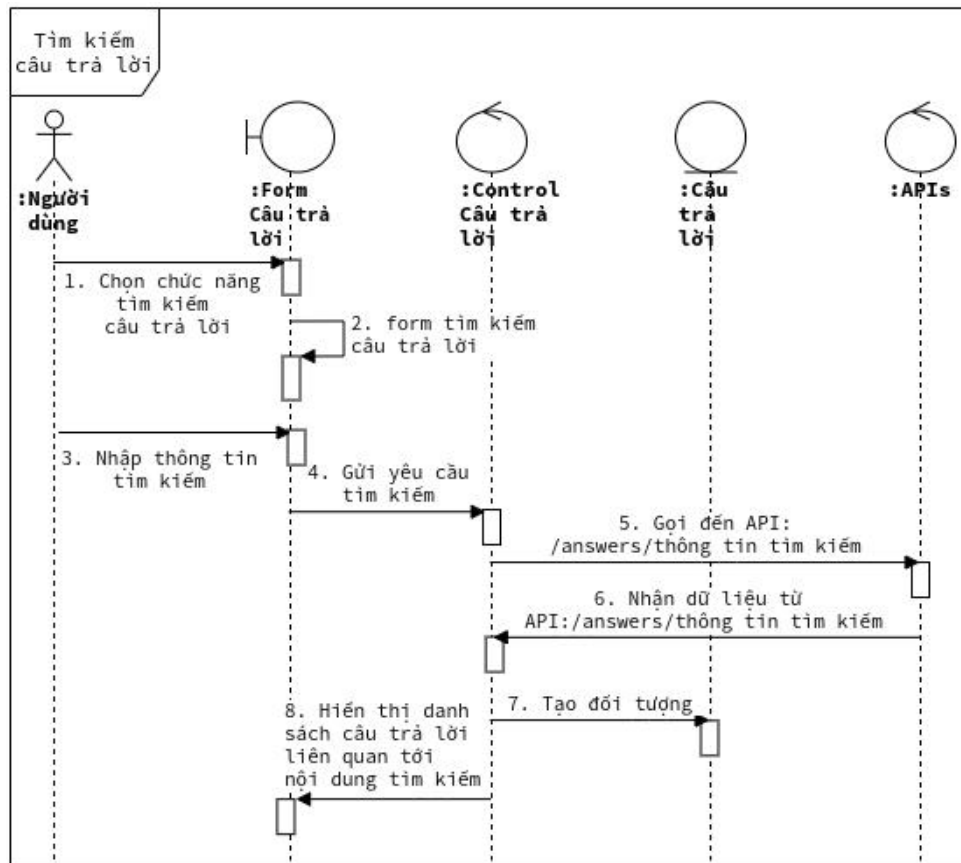
Hình 5.9. Sửa câu trả lời

### 5.2.2.3. Xóa câu trả lời



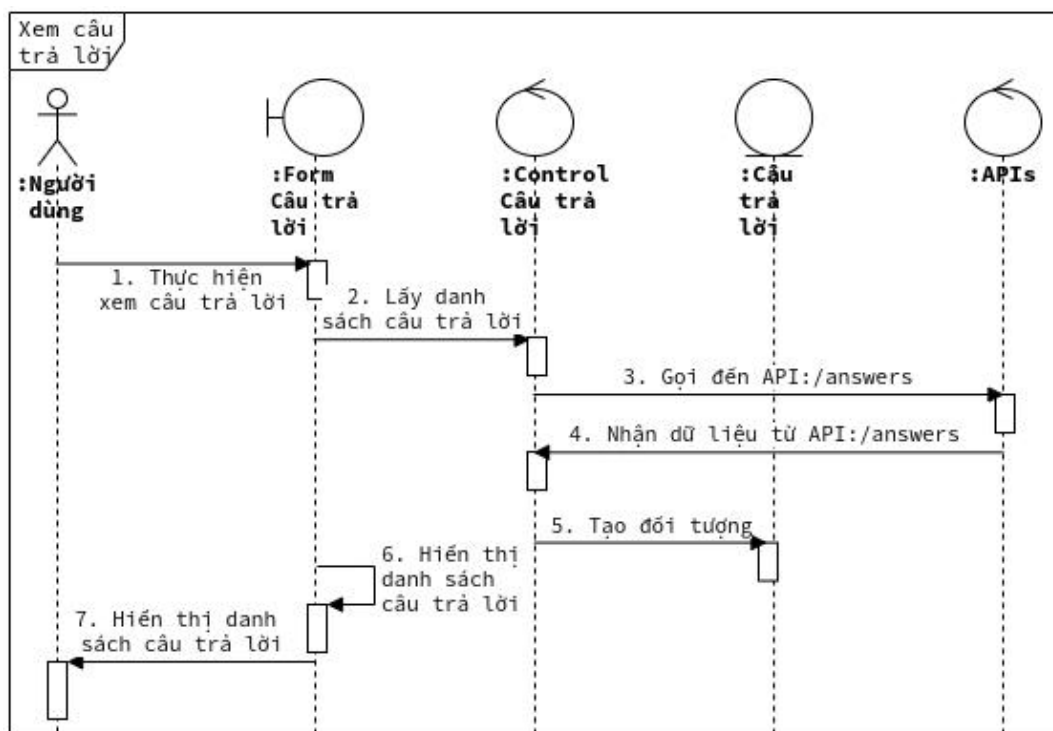
Hình 5.10. Mô tả chức năng xóa câu trả lời

#### 5.2.2.4. Tìm kiếm câu trả lời



Hình 5.11. Mô tả chức năng tìm kiếm câu trả lời

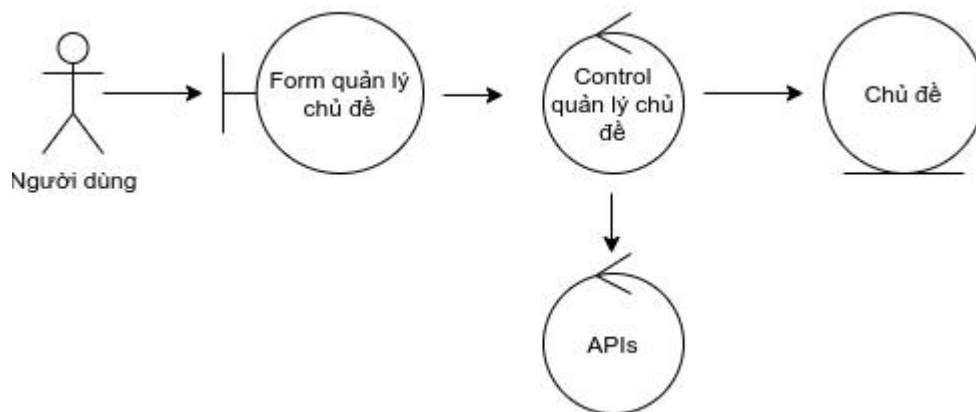
#### 5.2.2.5. Xem danh sách câu trả lời



Hình 5.12. Mô tả chức năng xóa câu trả lời

### 5.3. Quản lý chủ đề

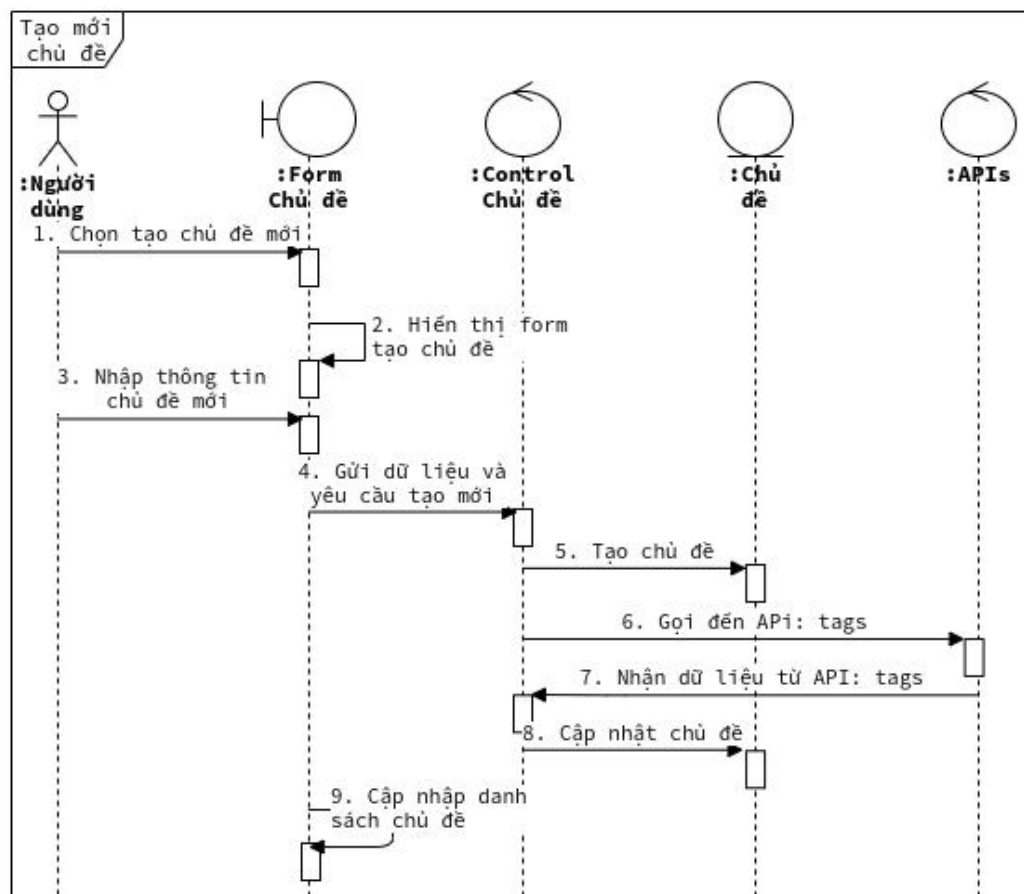
#### 5.3.1. Sơ đồ lớp



Hình 5.13. Mô tả chức năng quản lý chủ đề

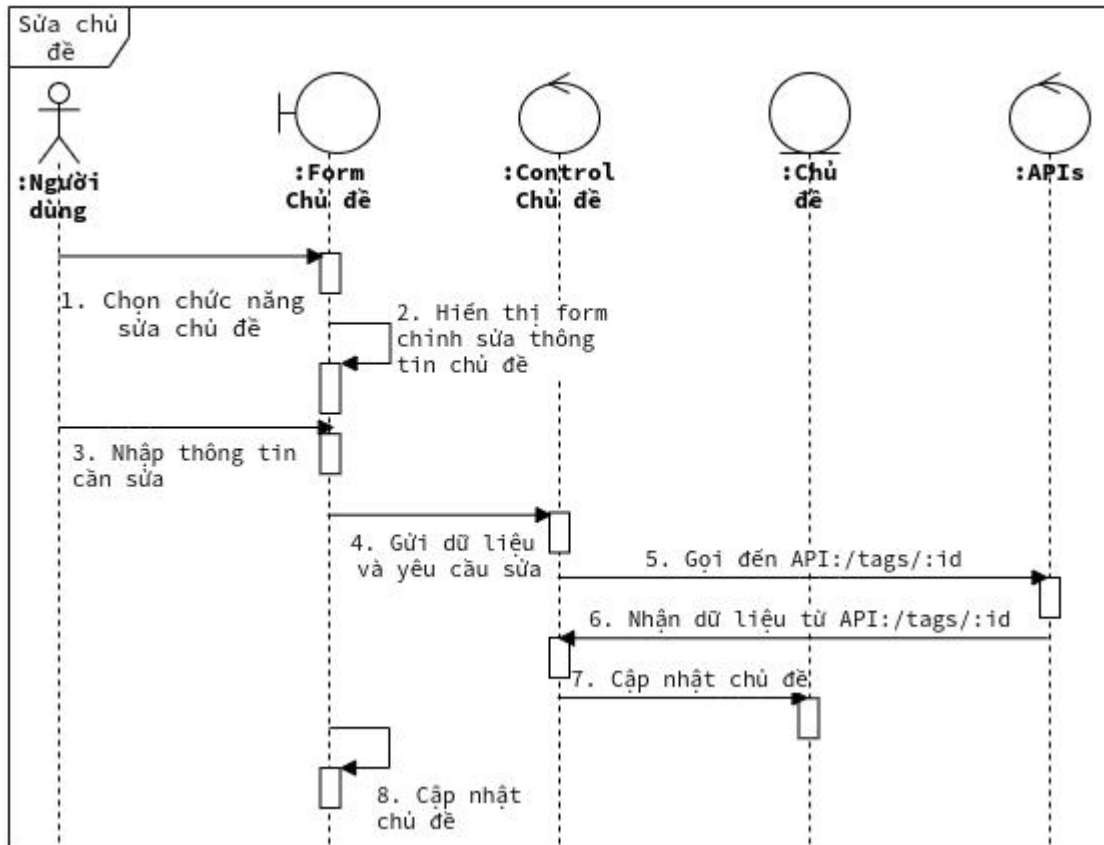
#### 5.3.2. Sơ đồ trình tự

##### 5.3.2.1. Tạo chủ đề



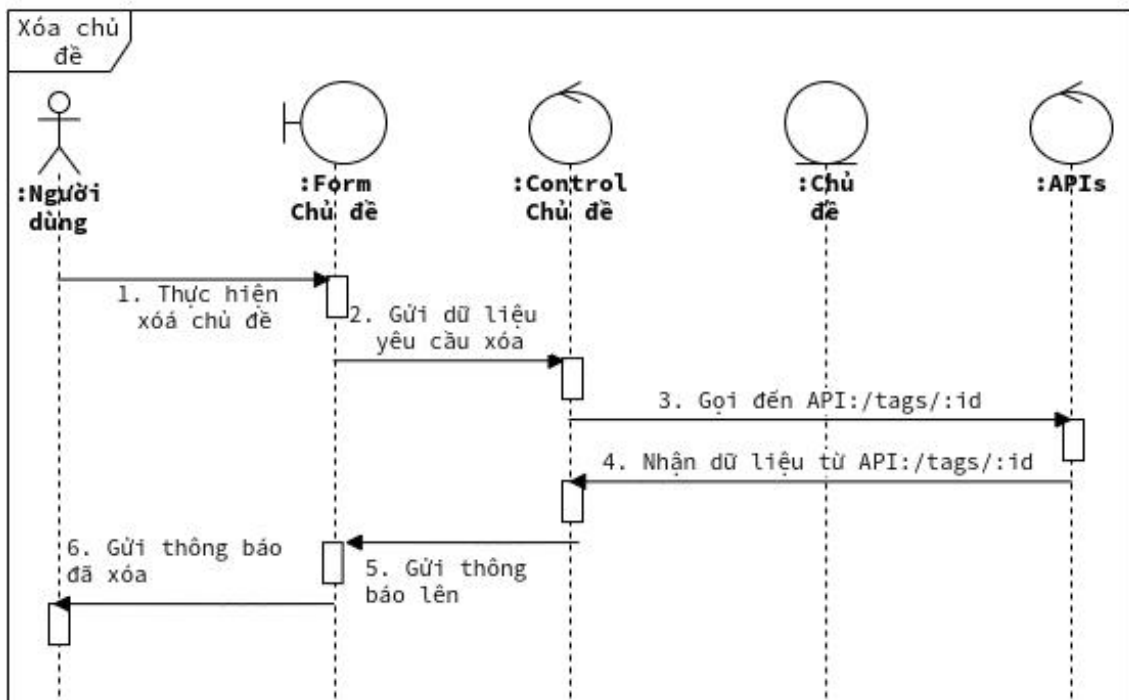
Hình 5.14. Mô tả chức năng tạo chủ đề

### 5.3.2.2. Sửa chủ đề



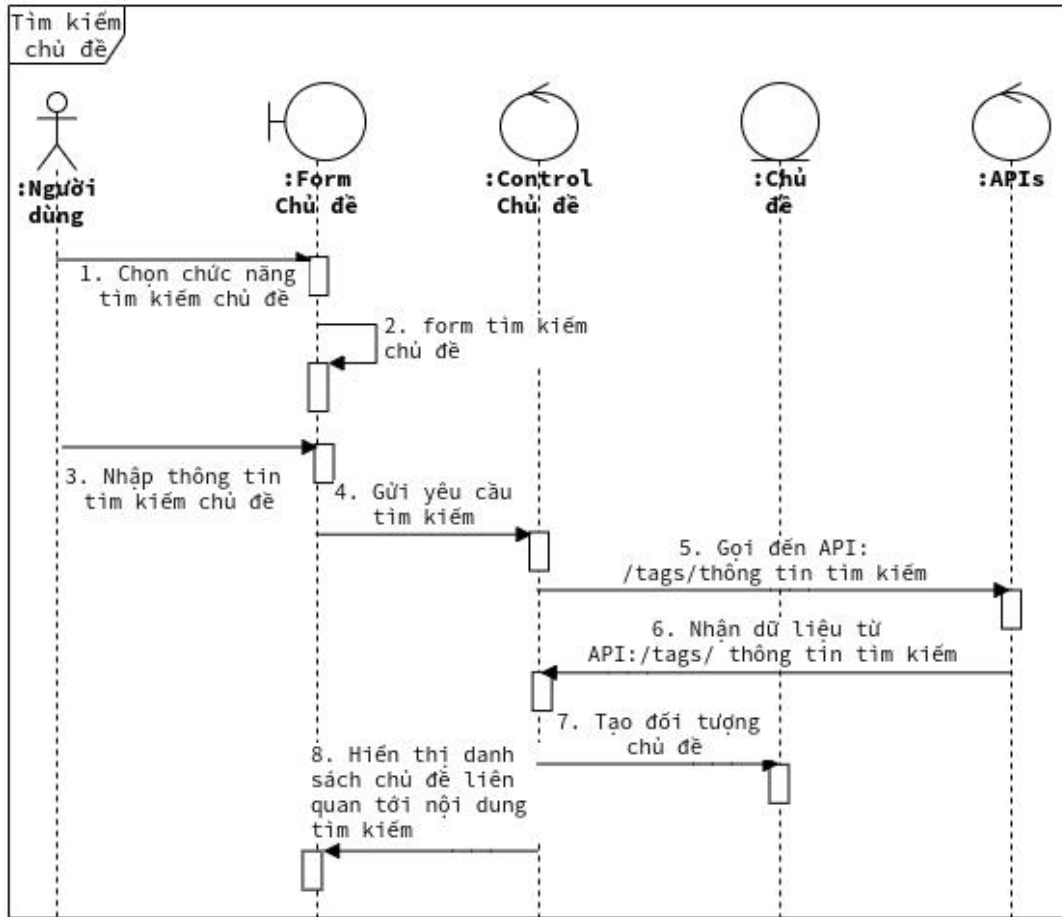
Hình 5.15. Mô tả chức năng sửa chủ đề

### 5.3.2.3. Xóa chủ đề



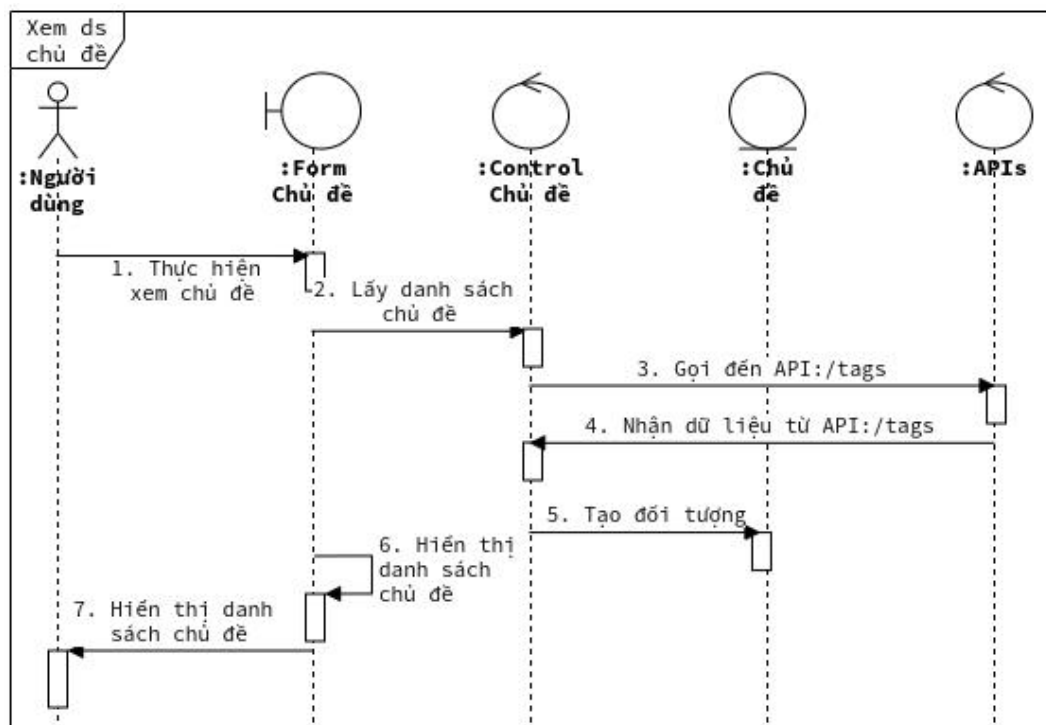
Hình 5.16. Mô tả chức năng xóa chủ đề

### 5.3.2.4. Tìm kiếm chủ đề



Hình 5.17. Mô tả chức năng tìm kiếm chủ đề

### 5.3.2.5. Xem danh sách chủ đề

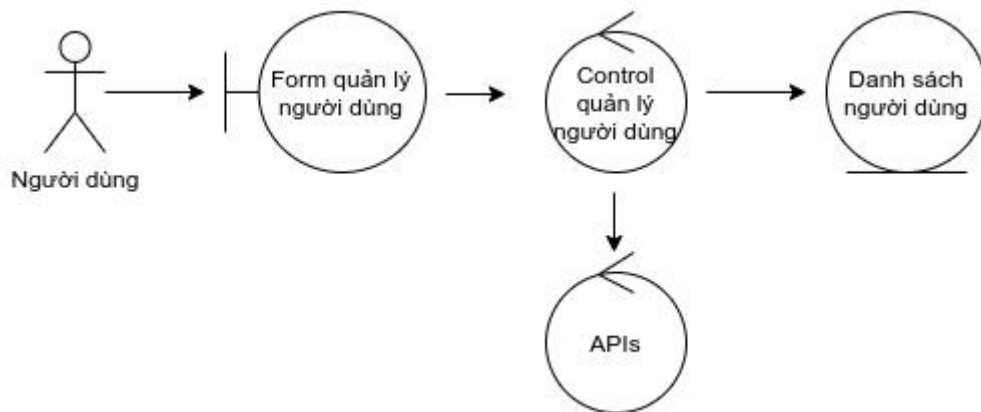


Hình 5.18. Mô tả chức năng xem danh sách chủ đề



## 5.4. Quản lý người dùng

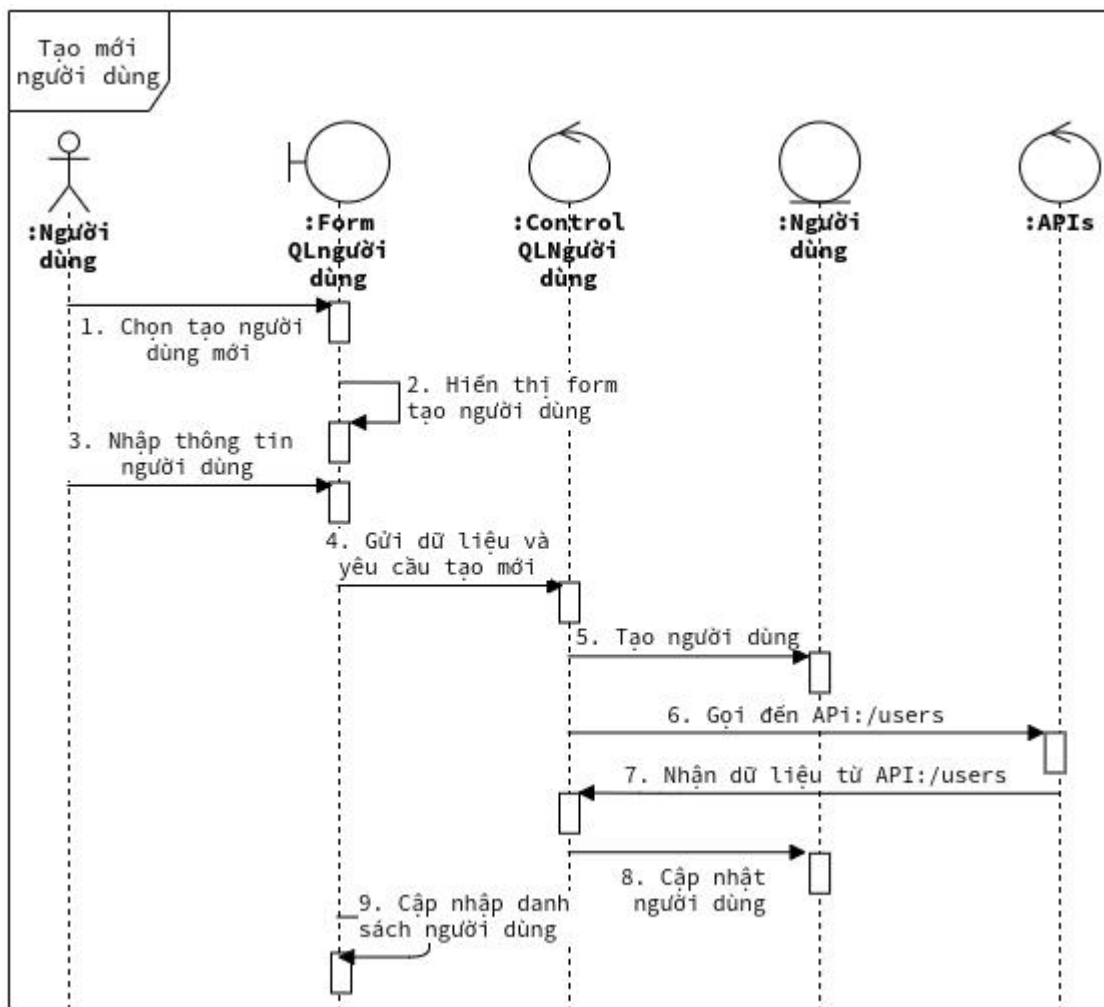
### 5.4.1. Sơ đồ lớp



Hình 5.19. Mô tả chức năng tạo chủ đề

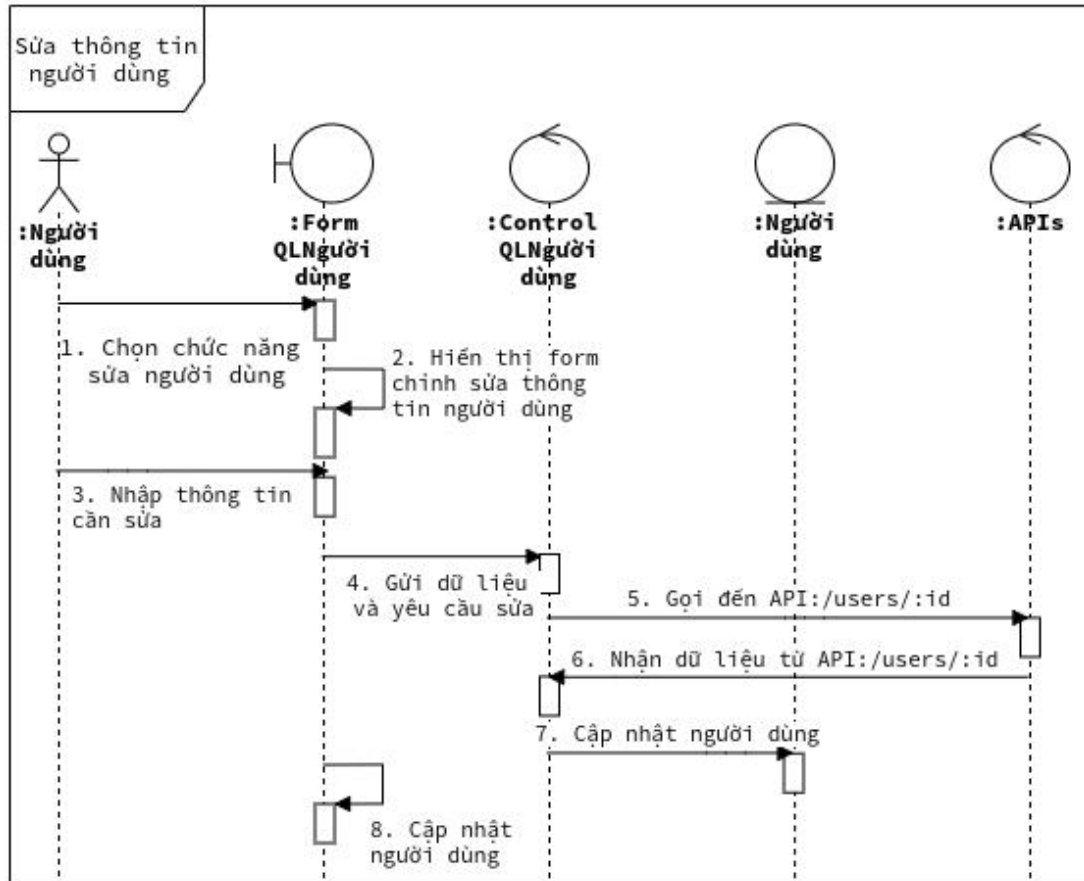
### 5.4.2. Sơ đồ trình tự

#### 5.4.2.1. Tạo người dùng



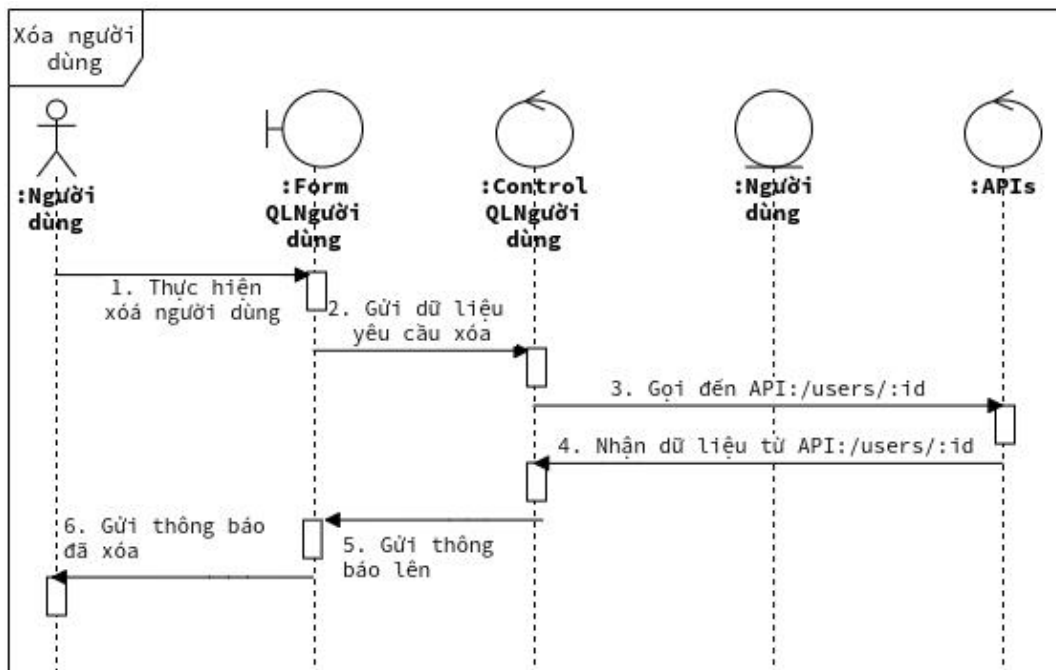
Hình 5.20. Mô tả chức năng tạo chủ đề

#### 5.4.2.2. Sửa người dùng



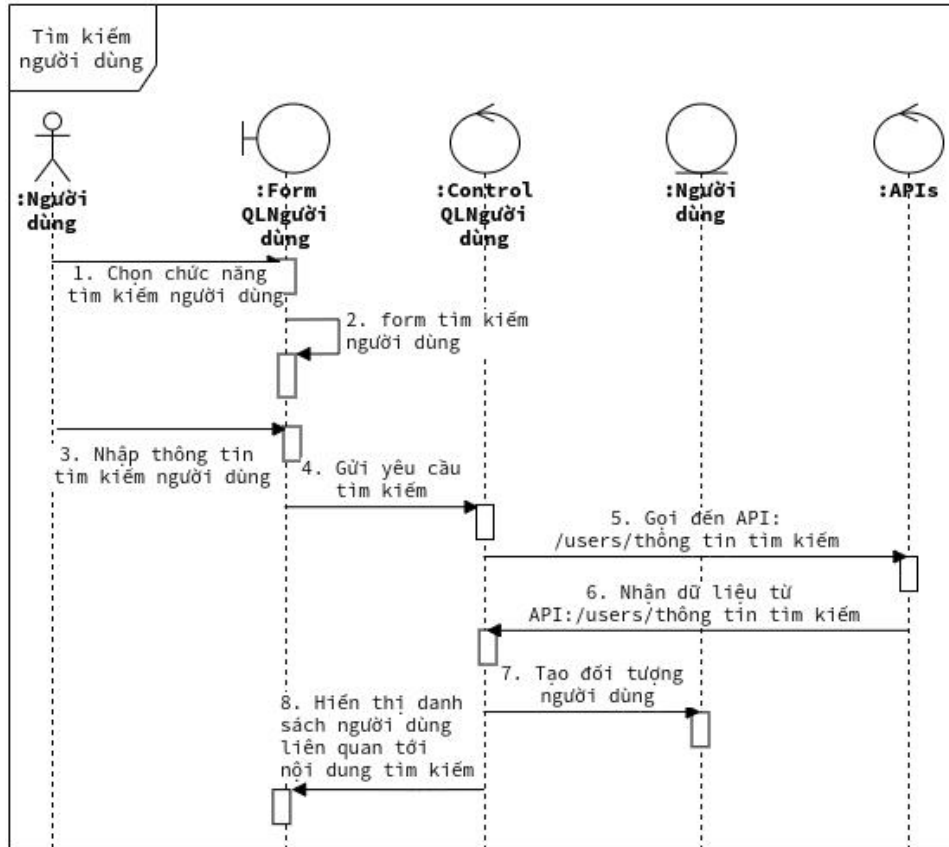
Hình 5.21. Mô tả chức năng sửa thông tin người dùng

#### 5.4.2.3. Xóa người dùng



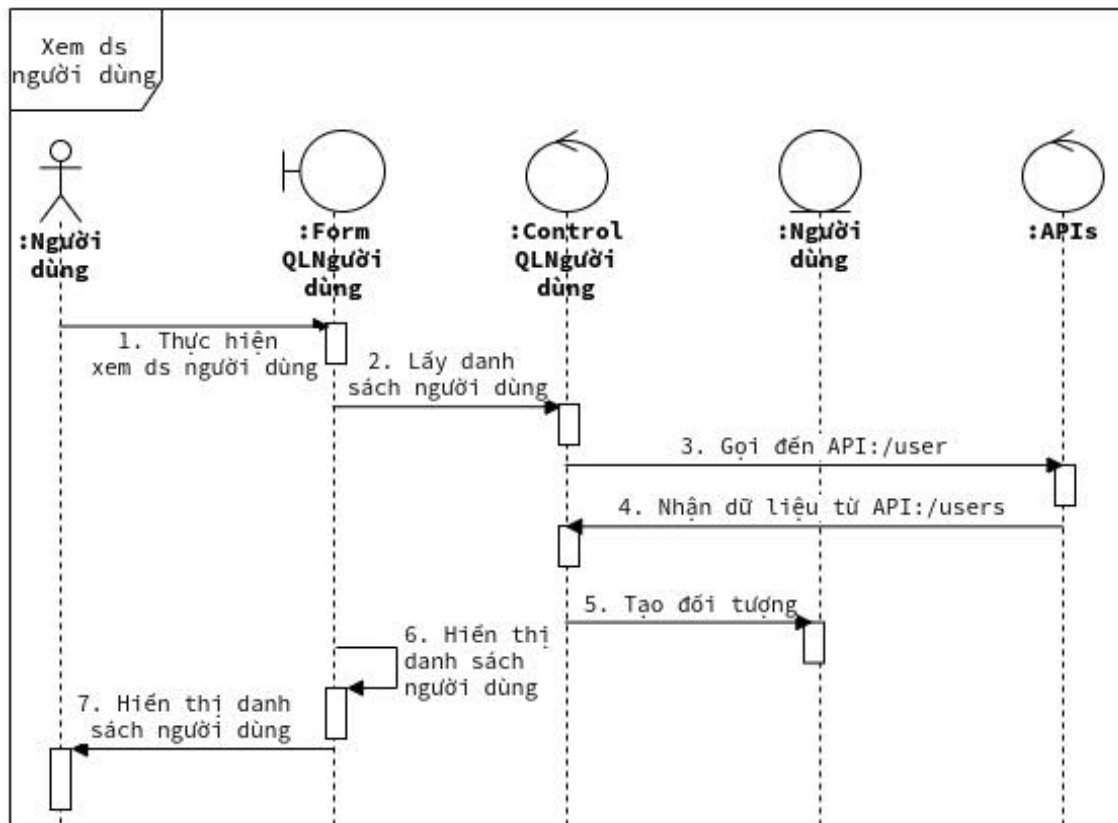
Hình 5.22. Mô tả chức năng xóa người dùng

#### 5.4.2.4. Tìm kiếm người dùng



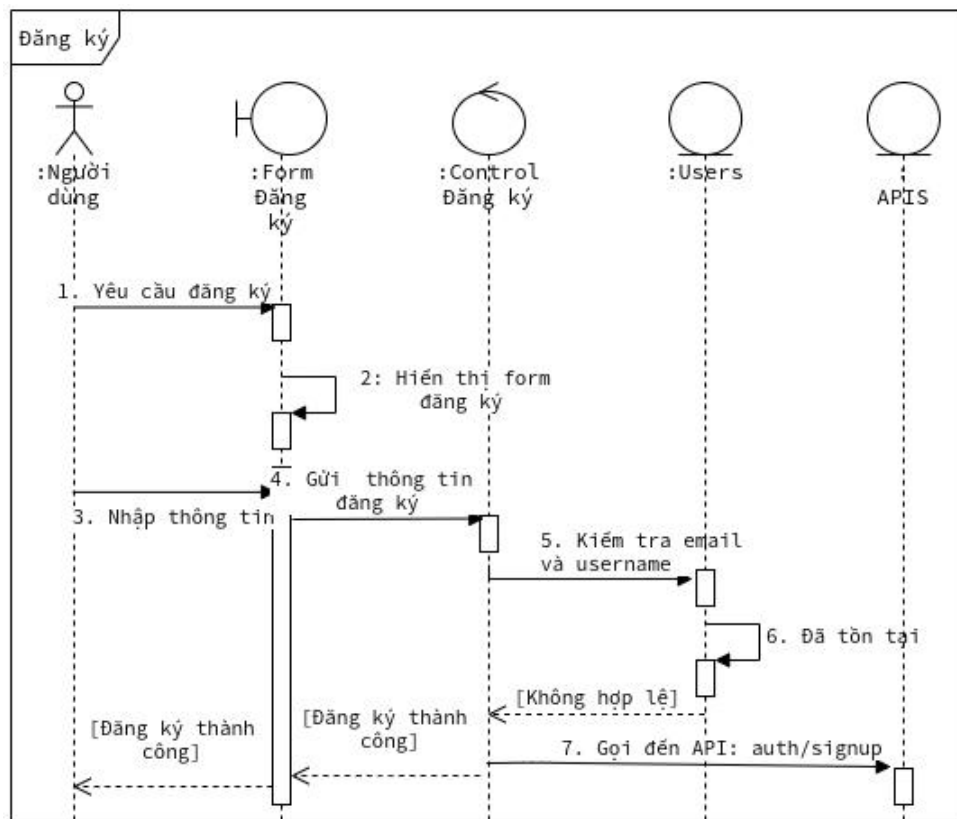
Hình 5.23. Mô tả chức năng tìm kiếm người dùng

#### 5.4.2.5. Xem danh sách người dùng



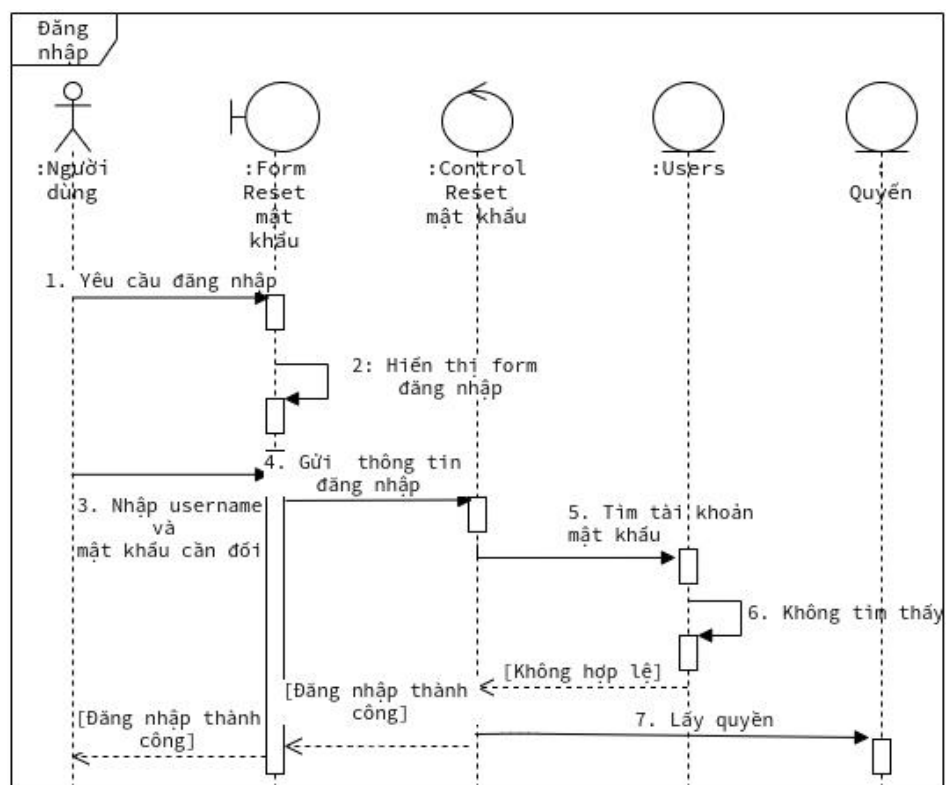
Hình 5.24. Mô tả chức năng danh sách người dùng

## 5.5. Đăng ký



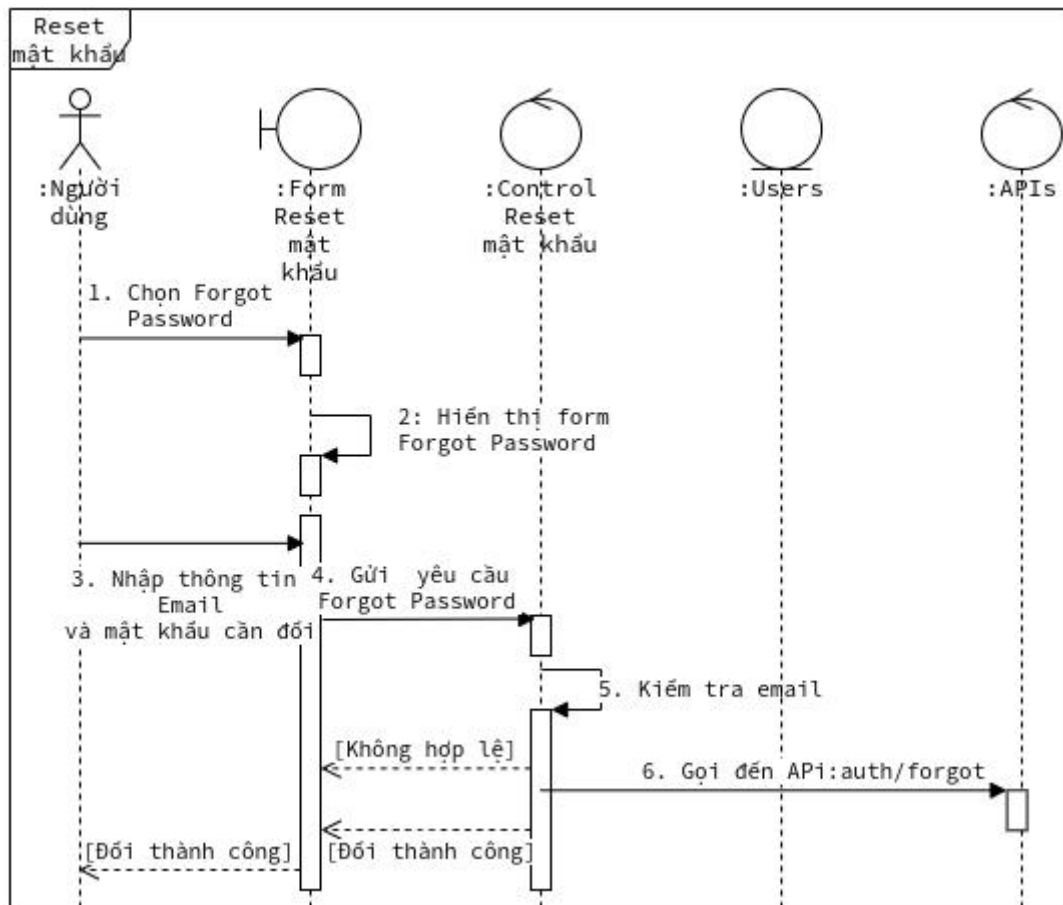
Hình 5.25. Mô tả chức năng đăng ký

## 5.6. Đăng nhập



Hình 5.26. Mô tả chức năng đăng nhập

## 5.7. Reset Mật khẩu



Hình 5.27. Mô tả chức năng Reset mật khẩu

**Kết luận chương:** Qua chương 5, em đã giới thiệu các thiết kế UML cũng như giới thiệu đặc tả chi tiết các chức năng chính của hệ thống.

## CHƯƠNG 6. KIỂM THỬ

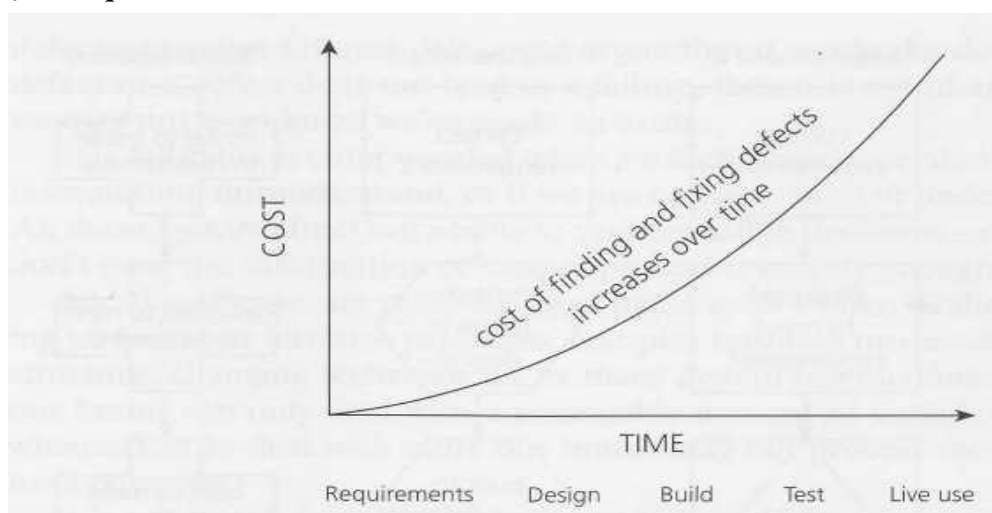
Trong phần này, em sẽ giới thiệu về tổng quát các khái niệm liên quan đến kiểm thử trong hệ thống, các loại kiểm thử trong hệ thống và một vài Test Case tiêu biểu.

### 6.1. Giới thiệu chung

#### 6.1.1. Kiểm thử phần mềm

- Kiểm thử phần mềm (software testing) là hoạt động nhằm tìm kiếm, phát hiện các lỗi của phần mềm.
- Kiểm thử phần mềm là một tiến trình hay một tập hợp các tiến trình được thiết kế để đảm bảo chương trình máy tính thực hiện theo đúng yêu cầu của khách hàng.
- Kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm lỗi (theo The Art of Software Testing).

#### 6.1.2. Tạo sao phải kiểm thử



Hình 6.1. Mô tả kiểm thử

Kiểm thử là quan trọng bởi vì các lỗi của phần mềm thường có chi phí sửa đắt đỏ hoặc thậm chí gây nguy hiểm.

#### 6.1.3. Ý nghĩa của kiểm thử

- Kiểm thử phần mềm không chỉ cần tìm lỗi phần mềm, mà còn là quá trình kiểm tra và xác minh một phần mềm đã đáp ứng được yêu cầu và mong đợi của khách hàng.
- Chi phí kiểm thử chiếm: 40% tổng công sức phát triển,  $\geq 30\%$  tổng thời gian phát triển. Với các phần mềm quan trọng chi phí có thể gấp từ 3 đến 5 lần tổng các chi phí khác cộng lại. Do đó, kiểm thử tốt sẽ:
- Giảm chi phí điều chỉnh và thời gian phát triển.
- Tăng độ tin cậy của phần mềm.

#### 6.1.4. Mục tiêu kiểm thử

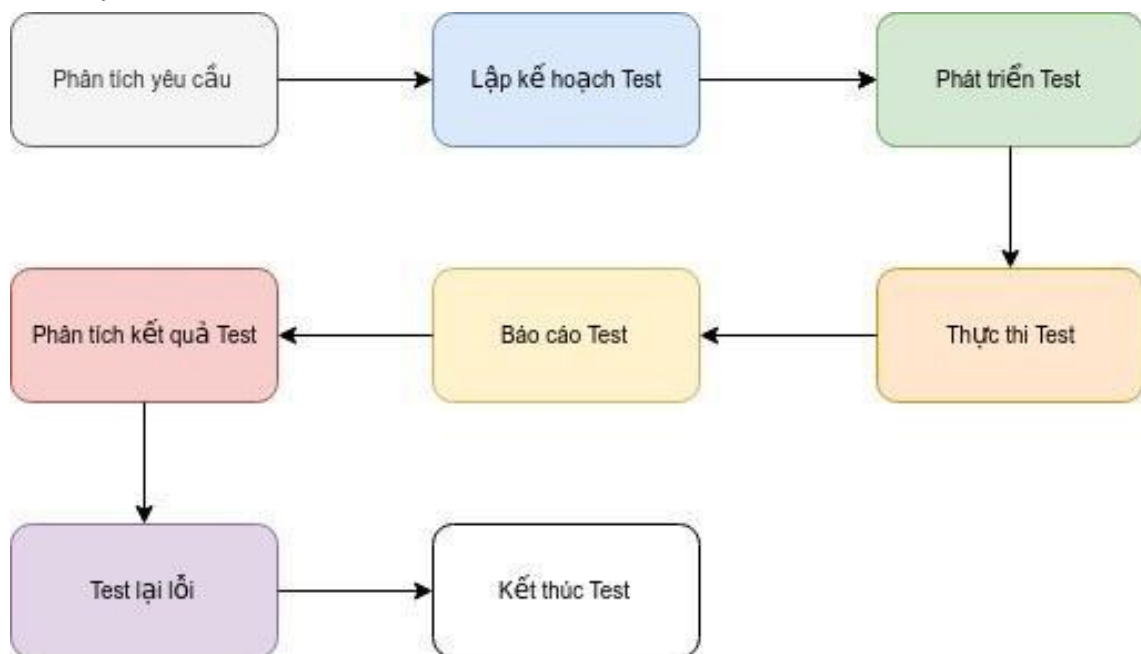
- Phát hiện càng nhiều lỗi càng tốt trong thời gian xác định trước.
- Và phải làm thế nào nghĩ ra cách ngăn chặn lỗi phát sinh.

- Chứng minh rằng sản phẩm phần mềm phù hợp với các đặc tả của nó.
- Chứng tỏ được rằng các testcase và các báo cáo tạo ra đúng và hữu dụng.

#### 6.1.5. Nguyên tắc kiểm thử

- Nguyên tắc kiểm thử cũng giống như nguyên tắc sử dụng thuốc trừ sâu, khi chúng ta sử dụng một loại thuốc trừ sâu mãi thì sẽ bị nhờn thuốc nên phải thay đổi loại thuốc khác. Trong kiểm thử phần mềm, khi dùng đi dùng lại một bộ kịch bản kiểm thử thì sẽ đến lúc không thể tìm ra lỗi mới nữa. Chính vì vậy các bộ kịch bản kiểm thử phải được thường xuyên xem xét và cập nhật, phù hợp với từng thành phần khác nhau của phần mềm, mang lại khả năng tìm thấy lỗi lớn nhất.
- Nguyên tắc kiểm thử:
  - Kiểm thử để chỉ ra sự hiện diện của lỗi
  - Kiểm thử toàn diện, toàn bộ là không thể
  - Kiểm thử càng sớm càng tốt
  - Phân nhóm lỗi để xác định một số module tập trung lỗi nhiều nhất
  - Nghịch lý thuốc trừ sâu
  - Kiểm thử được thực hiện khác nhau trong những bối cảnh khác nhau
  - Suy nghĩ "Không có lỗi" là một sai lầm

#### 6.1.6. Quy trình kiểm thử



Hình 6.2. Sơ đồ quy trình kiểm thử

- Phân tích yêu cầu: Xác định phạm vi test.
- Lập kế hoạch Test: Chiến lược test (test strategy), test plan.
- Phát triển Test: Viết test procedure, test scenario, test case, test data và test script.
- Thực thi Test: Tester thực thi phần mềm dựa trên test plan và test case.
- Báo cáo Test: Tester điền kết quả test vào test case và tạo báo cáo kết quả test.

- Phân tích kết quả Test: Quyết định lỗi nào được sửa và lỗi nào sẽ không sửa.
- Test lại lỗi: Sau khi một lỗi được DEV sửa xong, chuyển hệ thống cho tester test lại.
- Kết thúc Test: Khi test đã đáp ứng được điều kiện dừng. Từ đó rút ra các bài học kinh nghiệm.

## **6.2. Kiểm thử trong hệ thống**

### **6.2.1. Các loại kiểm thử trong hệ thống**

#### **6.2.1.1. Kiểm thử đơn vị (Unit Test)**

- Định nghĩa:
- Unit Test là một mức kiểm thử phần mềm với mục đích để xác nhận từng unit của phần mềm được phát triển đúng như được thiết kế. Các hàm (Function), thủ tục (Procedure), lớp (Class), hoặc các phương thức (Method) đều có thể được xem là unit. Unit Test là mức test nhỏ nhất trong bất kì phần mềm nào.
- Người thực hiện và thời gian thực hiện:
- Unit Test là mức kiểm thử đầu tiên trong các mức kiểm thử phần mềm. Unit Test thường do lập trình viên thực hiện, được thực hiện càng sớm càng tốt trong giai đoạn viết code và xuyên suốt quá trình phát triển phần mềm.
- Mục đích thực hiện:
- Tăng sự đảm bảo khi có sự thay đổi code
- Phát triển nhanh hơn
- Chi phí sửa lỗi thấp hơn so với các mức kiểm thử giai đoạn
- Debug dễ

#### **6.2.1.2. Kiểm thử tích hợp (Integration Test)**

- Định nghĩa: Kiểm thử tích hợp là một mức của kiểm thử phần mềm kiểm tra một nhóm các module nhỏ liên quan đến nhau xem chúng có hoạt động đúng chức năng như trong thiết kế hay không. Kiểm thử tích hợp được thực hiện để phát hiện các lỗi về giao diện hoặc trong tương tác giữa các thành phần hoặc hệ thống tích hợp (theo ISTQB - International Software Testing Qualifications Board).
- Người thực hiện và thời gian thực hiện: Kiểm thử tích hợp có thể được thực hiện bởi developer, một test team chuyên biệt hay một nhóm chuyên developer/kiểm thử viên tích hợp bao gồm cả kiểm thử phi chức năng. Kiểm thử tích hợp được thực hiện sau Unit Test.
- Mục đích thực hiện: Kiểm tra sự tích hợp 1 nhóm các thành phần riêng lẻ có liên quan xem chúng có hoạt động đúng như mong đợi hay không.

**\* Ta có bảng tổng quát như sau:**



<b>Các mức kiểm thử</b>	<b>Kiểm thử đơn vị (Unit Test)</b>	<b>Kiểm thử tích hợp (Integration Test)</b>
Định nghĩa	Là mức kiểm thử nhỏ nhất, test từng module nhỏ trong hệ thống	Kiểm thử tích hợp một nhóm các module riêng lẻ có liên quan đến nhau
Người thực hiện	Do chính DEV viết module đó	Kiểm thử độc lập hoặc do lập trình viên
Thời gian thực hiện	Trong suốt quá trình build module đó. Phải hoàn thành trước khi chuyển sang kiểm thử tích hợp	Sau khi các module nhỏ liên quan đến nhau đã được Unit Test
Mục đích	Đảm bảo mỗi module nhỏ của phần mềm được thực hiện đúng thiết kế	Kiểm tra chức năng, độ tin cậy và hiệu suất của hệ thống khi tích hợp
Ưu điểm	Tăng sự đảm bảo khi có sự thay đổi về code. Chi phí sửa lỗi thấp, dễ dàng Debug	Phát hiện lỗi ở đơn vị thấp của sản phẩm. Thuận tiện với các dự án nhỏ
Nhược điểm	DEV ít kinh nghiệm sẽ mất khá nhiều thời gian để fix bug	Có thể bỏ qua các bug giao diện nhỏ trong quá trình tìm bug
Kỹ thuật kiểm thử	Kiểm thử hộp trắng	Kiểm thử hộp trắng, đen và xám

### 6.2.2. Kiểm thử trên Web Client

- Kiểm thử tích hợp chức năng Quản lý người dùng:

STT	Mục tiêu kiểm thử	Mô tả thao tác kiểm thử	Kết quả chờ đợi	Kết quả thực tế
<b>TC_02.1</b>	<b>Kiểm tra giao diện</b>			
1	Hiển thị màn hình “Danh sách người dùng” ở trạng thái khởi tạo	Bảng trực quan kiểm tra hiển thị trên màn hình	Tại màn hình hiển thị giao diện Xem danh sách người dùng, bằng trực quan kiểm tra hiển thị trên màn hình từ trên xuống dưới từ trái sang phải: Title “Danh sách người dùng” Bảng danh sách với các cột: <ul style="list-style-type: none"> <li>● Tên đăng nhập</li> <li>● Câu châm ngôn</li> <li>● Email</li> </ul> Dữ liệu hiển thị trên 5 dòng ứng với 5 kết quả.	Tại màn hình hiển thị giao diện Xem danh sách người dùng, bằng trực quan kiểm tra hiển thị trên màn hình từ trên xuống dưới từ trái sang phải: Title “Danh sách người dùng” Bảng danh sách với các cột: <ul style="list-style-type: none"> <li>● Tên đăng nhập</li> <li>● Câu châm ngôn</li> <li>● Email</li> </ul> Dữ liệu hiển thị trên 5 dòng ứng với 5 kết quả.
...	...	...	...	...
<b>TC_02.2</b>	<b>Kiểm tra tính hợp lệ của các input</b>			
1	Tính hợp lệ của các trường dữ liệu input	Tại màn hình Tạo mới người dùng, không điền các thông tin -> chọn	<u>Hiển thị thông báo điền đầy đủ các thông tin bắt buộc.</u>	<u>Hiển thị thông báo điền đầy đủ các thông tin bắt buộc.</u>

		Tạo mới		
2		Điền chữ tiếng việt vào input username	<u>Không hiển thị chữ.</u>	<u>Hiển thị chữ.</u>
...	...	...	...	...
<b>TC_02.3</b>	<b>Kiểm tra luồng navigation, các button, thao tác</b>			
1	Button “Tạo mới chủ đề”	Tại màn hình Tạo mới người dùng, điền đầy đủ thông tin và chọn “Tạo mới”.	<u>Thông báo “Tạo mới thành công” -&gt; chuyển sang giao diện Danh sách chủ đề.</u>	<u>Thông báo “Tạo mới thành công” -&gt; chuyển sang giao diện Danh sách chủ đề.</u>
...	...	...	...	...

● Tổng kết:

<b>Tên TC</b>	Kiểm tra giao diện hiển thị chức năng quản lý người dùng và thêm chủ đề .	<b>Tổng số TC Pass</b>	16
<b>Mã TC</b>	TC_02	<b>Tổng số TC Fail</b>	2
<b>Người tạo TC</b>	Nhâm Ngọc Tuấn Anh	<b>Tổng số TC Not - Tested</b>	0
<b>Ngày tạo TC</b>	6/04/2021	<b>Tổng số TC thực hiện</b>	18

**Nhận xét:** Về dữ liệu, đa phần đều được Kiểm thử đơn vị trước nên vấn đề lấy dữ liệu ít xảy ra lỗi. Đến Kiểm thử tích hợp, DEV vừa làm, vừa kiểm thử, phát hiện lỗi ở đâu sửa ngay ở đó để đáp ứng được các yêu cầu của hệ thống. Lỗi đa phần là về điều kiện của các trường dữ liệu khi tạo mới người dùng chưa đáp ứng đúng với yêu cầu của hệ thống.

### 6.2.3. Kiểm thử trên Server

- Kiểm thử gọi API lấy dữ liệu
- Mục tiêu kiểm thử:

- Kiểm thử gọi API của tất cả các chức năng: các chức năng quản lý và chức năng xem thông số.
- *Kết quả chờ đợi:*
- Các dữ liệu hiển thị đúng và đủ các trường dữ liệu cần hiển thị trên giao diện người dùng.
- *Kết quả thực tế:* Em kiểm thử gọi API qua Postman và dưới đây là một vài kết quả được lấy ra:

STT	URI	Phương thức	Đường link	Kết quả chờ đợi	Kết quả thực tế	Kết quả (P/F)
<b>TC_01.1</b>	<b>Kiểm tra API quản lý người dùng</b>					
1	api/users	POST	http://localhost:8080/api/users	Tạo thành công, lưu dữ liệu vào db và báo thành công.	Tạo thành công, lưu dữ liệu vào DB .	<b>P</b>
2	api/users	GET	http://localhost:8080/api/users	Hiển thị danh sách người dùng phân theo trang.	Hiển thị danh sách người dùng phân theo trang.	<b>P</b>
...	...	...	...	...	...	...
<b>TC_01.2</b>	<b>Kiểm tra API quản lý các chủ đề</b>					
1	api/tags/:id	POST	http://localhost:8080/api/tags/:id	Cập nhật chủ đề	Cập nhật chủ đề thành công	<b>P</b>
2	api/tags	GET	http://localhost:8080/api/tags	Hiển thị đầy đủ danh sách.	Hiển thị đầy đủ danh sách.	<b>P</b>
...	...	...	...	...	...	...
...	...					

- Tổng kết

<b>Tên TC</b>	Kiểm tra dữ liệu khi gọi API lấy dữ liệu từ server về client.	<b>Tổng số TC Pass</b>	16
<b>Mã TC</b>	TC_01	<b>Tổng số TC Fail</b>	2
<b>Người tạo TC</b>	Nhâm Ngọc Tuấn Anh	<b>Tổng số TC Not - Tested</b>	0
<b>Ngày tạo TC</b>	6/04/2021	<b>Tổng số TC thực hiện</b>	18

- Lý do và cách khắc phục:

- Về dữ liệu để thực hiện chức năng: Xem danh sách câu hỏi của chủ đề: Dữ liệu cập nhật rất chậm cho thuật toán chưa tối ưu.
- Về dữ liệu để thực hiện các chức năng quản lý:
  - ◆ Dữ liệu trả về chưa đủ để đáp ứng yêu cầu của phía client
  - ◆ Dữ liệu trả về không đúng do API bên server đang lỗi

**Kết luận chương:** Thông qua chương 6, chúng ta có thể kiểm nghiệm được tính ổn định của hệ thống, phát hiện được lỗi phát sinh và có thể đề ra các hướng phát triển sau này.

# KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

Một forum để có thể mang tính ứng dụng cao đòi hỏi phải xử lý, và xây dựng được các chức năng có tính tương tác tốt giữa người dùng với người dùng, đồng thời phải có khả năng tính toán và xử lý khối lượng dữ liệu lớn. Forum toán học do em xây dựng vẫn còn nhiều vấn đề cần giải quyết, tuy nhiên những gì mà em đã làm được trong đề tài của mình cũng phần nào xây dựng, và định hình lên được một mạng diễn đàn toán học với những chức năng cơ bản, và mang một màu sắc riêng biệt.

Sau khi hoàn thiện đề tài của mình, em đã tích lũy được cho mình rất nhiều kiến thức và các kỹ năng trong phát triển phần mềm như:

- Phân tích yêu cầu nghiệp vụ
- Tìm kiếm và đưa ra các giải pháp tối ưu cho các vấn đề của bài toán.
- Xử lý lỗi cả trong thiết kế lẫn lập trình
- Kỹ năng báo cáo, quản lý công việc.

Ngoài ra, em cũng có thêm rất nhiều kinh nghiệm về các nền tảng ReactJS, các ngôn ngữ như Java, XML, Javascript, các design pattern. Em cũng học được cách làm việc với nền tảng quản lý phiên bản mã nguồn là Git. Đây đều là những kinh nghiệm vô cùng quý báu là những hành trang không thể thiếu giúp em tiếp tục theo đuổi con đường phát triển phần mềm trong tương lai.

Đối với diễn đàn Toán học, chắc chắn sẽ cần rất nhiều những nâng cấp và cải tiến, điển hình như:

- Cập nhật thêm nhiều dạng sự kiện khác nhau trong Forum, giúp người dùng có thêm nhiều tương tác hơn.
- Xây dựng chức năng trò chuyện riêng giữa các thành viên.
- Đưa ra thêm nhiều chức năng đánh giá người dùng, câu hỏi và câu trả lời.
- Nâng cấp hệ thống quản trị.
- Thêm tính năng khen thưởng người dùng.
- Thiết kế responsive để phù hợp với nhiều loại màn hình khác nhau

# TÀI LIỆU THAM KHẢO

## ❖ Tài liệu

1. The Power of Habit: Why We Do What We Do in Life and Business – Charles Duhigg
2. The Pragmatic Programmer: From Journeyman to Master – Andrew Hunt and David Thomas
3. Code Complete: A Practical Handbook of Software Construction, Second Edition – Steve McConnell
4. Clean Code: A Handbook of Agile Software Craftsmanship– Robert C Martin
5. Starting to Unit Test: Not as Hard as You Think – Erik Dietrich
6. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability – Steve Krug

## ❖ Các website tham khảo:

1. [https://en.wikipedia.org/wiki/React\\_\(JavaScript\\_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))
2. <https://viblo.asia/p/tong-quan-ve-kien-truc-he-thong-4P856nAL5Y3>
3. <https://topdev.vn/blog/24-code-es6-tan-tien-nhat-de-khac-phuc-loi-thuc-hanh-javascript/>
4. <https://viblo.asia/p/kien-truc-web-hien-dai-RQqKLvgMl7z>
5. <https://viblo.asia/p/the-nao-la-restful-web-service-gAm5yR4LKdb>
6. <https://topdev.vn/blog/component-trong-react-va-cach-quan-ly-chung/>
7. <https://topdev.vn/blog/bi-kip-toan-thu-ve-react-ma-ban-can-phai-biet-phan-1/>
8. <https://nordiccoder.com/blog/huong-dan-reactjs-co-ban-cam-nang-hoc-reactjs-2019/>
9. <https://topdev.vn/blog/reactjs-co-gi-loi-hai-ma-lazada-tiki-shopee-sendo-va-cac-trang-thuong-mai-dien-tu-lon-khac-deu-dang-su-dung/>
10. [https://topdev.vn/blog/nam-vung-5-khai-niem-sau-xem-nhu-master-react/?utm\\_source=facebook&utm\\_medium=ltvcfs\\_page&utm\\_campaign=blog](https://topdev.vn/blog/nam-vung-5-khai-niem-sau-xem-nhu-master-react/?utm_source=facebook&utm_medium=ltvcfs_page&utm_campaign=blog)
11. <https://viettuts.vn/java#:~:text=Java%20l%C3%A0%20m%E1%BB%99t%20m%E1%BB%99t%20ng%C3%B4n,Java%20%C4%91%C6%B0%E1%BB%A3c%20g%E1%BB%8D%20l%C3%A0%20Platform.>
12. <https://techmaster.vn/posts/36274/huong-dan-su-dung-jpa-va-hibernate-bang-spring-boot-data-jpa-phan-1>
13. <https://huongdanjava.com/vi/tong-quan-ve-spring-data-jpa.html>
14. <https://viblo.asia/p/dung-hibernate-da-lau-the-ban-co-biet-jpa-la-gi-eW65G1zxZDO>
15. <https://lcdung.top/hibernate-la-khi-gi-sao-phai-dung-no-thay-jdbc/>

16. [https://topdev.vn/blog/nam-vung-5-khai-niem-sau-xem-nhu-master-react/?utm\\_source=facebook&utm\\_medium=ltvcfs\\_page&utm\\_campaign=blog](https://topdev.vn/blog/nam-vung-5-khai-niem-sau-xem-nhu-master-react/?utm_source=facebook&utm_medium=ltvcfs_page&utm_campaign=blog)
17. <https://topdev.vn/blog/tao-slider-component-trong-react-native-bang-panresponder/>
18. <https://openplanning.net/10369/ckeditor>
19. <https://topdev.vn/blog/mot-vai-pattern-de-viet-component-cua-react-can-dung-chung-state/>
20. <https://topdev.vn/blog/bi-kip-toan-thu-ve-react-ma-ban-can-phai-biet-phan-2/>
21. <https://nordiccoder.com/blog/huong-dan-reactjs-co-ban-cam-nang-hoc-reactjs-2019/>
22. <https://topdev.vn/blog/restful-api-la-gi/#:~:text=RESTful%20API%20l%C3%A0%20m%E1%BB%99t%20ti%C3%Au,%C4%91%C6%B0%E1%BB%A3c%20truy%E1%BB%81n%20t%E1%BA%A3i%20qua%20HTTP.>
23. <https://tailieu.vn/doc/do-an-tot-nghiep-phan-tich-thiet-ke-he-thong-xay-dung-dien-dan-tren-website-1224889.html>
24. <https://hocspringmvc.net/co-che-hoat-dong-cua-authentication-xac-thuc-trong-spring-security/>
25. <https://phauer.com/2015/restful-api-design-best-practices/>