

KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM



ĐỒ ÁN MÔN HỌC
THIẾT KẾ KIẾN TRÚC PHẦN MỀM

Đề tài

Website Blog

Nhóm sinh viên thực hiện: **Nhóm 3**

Danh sách sinh viên:

- | | | | |
|----|----------------|---------|------|
| 1. | Hoàng Tuấn Anh | 0178866 | 66PM |
| 2. | Đặng Xuân Thái | 0203966 | 66PM |

Giảng viên: **ThS. Phạm Hữu Tùng**

Hà Nội, tháng 3/2025

Mục Lục

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	6
1.1 Lý do chọn đề tài	6
1.1.1 Cơ sở thực tiễn.....	6
1.1.2 Đặt vấn đề	6
1.2 Mục tiêu đồ án	6
1.3 Phạm vi và giới hạn	6
1.4 Phương pháp thực hiện	7
CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG	8
2.1 Mô tả bài toán	8
2.2 Yêu cầu chức năng.....	8
2.2.1 Quản lý người dùng	8
2.3 Yêu cầu phi chức năng	8
2.4 Use Case Diagram tổng quát	9
2.5 Mô tả các Use Case chính.....	9
2.6 Biểu đồ hoạt động.....	16
2.6.1 Biểu đồ hoạt động Đăng bài viết	16
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG.....	17
3.1 Lựa chọn mô hình kiến trúc.....	17
3.1.1 Khái niệm về Microservices	17
3.1.2 Khái niệm về Spring Cloud	17
3.2 Sơ đồ kiến trúc tổng thể.....	17
3.3 Thiết kế biểu đồ trình tự	18
3.3.1 Biểu đồ trình tự: Đăng nhập/Đăng ký.....	18
3.3.2 Biểu đồ trình tự: Xem danh bài viết	20
3.3.3 Biểu đồ trình tự: Tạo bài viết.....	21
3.3.4 Biểu đồ trình tự: Sửa bài viết.....	22
3.3.5 Biểu đồ trình tự: Chỉnh sửa thông tin cá nhân.....	23
3.3.6 Biểu đồ trình tự: Xóa bài viết	24
3.3.7 Biểu đồ trình tự: Tóm tắt bài viết	25
3.4 Thiết kế biểu đồ lớp.....	26

3.4.1 Biểu đồ lớp AuthService	26
3.4.2 Biểu đồ lớp BlogService.....	26
3.4.3 Biểu đồ lớp SmartProcessing.....	28
3.5 Thiết kế cơ sở dữ liệu	29
3.5.1 Sơ đồ quan hệ thực thể tổng quát	29
3.5.2 Bảng Users.....	30
3.5.3 Bảng Post.....	30
3.5.4 Bảng Follow	30
3.5.5 Bảng Hashtag.....	30
3.5.6 Bảng Category	31
3.5.7 Bảng Category_Blog	31
3.5.8 Bảng Comment.....	31
3.5.9 Bảng Notification	31
3.6 Thiết kế giao diện	32
3.6.1 Giao diện người dùng	32
3.6.2 Giao diện quản trị viên	35
CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG.....	38
4.1 Môi trường triển khai.....	38
4.2 Cài đặt hệ thống.....	38
4.2.1 Yêu cầu hệ thống	38
4.2.2 Các bước cài đặt chương trình.....	39
CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ	40
5.1 Kết quả thử nghiệm hệ thống.....	40
5.2 Đánh giá hiệu quả hệ thống	40

Mục Lục Bảng

Bảng 1: Use case Đăng Nhập	10
Bảng 2: Use case Đăng Ký	11
Bảng 3: Use case Tìm kiếm bài viết	12
Bảng 4: Use case Thêm bài viết mới	12
Bảng 5: Use case Sửa thông tin bài viết	12
Bảng 6: Use case Xóa thông tin bài viết.....	13
Bảng 7: Use case Thêm thể loại	14
Bảng 8: Usecase Sửa thông tin sân.....	14
Bảng 9: Use case Xóa thể loại	15
Bảng 10: Use case Sửa thông tin người dùng.....	16
Bảng 11: Bảng Users	30
Bảng 12: Bảng Post	30
Bảng 13: Bảng Follow	30
Bảng 14: Bảng Hashtag	30
Bảng 15: Bảng Category.....	31
Bảng 16: Bảng CategoryBlog.....	31
Bảng 17: Bảng Comment	31
Bảng 18: Bảng Notification.....	31

Mục Lục Hình Ảnh

Hình 1: Use case tổng quát hệ thống	9
Hình 2: Biểu đồ hoạt động Đăng bài viết	16
Hình 3: Sơ đồ kiến trúc tổng thể.....	17
Hình 4: Sequence Diagram: Đăng Nhập	18
Hình 5: Sequence Diagram: Đăng Ký	19
Hình 6: Sequence Diagram: Tìm Kiếm Sân	20
Hình 7: Sequence Diagram: Đặt Sân.....	21
Hình 8: Sequence Diagram: Sửa Bài viết.....	22
Hình 9: Sequence Diagram: Kích hoạt bài viết	23
Hình 10: Sequence Diagram: Xóa bài viết	24

Hình 11: Sequence Diagram: Tóm tắt bài viết	25
Hình 12: Biểu đồ lớp AuthService	26
Hình 13: Biểu đồ lớp PostService	26
Hình 14: Biểu đồ lớp CommentService.....	27
Hình 15: Biểu đồ lớp FollowService	27
Hình 16: Biểu đồ lớp FilterService.....	28
Hình 17: Biểu đồ lớp AIService	28
Hình 18: Sơ đồ quan hệ thực thể tổng quát	29
Hình 19: Giao diện đăng nhập người dùng.....	32
Hình 20: Giao diện đăng ký người dùng	32
Hình 21: Giao diện trang chủ	33
Hình 22: Giao diện tạo bài viết.....	33
Hình 23: Giao diện tạo bài viết.....	34
Hình 24: Giao diện trang cá nhân	34
Hình 25: Giao diện chỉnh sửa trang cá nhân.....	35
Hình 26: Giao diện trang chủ quản trị viên	35
Hình 27: Giao diện Quản lý bài viết.....	36
Hình 28: Giao diện quản lý người dùng	36
Hình 29: Giao diện quản lý thể loại.....	37

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1 Lý do chọn đề tài

1.1.1 Cơ sở thực tiễn

Ngày nay, việc chia sẻ suy nghĩ, trải nghiệm hay kiến thức qua nền tảng blog đã trở nên rất phổ biến, đặc biệt là với giới trẻ, sinh viên, người làm nội dung hoặc những ai yêu thích viết lách. Tuy nhiên, phần lớn các nền tảng blog hiện nay như WordPress, Medium hay Blogger lại khá cồng kềnh hoặc bị giới hạn trong tùy chỉnh, chưa thật sự phù hợp với người mới hoặc những ai muốn tạo dấu ấn cá nhân riêng biệt.

Bên cạnh đó, thói quen đọc của giới trẻ đang thay đổi rất nhanh. Không phải ai cũng đủ kiên nhẫn để đọc hết một bài viết dài. Họ thường lướt qua để xem nội dung chính, nếu thấy hay mới đọc kỹ. Điều đó đặt ra một vấn đề: làm sao để nội dung blog vẫn truyền tải đầy đủ thông tin nhưng không làm người đọc "ngán"?

1.1.2 Đặt vấn đề

Từ thực trạng trên, nhóm nhận thấy có 2 vấn đề lớn:

- Việc tạo và quản lý blog cá nhân vẫn chưa dễ dàng cho số đông người dùng không chuyên về công nghệ.
- Người đọc, đặc biệt là giới trẻ, thường có xu hướng đọc nhanh, đọc lướt, nên các bài viết dài đôi khi không phát huy được hết giá trị.

Vì vậy, nhóm quyết định xây dựng một hệ thống blog đơn giản, dễ sử dụng, đồng thời tích hợp thêm một chức năng rất đặc biệt: tóm tắt nội dung bài viết bằng trí tuệ nhân tạo (AI). Nhờ đó, người đọc có thể nhanh chóng nắm được nội dung chính trước khi quyết định đọc tiếp toàn bộ bài viết.

1.2 Mục tiêu đề án

Đề án nhằm thiết kế một hệ thống đặt sân thể thao trực tuyến để:

- Xây dựng một website blog đơn giản, hiện đại, thân thiện với người dùng.
- Hỗ trợ các chức năng cơ bản như đăng ký, đăng nhập, viết bài, sửa bài, xóa bài.
- Tích hợp chức năng tóm tắt bài viết bằng AI giúp người đọc dễ dàng hình dung được nội dung của bài viết.
- Cung cấp tính năng tìm kiếm, phân loại bài viết theo chủ đề, hỗ trợ bình luận.
- Hệ thống có khả năng mở rộng và dễ bảo trì, để sau này có thể thêm các chức năng mới.

1.3 Phạm vi và giới hạn

- Phạm vi nghiên cứu:

- Phát triển một website blog sử dụng các công nghệ web phổ biến (HTML, CSS, JavaScript, và một framework như React).
- Tích hợp một mô hình GPT thông qua API để xử lý tóm tắt bài viết.
- Tập trung vào các tính năng cốt lõi như giao diện blog, hệ thống quản lý nội dung, và tích hợp AI.

- Giới hạn:

- Do hạn chế về thời gian và nguồn lực, dự án sẽ không phát triển mô hình GPT từ đầu mà sử dụng các API có sẵn.
- Website sẽ được tối ưu hóa cho một số trình duyệt phổ biến (Chrome, Firefox) và chưa hỗ trợ đầy đủ trên các nền tảng ít phổ biến hơn.

1.4 Phương pháp thực hiện

Ngôn ngữ lập trình: Java, Python, Javascript

Hệ quản trị cơ sở dữ liệu: MySQL

CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG

2.1 Mô tả bài toán

Bài toán đặt ra là xây dựng một **website blog** nhằm cung cấp một nền tảng trực tuyến cho phép người dùng:

- Tạo và quản lý nội dung blog: Người dùng (quản trị viên) có thể đăng bài viết, chỉnh sửa, và quản lý nội dung một cách dễ dàng, tương tự như các nền tảng blog truyền thống (WordPress, Medium).
- Tích hợp GPT để hỗ trợ thông minh: Sử dụng mô hình ngôn ngữ để tóm tắt nội dung bài viết cho người đọc.
- Tích hợp lọc từ ngữ nhạy cảm, đem đến trải nghiệm tốt hơn cho người dùng.
- Cung cấp trải nghiệm người dùng tốt: Website cần có giao diện thân thiện, dễ sử dụng trên cả máy tính và thiết bị di động, đồng thời đảm bảo hiệu suất cao và bảo mật thông tin.

2.2 Yêu cầu chức năng

2.2.1 Quản lý người dùng

- Đăng ký/Đăng nhập: Người dùng có thể tạo tài khoản bằng email.
- Quản lý hồ sơ: Cập nhật thông tin cá nhân (tên, ngày sinh, ...).
- Phân quyền người dùng:
 - Người dùng: thêm, sửa, xóa, xem bài viết.
 - Quản trị viên: Quản lý người dùng, quản lý bài viết, báo cáo thống kê

2.2.2 Quản lý bài viết

- Thêm/Sửa/Xóa bài viết: Người dùng có thể thêm, sửa, xóa bài viết của riêng mình.
- Tìm kiếm bài viết: tìm kiếm theo nội dung, chủ đề, ...

2.2.3 Bình luận

- Bình luận ý kiến cá nhân vào các bài viết, hoặc trả lời các bình luận của người khác.

2.2.4 Báo cáo và thống kê

- Thống kê lượng bài viết, theo các doanh mục, theo loại.

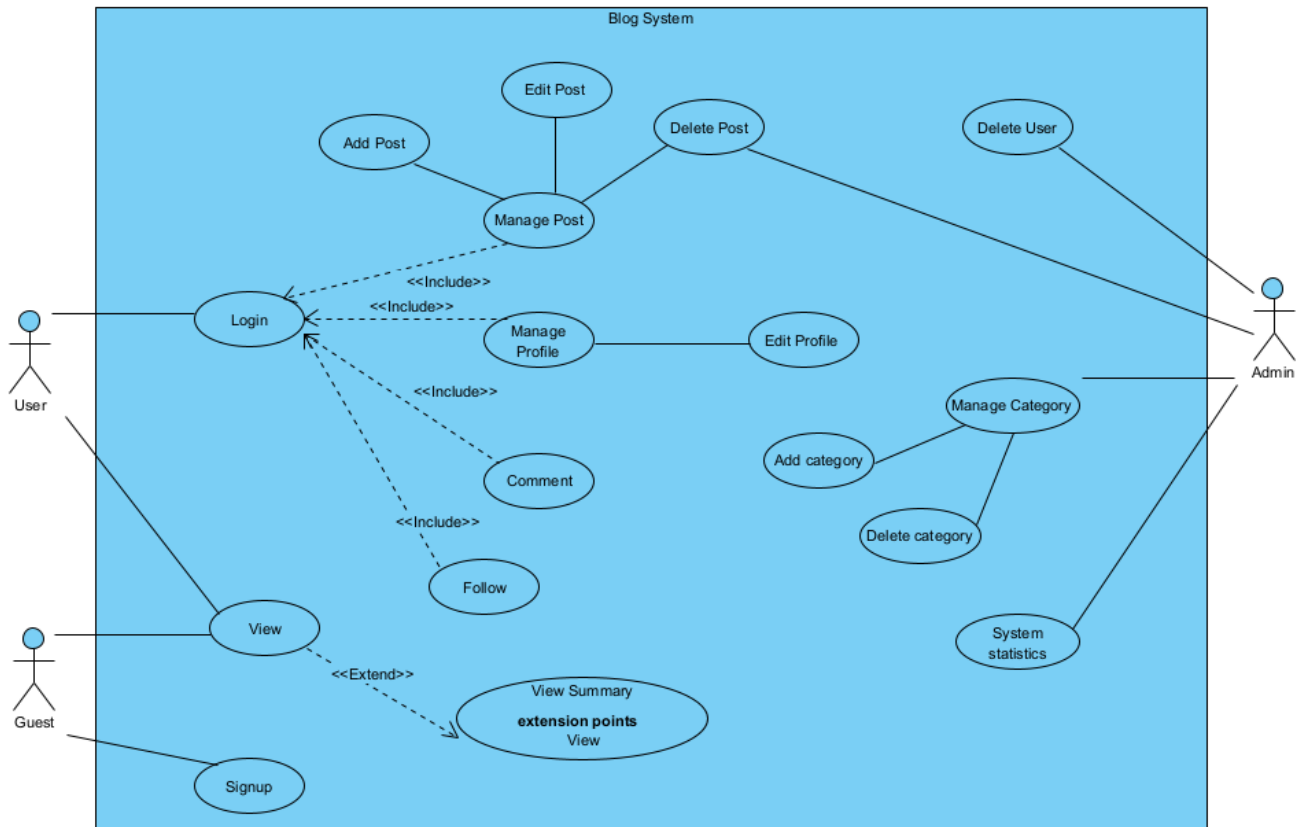
2.3 Yêu cầu phi chức năng

- Giao diện thân thiện, dễ sử dụng với người dùng.
- Ổn định, xử lý hiệu quả.
- Khả năng lưu trữ dữ liệu và phục hồi khi gặp sự cố.
- Yêu cầu phân quyền người dùng.

2.4 Use Case Diagram tổng quát

Phần mềm hệ thống quản lý và cho thuê sân thể thao bao gồm 4 tác nhân chính như sau:

- Khách vãng lai (người dùng chưa có tài khoản trong hệ thống)
- Người dùng hệ thống (người dùng đã có tài khoản và có quyền user)
- Quản trị viên (người dùng đã có tài khoản và có quyền admin)



Hình 1: Use case tổng quát hệ thống

2.5 Mô tả các Use Case chính

Use Case: Đăng Nhập

Use Case ID	UC-1
Use Case Name	Đăng nhập
Actor	User, Admin
Priority	Phải có
Brief-Description	Use case cho phép người dùng đăng nhập vào trang web
Pre-condition	Người dùng có tài khoản và mật khẩu để đăng nhập vào hệ thống
Trigger	Người dùng truy cập vào trang đăng nhập và thực hiện nhập thông tin tài khoản, mật khẩu.

Basic Flows	<ol style="list-style-type: none"> 1. Trang web hiển thị giao diện đăng nhập 2. Người dùng nhập tài khoản và mật khẩu 3. Hệ thống sẽ kiểm tra và xác nhận thông tin đăng nhập 4. Nếu thông tin hợp lệ chuyển đến giao diện chính
Alternative Flows	N/A
Exception Flows	3a. Nếu người dùng nhập sai tài khoản/mật khẩu, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.
Post-conditions	<ul style="list-style-type: none"> - Nếu đăng nhập thành công, hệ thống hiển thị giao diện chính của người dùng. - Nếu đăng nhập thất bại, hệ thống vẫn ở trang đăng nhập và hiển thị thông báo phù hợp
Business Rules	<ul style="list-style-type: none"> - Mật khẩu phải đáp ứng tiêu chuẩn bảo mật (ít nhất 8 ký tự, bao gồm chữ hoa, chữ thường, số hoặc ký tự đặc biệt). - Sau 5 lần nhập sai liên tiếp, tài khoản sẽ bị khóa trong 15 phút.
Non-Functional Requirements	<ul style="list-style-type: none"> - Hệ thống phải xử lý yêu cầu đăng nhập trong vòng dưới 2 giây. - Thông tin đăng nhập phải được mã hóa để đảm bảo an toàn dữ liệu.
Special Requirements	N/A

Bảng 1: Use case Đăng Nhập

Use Case: Đăng Ký

Use Case ID	UC-2
Use Case Name	Đăng ký
Actor	Guest
Priority	Phải có
Brief-Description	Use case cho phép người dùng đăng ký vào trang web
Pre-condition	Người dùng chưa có tài khoản đăng nhập
Trigger	Người dùng muốn tạo tài khoản
Basic Flows	<ol style="list-style-type: none"> 1. Trang web hiển thị giao diện đăng ký 2. Người dùng nhập tài khoản và mật khẩu mong muốn 3. Hệ thống sẽ kiểm tra và xác nhận thông tin đăng ký 4. Hệ thống hiển thị thông báo đăng ký thành công và chuyển hướng đến trang đăng nhập

Exception Flows	<p>3a. Nếu người dùng nhập thiếu thông tin, hệ thống yêu cầu nhập đầy đủ.</p> <p>4a. Nếu có lỗi hệ thống trong quá trình đăng ký, hệ thống hiển thị thông báo và yêu cầu thử lại sau.</p>
Post-conditions	<ul style="list-style-type: none"> - Nếu đăng ký thành công, tài khoản mới được tạo và có thể đăng nhập - Nếu đăng ký thất bại, hệ thống hiển thị thông báo lỗi phù hợp.
Business Rules	<ul style="list-style-type: none"> - Mật khẩu tối thiểu 8 ký tự, gồm chữ hoa, chữ thường, số hoặc ký tự đặc biệt.
Non-Functional Requirements	<ul style="list-style-type: none"> - Xử lý đăng ký trong ≤ 3 giây. - Mật khẩu được mã hóa trước khi lưu.
Special Requirements	N/A

Bảng 2: Use case Đăng Ký

Use Case: Tìm Bài Viết

Use Case ID	UC-3
Use Case Name	Tìm bài viết
Actor	User, Admin, Guest
Priority	Phải có
Brief-Description	Use case cho phép người dùng tìm kiếm bài viết trên web
Pre-condition	N/A
Trigger	Người dùng chọn thanh tìm kiếm bài viết và nhập thông tin cần tìm
Basic Flows	<ol style="list-style-type: none"> 1. Người dùng truy cập trang chủ có chức năng tìm kiếm bài viết. 2. Người dùng nhập thông tin tìm kiếm (nội dung, chủ đề, tác giả, ...). 3. Hệ thống kiểm tra dữ liệu và thực hiện tìm kiếm. 4. Hệ thống hiển thị danh sách bài viết phù hợp với tiêu chí tìm kiếm
Alternative Flows	<p>2a. Nếu người dùng không nhập dữ liệu tìm kiếm, hệ thống hiển thị danh sách tất cả bài viết có sẵn.</p> <p>4a. Nếu không tìm thấy kết quả, hệ thống hiển thị thông báo "Không có bài viết phù hợp".</p>
Exception Flows	N/A
Post-conditions	<ul style="list-style-type: none"> - Nếu tìm kiếm thành công, hệ thống hiển thị danh sách bài viết phù hợp. - Nếu không có kết quả, hệ thống thông báo và đề xuất các lựa chọn khác.
Business Rules	<ul style="list-style-type: none"> - Người dùng có thể tìm kiếm theo nội dung, tiêu đề, ... - Kết quả tìm kiếm sắp xếp theo thời gian đăng tải.
Non-Functional Requirements	<ul style="list-style-type: none"> - Thời gian tìm kiếm không quá 3 giây. - Giao diện thân thiện, dễ lọc và sắp xếp kết quả.
Special Requirements	- Hệ thống cần đảm bảo tốc độ tìm kiếm nhanh và chính xác.

Bảng 3: Use case Tìm kiếm bài viết

Use Case: Thêm Bài Viết

Use Case ID	UC-4
Use Case Name	Thêm bài viết điểm
Description	Người dùng thêm bài viết mới vào danh sách bài viết.
Actors	User
Priority	Must Have
Trigger	Người dùng muốn thêm bài viết mới vào danh sách bài viết
Pre-Conditions	Sử dụng tài khoản đã được phân quyền truy cập vào hệ thống.
Post-Conditions	Thêm bài viết mới thành công
Basic Flow	1. Người dùng chọn “Viết bài”. 2. Hệ thống nhận được yêu cầu và hiển thị biểu mẫu thêm bài viết mới. 4. Người dùng nhập đầy đủ thông tin về bài viết mới rồi xác nhận. 5. Hệ thống kiểm tra thông tin và thông báo “Thêm mới thành công”.
Alternative Flow	4.a. Người dùng chọn “Hủy” thêm mới. (Use Case kết thúc).
Exception Flow	5.a. Thông tin điền vào biểu mẫu không hợp lệ, hệ thống thông báo “Yêu cầu kiểm tra và nhập lại”.
Business Rules	N/A
Non-Functional Requirements	N/A

Bảng 4: Use case Thêm bài viết mới

Use Case: Sửa Thông Tin Bài Viết

Use Case ID	UC-5
Use Case Name	Sửa thông tin bài viết
Description	Chủ sở hữu sửa đổi thông tin bài viết trong danh sách bài viết.
Actors	Users
Priority	Must Have
Trigger	Chủ sở hữu muốn sửa đổi thông tin của một bài viết trong danh sách bài viết.
Pre-Conditions	Sử dụng tài khoản đã được phân quyền truy cập vào hệ thống.
Post-Conditions	Sửa đổi thông tin bài viết thành công.
Basic Flow	1. Chủ sở hữu chọn vào profile. 2. Chủ sở hữu chọn “Sửa” bài viết cần sửa. 3. Hệ thống nhận được yêu cầu và hiển thị biểu mẫu sửa thông tin bài viết. 4. Chủ sở hữu nhập thông tin cần sửa và xác nhận. 5. Hệ thống kiểm tra thông tin và thông báo “Cập nhật thành công”.
Alternative Flow	4.a. Chủ sở hữu chọn “Hủy”. (Use Case kết thúc).
Exception Flow	4.a. Thông tin điền vào biểu mẫu không hợp lệ, hệ thống thông báo “Yêu cầu kiểm tra và nhập lại”.
Business Rules	N/A
Non-Functional Requirements	N/A

Bảng 5: Use case Sửa thông tin bài viết

Use Case: Xóa Thông Tin Bài viết

Use Case ID	UC-6
Use Case Name	Xoá bài viết
Description	Chủ sở hữu xoá bài viết trong danh sách bài viết.
Actors	Users
Priority	Must Have
Trigger	Chủ sở hữu muốn xoá bài viết trong danh sách bài viết.
Pre-Conditions	Sử dụng tài khoản đã được phân quyền truy cập vào hệ thống.
Post-Conditions	Xoá bài viết thành công.
Basic Flow	<ol style="list-style-type: none"> 1. Chủ sở hữu chọn vào profile. 2. Chủ sở hữu chọn “Xoá” bài viết cần xoá. 3. Hệ thống nhận được yêu cầu và hiển thị thông báo “Xác nhận xoá bài viết này”. 4. Chủ sở hữu chọn “Xác nhận” xoá. 5. Hệ thống xoá thông tin bài viết được chọn và thông báo “Xoá bài viết thành công”.
Alternative Flow	4.a. Chủ sở hữu chọn “Huỷ”. (Use Case kết thúc).
Exception Flow	N/A
Business Rules	N/A
Non-Functional Requirements	N/A

*Bảng 6: Use case Xóa thông tin bài viết***Use Case: Thêm Thẻ Loại**

Use Case ID	UC-7
Use Case Name	Thêm thẻ loại
Actor	Admin
Priority	Phải có
Brief-Description	Use case cho phép Admin thêm thẻ loại vào trang web
Pre-condition	Người dùng có tài khoản đăng nhập vào trang web với role Admin
Trigger	Người dùng nhấn vào nút "Thêm thẻ loại" trong hệ thống quản lý.
Basic Flows	<ol style="list-style-type: none"> 1. Người dùng truy cập trang quản lý thẻ loại và chọn "Thêm thẻ loại". 2. Người dùng nhập thông tin thẻ loại (tên, mô tả). 3. Hệ thống hiển thị yêu cầu xác nhận trước khi lưu. 4. Người dùng kiểm tra và xác nhận thông tin. 5. Hệ thống kiểm tra hợp lệ và lưu thẻ loại vào cơ sở dữ liệu. 6. Hệ thống hiển thị thẻ loại mới trên trang quản lý.
Alternative Flows	2a. Nếu người dùng không nhập đầy đủ thông tin, hệ thống hiển thị thông báo lỗi và yêu cầu bổ sung.
Exception Flows	5a. Nếu hệ thống gặp lỗi khi lưu dữ liệu, hiển thị thông báo "Có lỗi xảy ra, vui lòng thử lại".
Post-conditions	- Thẻ loại mới được thêm thành công vào hệ thống và hiển thị trên giao diện quản lý.
Business Rules	- Mỗi thẻ loại phải có tên, mô tả

Non-Functional Requirements	- Quá trình thêm thẻ loại hoàn tất trong ≤ 3 giây. - Hệ thống lưu trữ thông tin thẻ loại an toàn và đồng bộ theo thời gian thực.
Special Requirements	Thông tin thẻ loại cần được hiển thị rõ ràng, dễ quản lý.

Bảng 7: Use case Thêm thẻ loại

Use Case: Sửa Thông Tin Thẻ Loại

Use Case ID	UC-8
Use Case Name	Sửa thông tin thẻ loại
Actor	Admin
Priority	Phải có
Brief-Description	Use case cho phép Admin sửa thông tin sân trên trang web
Pre-condition	Người dùng có tài khoản đăng nhập vào trang web với role Admin
Trigger	Người dùng nhấn vào nút "Chỉnh sửa" trên giao diện quản lý thẻ loại.
Basic Flows	<ol style="list-style-type: none"> 1. Người dùng truy cập trang quản lý thẻ loại. 2. Người dùng chọn thẻ loại cần chỉnh sửa và cập nhật thông tin mới. 3. Hệ thống hiển thị yêu cầu xác nhận thay đổi. 4. Người dùng kiểm tra lại thông tin và xác nhận. 5. Hệ thống kiểm tra hợp lệ, cập nhật dữ liệu và hiển thị thông tin thẻ loại mới
Alternative Flows	5a. Nếu người dùng chưa nhập đủ thông tin, hệ thống hiển thị thông báo lỗi và yêu cầu bổ sung
Exception Flows	5b. Nếu xảy ra lỗi khi lưu dữ liệu, hệ thống hiển thị thông báo lỗi và yêu cầu thử lại
Post-conditions	Thông tin thẻ loại được cập nhật thành công và hiển thị trên giao diện quản lý.
Business Rules	<ul style="list-style-type: none"> - Chỉ Admin mới có quyền chỉnh sửa thông tin thẻ loại. - Các trường quan trọng như tên sân, địa chỉ không được trùng với sân khác
Non-Functional Requirements	- Thay đổi thông tin thẻ loại phải cập nhật trong hệ thống ≤ 3 giây.
Special Requirements	N/A

Bảng 8: Usecase Sửa thông tin sân

Use Case: Xóa Thẻ Loại

Use Case ID	UC-9
Use Case Name	Xóa thẻ loại
Actor	Admin
Priority	Phải có
Brief-Description	Use case cho phép Admin xóa thẻ loại trên trang web
Pre-condition	Người dùng có tài khoản đăng nhập vào trang web với role Admin
Trigger	Người dùng nhấn vào nút "Xóa thẻ loại" trong giao diện quản lý.

Basic Flows	<ol style="list-style-type: none"> 1. Người dùng truy cập trang quản lý thẻ loại. 2. Người dùng chọn thẻ loại muốn xóa. 3. Hệ thống hiển thị yêu cầu xác nhận. 4. Người dùng xác nhận xóa thẻ loại. 5. Hệ thống kiểm tra, xóa thẻ loại khỏi hệ thống và cập nhật giao diện.
Alternative Flows	4a. Nếu người dùng hủy thao tác xóa, hệ thống giữ nguyên dữ liệu sẵn.
Exception Flows	5a. Nếu xảy ra lỗi trong quá trình xóa, hệ thống hiển thị thông báo lỗi và yêu cầu thử lại.
Post-conditions	1. Thẻ loại đã được ẩn khỏi hệ thống..
Business Rules	<ul style="list-style-type: none"> - Chỉ Admin có quyền xóa thẻ loại. - Hệ thống yêu cầu xác nhận trước khi xóa để tránh thao tác nhầm.
Non-Functional Requirements	<ul style="list-style-type: none"> - Xóa thẻ loại phải hoàn tất trong ≤ 3 giây. - Hệ thống đảm bảo không ảnh hưởng đến các dữ liệu khác khi xóa.
Special Requirements	N/A

Bảng 9: Use case Xóa thẻ loại

Use Case: Sửa Thông Tin Người Dùng

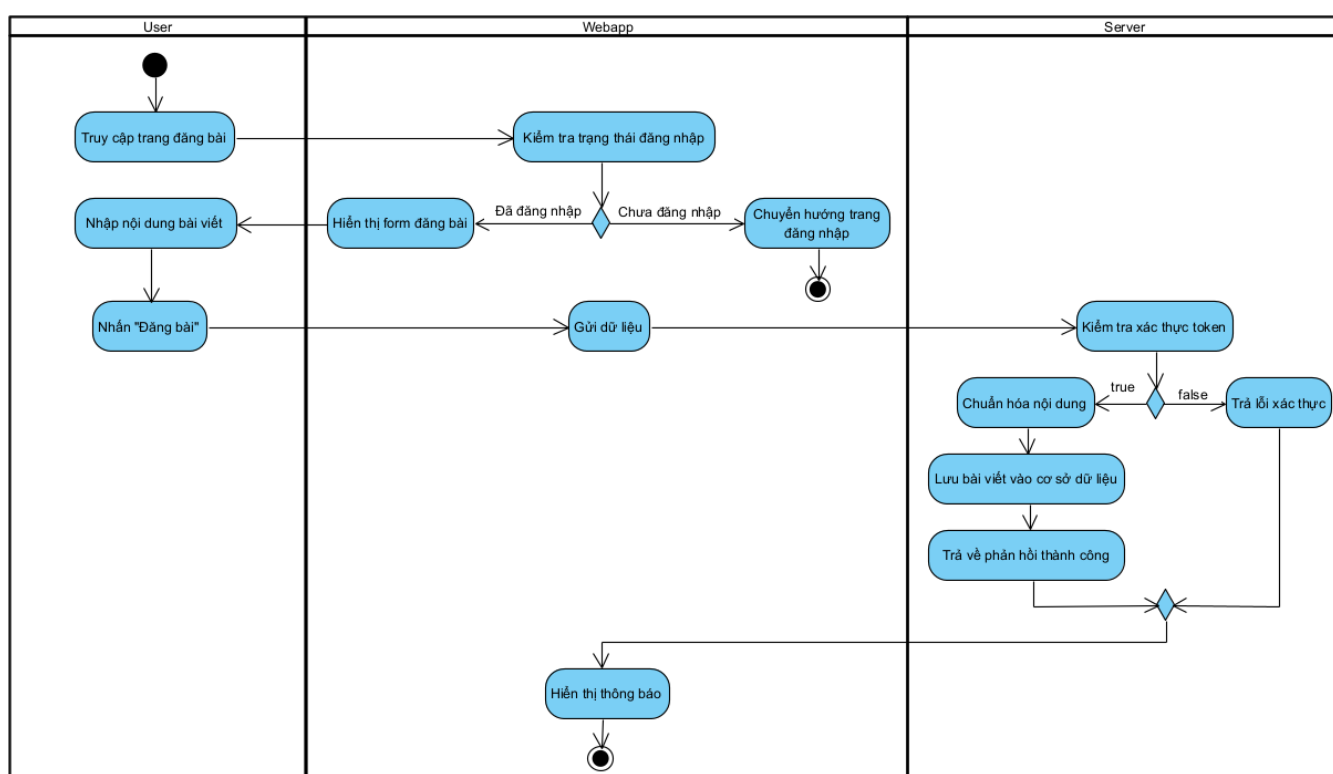
Use Case ID	UC-10
Use Case Name	Sửa thông tin người dùng.
Description	Chủ sân sửa thông tin người dùng.
Actors	Users
Priority	Must Have
Trigger	Người dùng muốn sửa đổi thông tin cá nhân.
Pre-Conditions	- Sử dụng tài khoản đã được phân quyền truy cập vào hệ thống.
Post-Conditions	Sửa thông tin cá nhân thành công
Basic Flow	<ol style="list-style-type: none"> 1. Người dùng truy cập trang cá nhân. 2. Hệ thống hiển thị thông tin cá nhân của người dùng. 3. Người dùng chọn “sửa thông tin”. 4. Hệ thống hiển thị biểu mẫu sửa. 5. Người dùng nhập thông tin vào các trường tương ứng và chọn “Xác nhận”. 6. Hệ thống kiểm tra dữ liệu được nhập vào hợp lệ. 7. Hệ thống cập nhật lại thông tin người dùng
Alternative Flow	9.a. Người dùng chọn “Hủy”. (Use Case kết thúc).

Exception Flow	6.b. Thông tin điền vào biểu mẫu không hợp lệ, hệ thống thông báo “Yêu cầu kiểm tra và nhập lại”.
Business Rules	N/A
Non-Functional Requirements	N/A

Bảng 10: Use case Sửa thông tin người dùng

2.6 Biểu đồ hoạt động

2.6.1 Biểu đồ hoạt động Đăng bài viết



Hình 2: Biểu đồ hoạt động Đăng bài viết

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

3.1 Lựa chọn mô hình kiến trúc

Dự án Website Blog được thiết kế dựa trên kiến trúc Microservices

3.1.1 Khái niệm về Microservices

Kiến trúc Microservices là mô hình thiết kế hệ thống trong đó ứng dụng được chia nhỏ thành nhiều dịch vụ độc lập, mỗi dịch vụ đảm nhiệm một chức năng riêng biệt và có thể triển khai, mở rộng, hay bảo trì riêng lẻ.

Ưu điểm:

- Dễ mở rộng và bảo trì.
- Phù hợp với phát triển theo nhóm độc lập.
- Tăng độ tin cậy vì lỗi ở một dịch vụ không ảnh hưởng toàn bộ hệ thống.

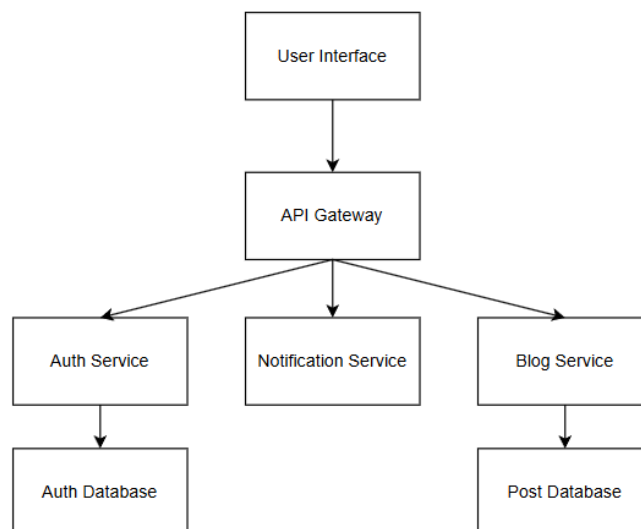
Nhược điểm:

- Phức tạp trong quản lý và triển khai.
- Giao tiếp mạng giữa các dịch vụ có thể phát sinh độ trễ.
- Cần quản lý dữ liệu phân tán, giám sát và logging phức tạp hơn.

3.1.2 Khái niệm về Spring Cloud

Spring Cloud là một framework mở rộng từ Spring, hỗ trợ xây dựng và triển khai các hệ thống ứng dụng phân tán một cách hiệu quả. Nó cung cấp các công cụ giải quyết các vấn đề như cấu hình tập trung, cân bằng tải và giám sát hệ thống. Spring Cloud tích hợp với nhiều công nghệ mã nguồn mở như Eureka, Config Server, giúp đơn giản hóa việc phát triển và quản lý hệ thống microservices.

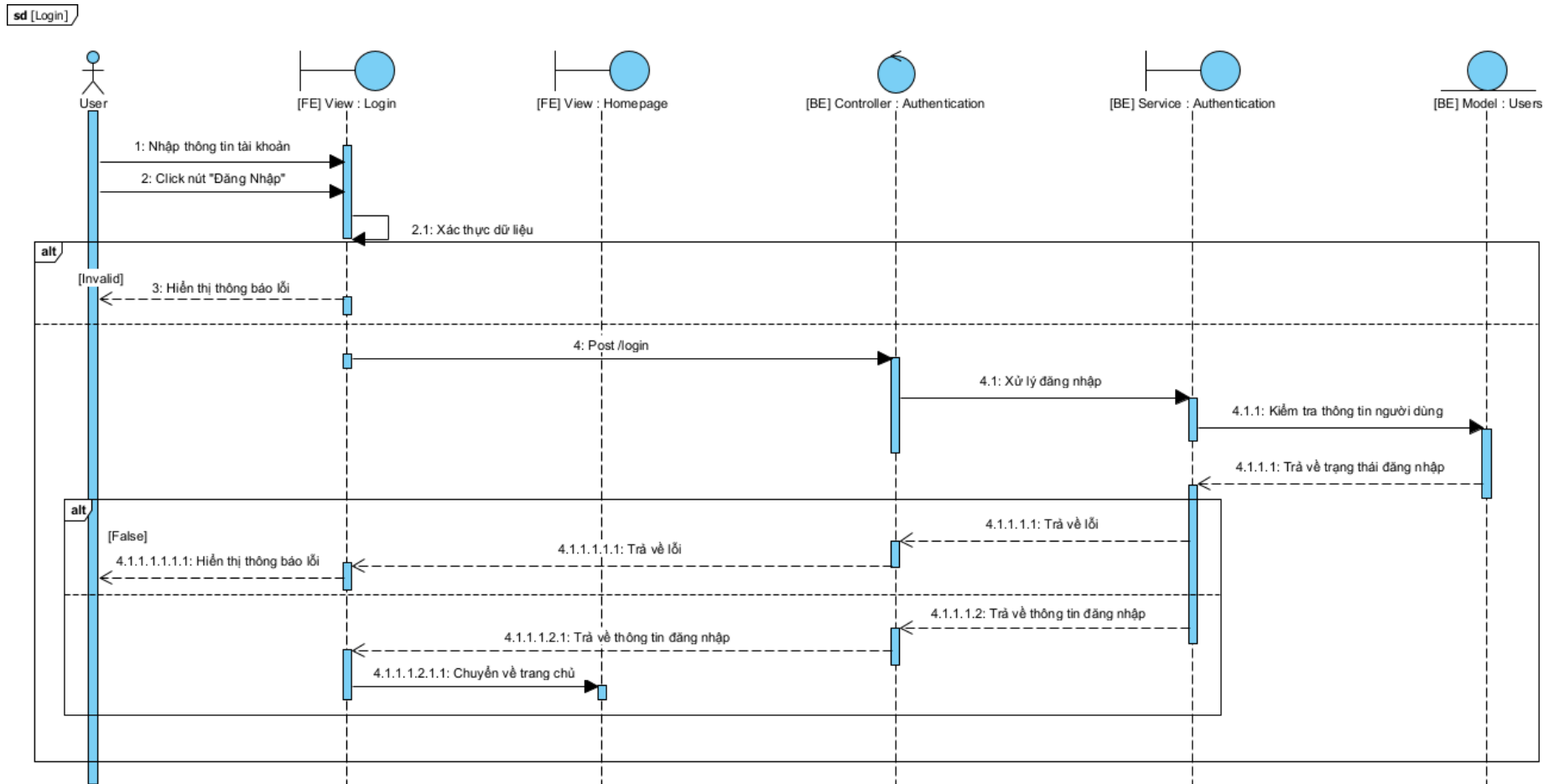
3.2 Sơ đồ kiến trúc tổng thể



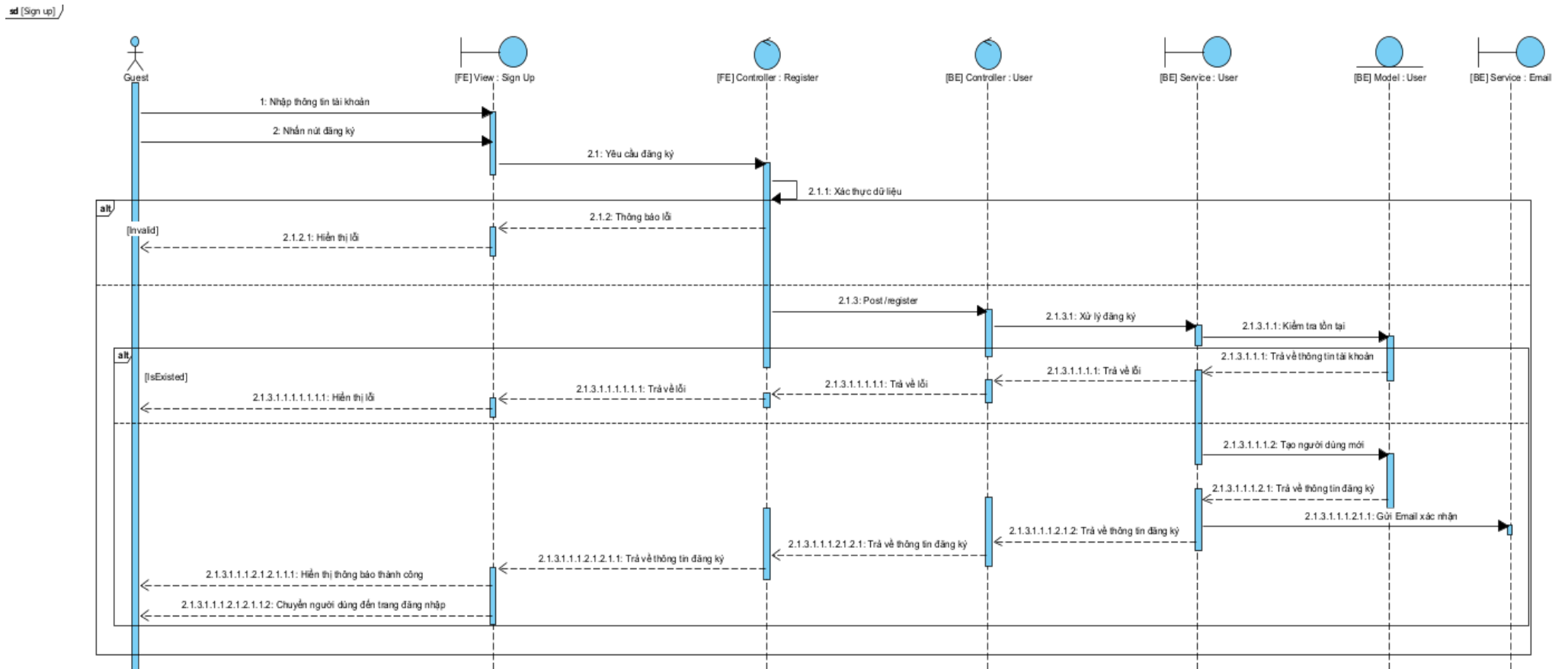
Hình 3: Sơ đồ kiến trúc tổng thể

3.3 Thiết kế biểu đồ trình tự

3.3.1 Biểu đồ trình tự: Đăng nhập/Đăng ký

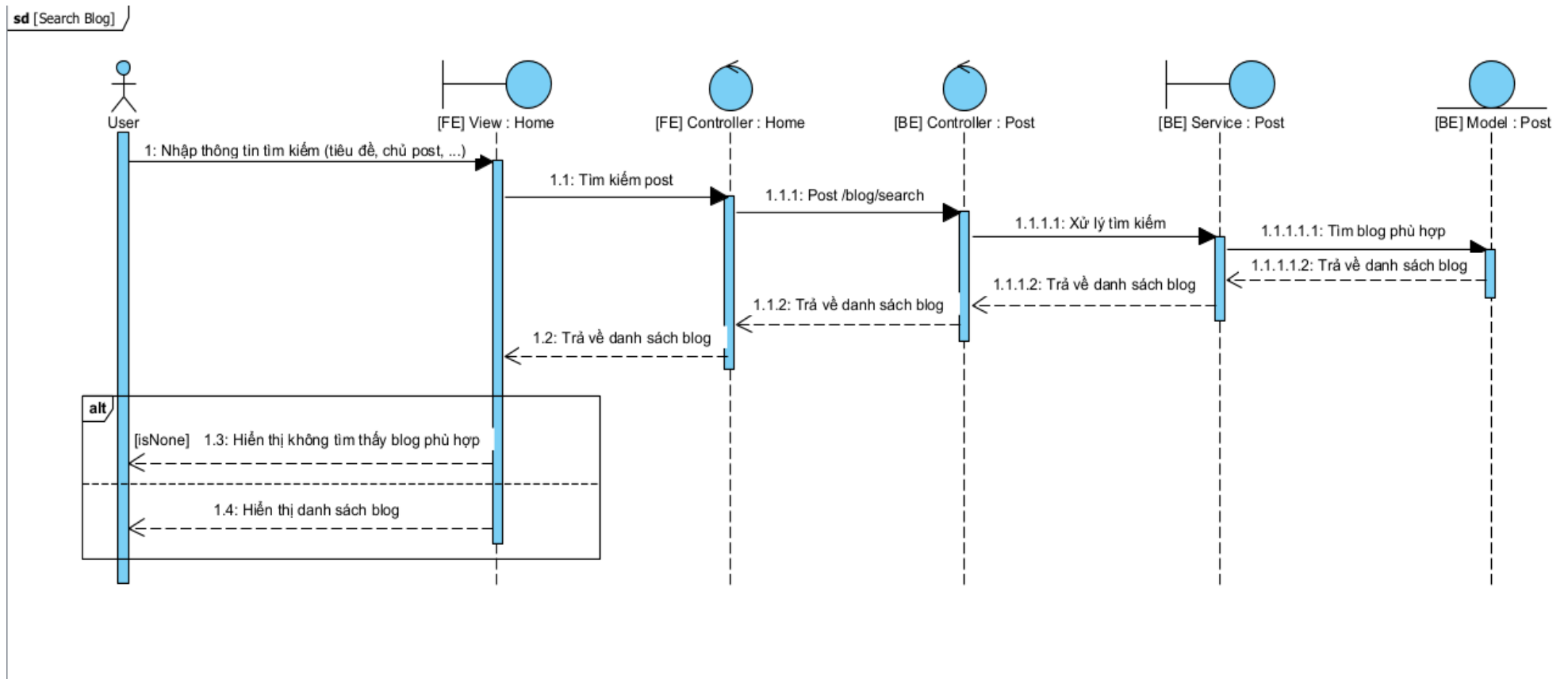


Hình 4: Sequence Diagram: Đăng Nhập



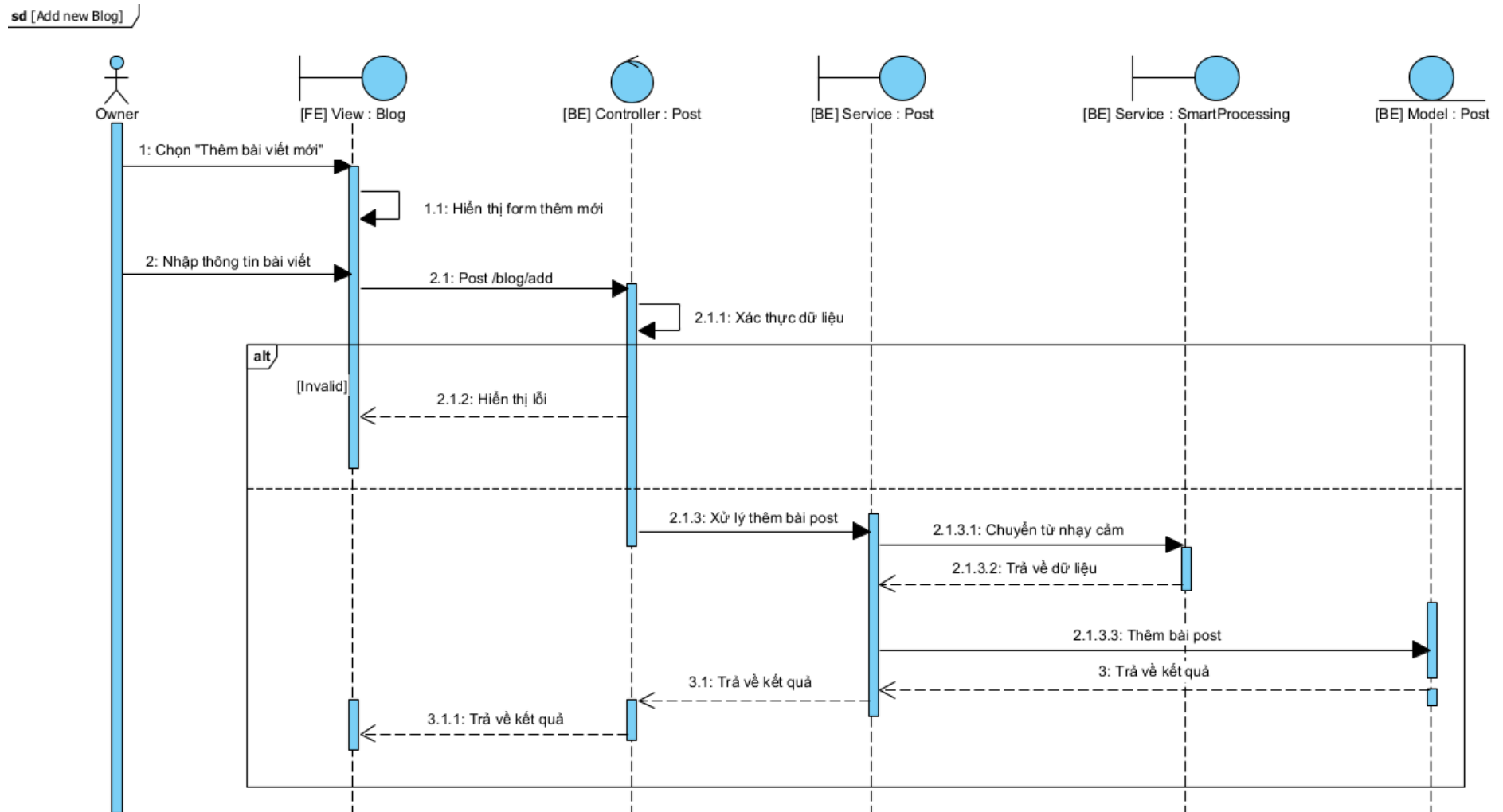
Hình 5: Sequence Diagram: Đăng Ký

3.3.2 Biểu đồ trình tự: Xem danh bài viết



Hình 6: Sequence Diagram: Tìm Kiếm Sản

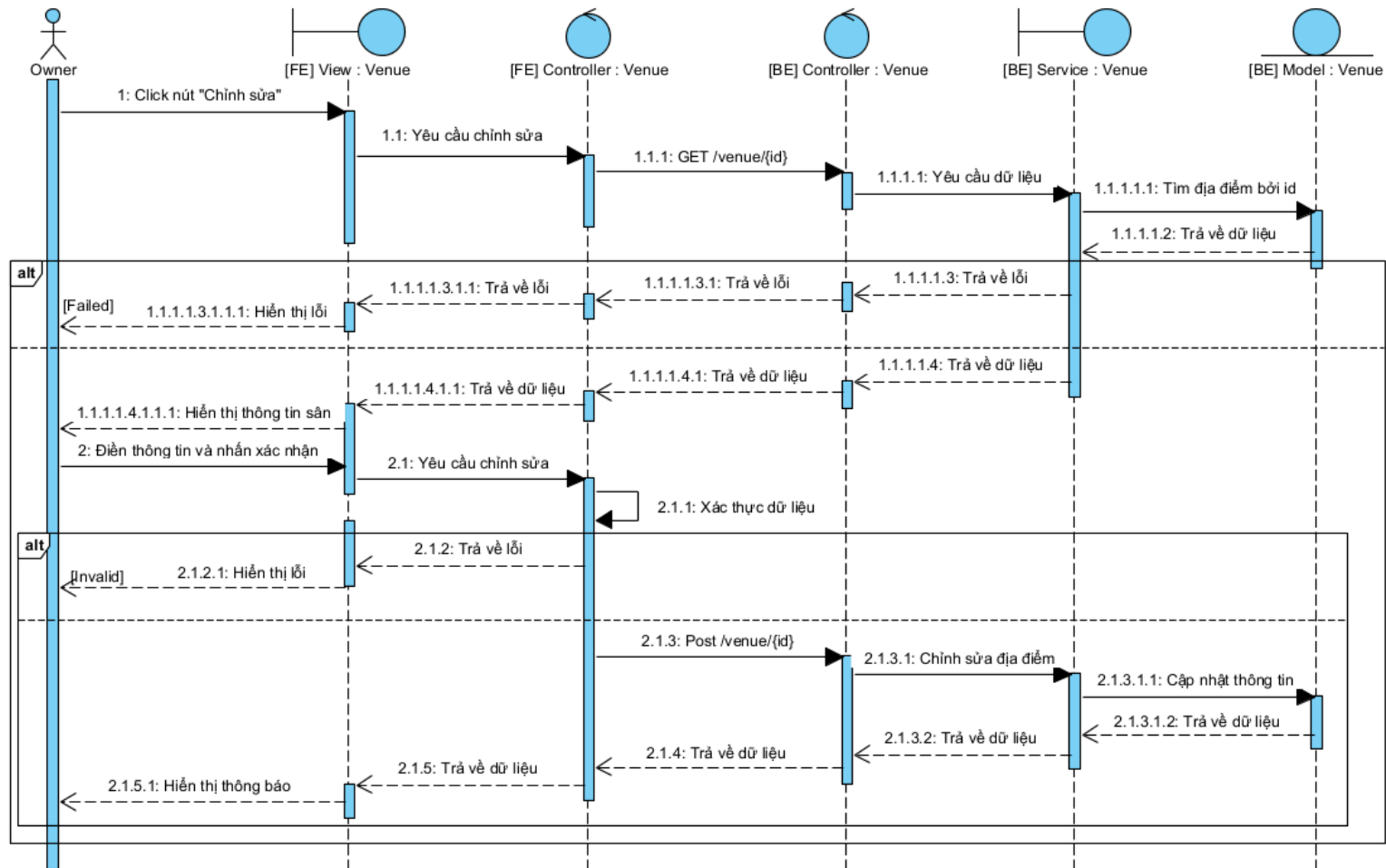
3.3.3 Biểu đồ trình tự: Tạo bài viết



Hình 7: Sequence Diagram: Đặt Sân

3.3.4 Biểu đồ trình tự: Sửa bài viết

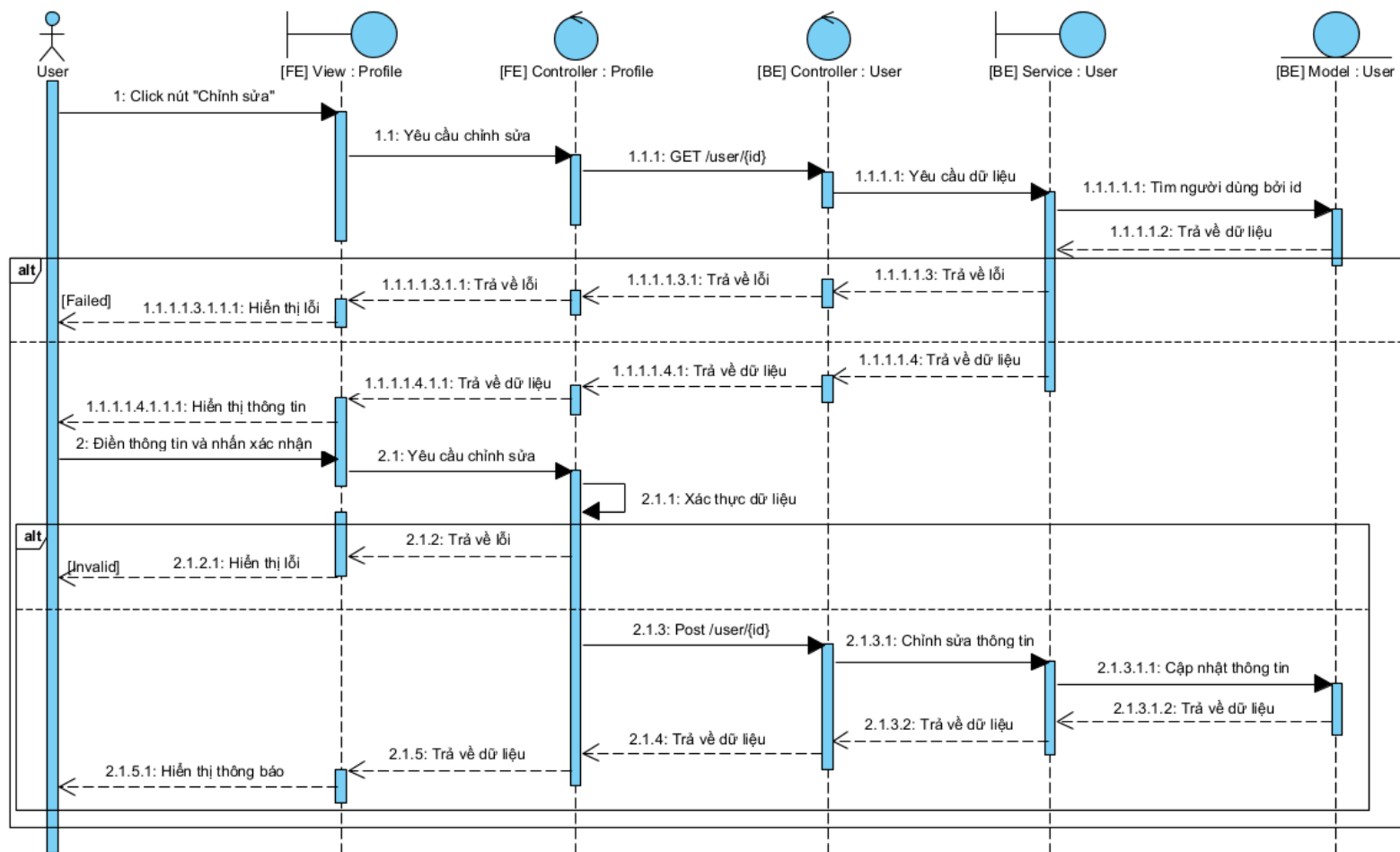
sd [Update Blog]



Hình 8: Sequence Diagram: Sửa Bài viết

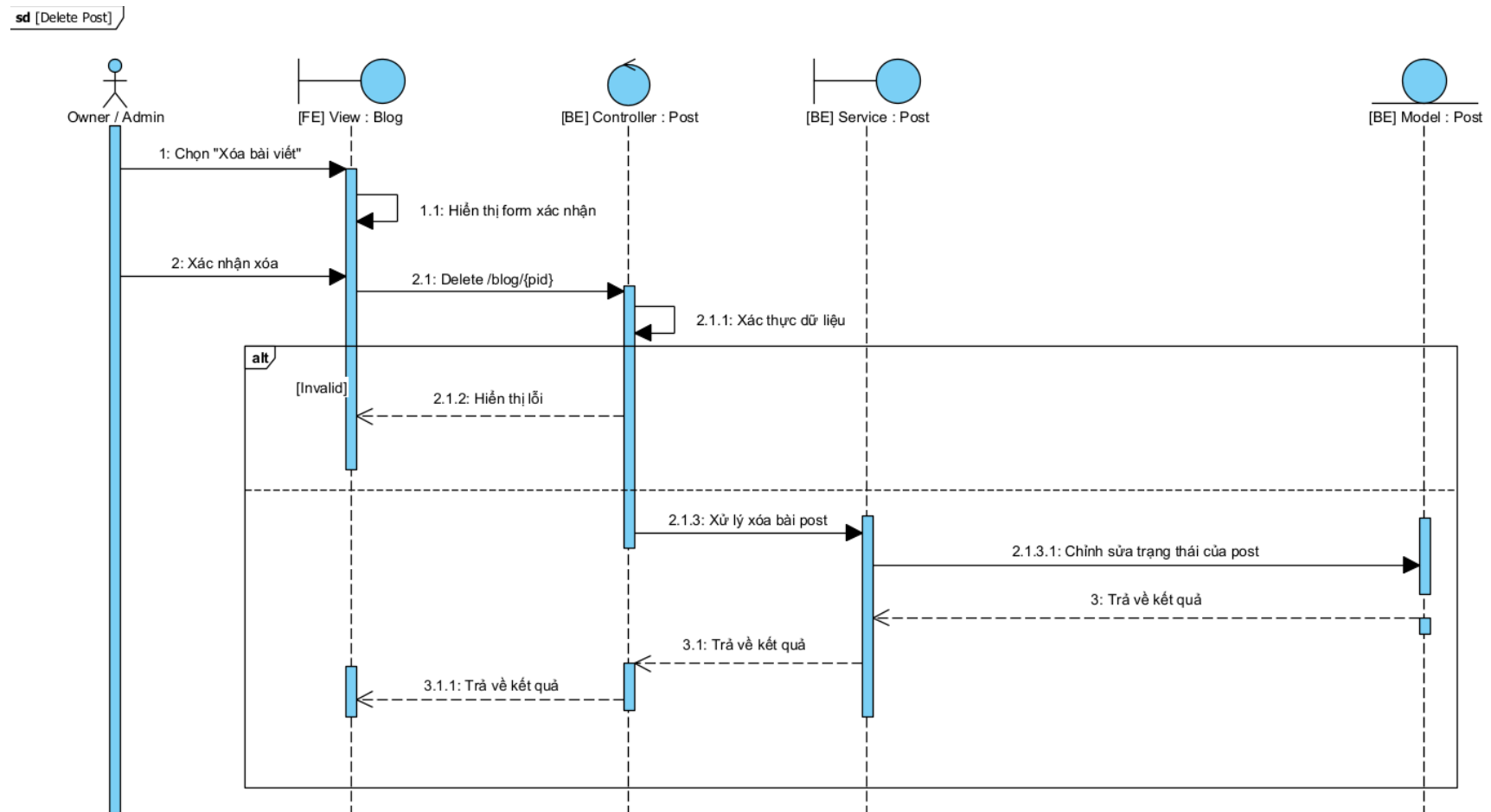
3.3.5 Biểu đồ trình tự: Chỉnh sửa thông tin cá nhân

sd [Update Profile]



Hình 9: Sequence Diagram: Kích hoạt bài viết

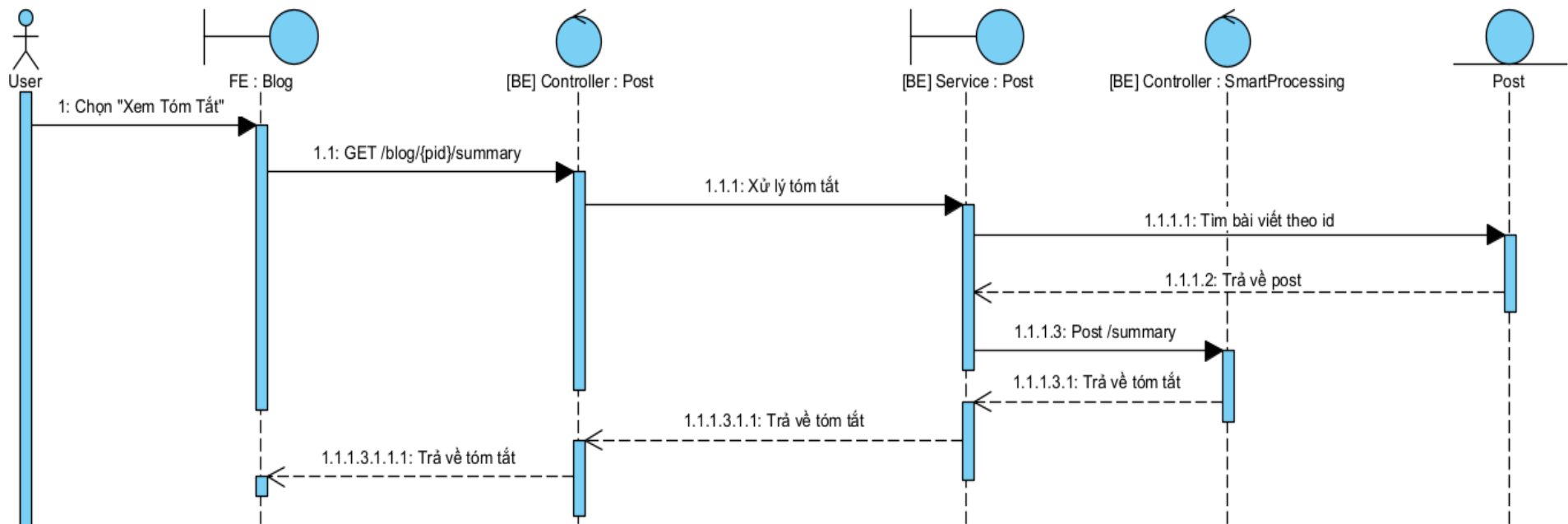
3.3.6 Biểu đồ trình tự: Xóa bài viết



Hình 10: Sequence Diagram: Xóa bài viết

3.3.7 Biểu đồ trình tự: Tóm tắt bài viết

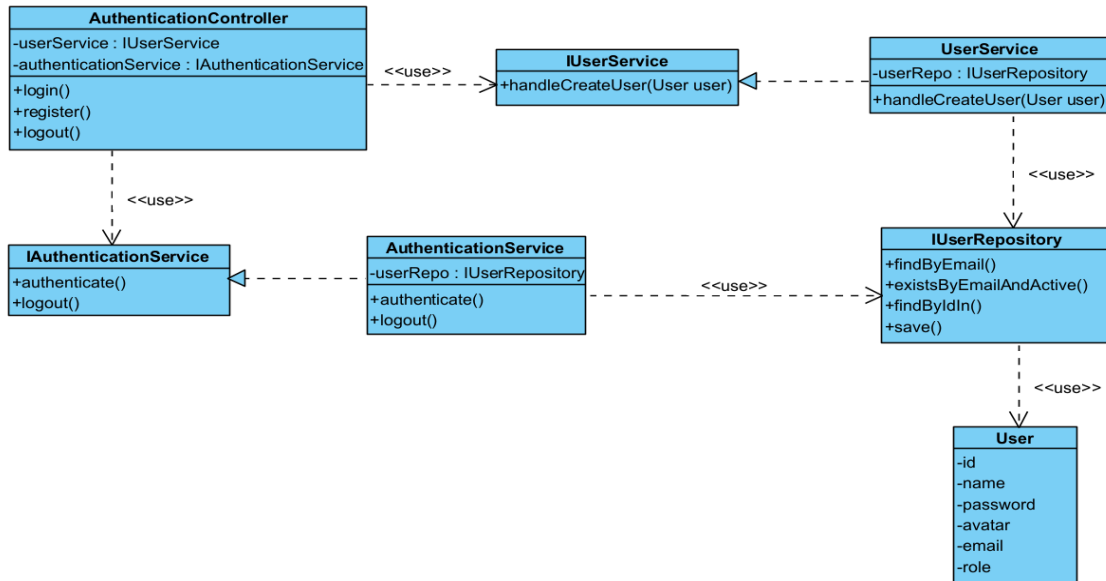
sd [AI Summary]



Hình 11: Sequence Diagram: Tóm tắt bài viết

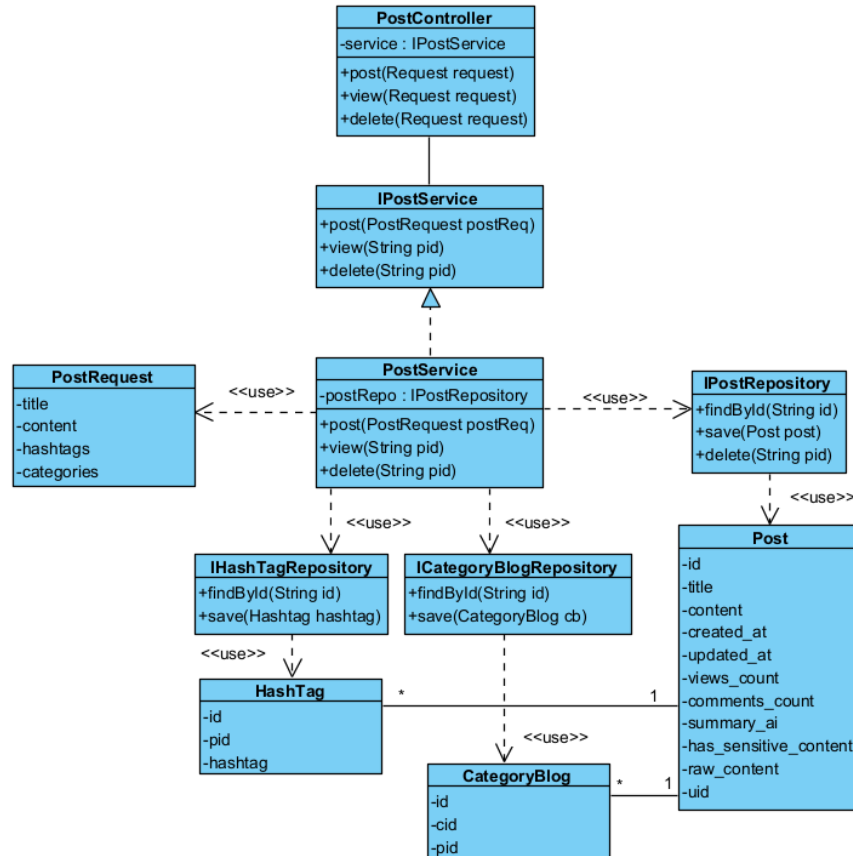
3.4 Thiết kế biểu đồ lớp

3.4.1 Biểu đồ lớp AuthService

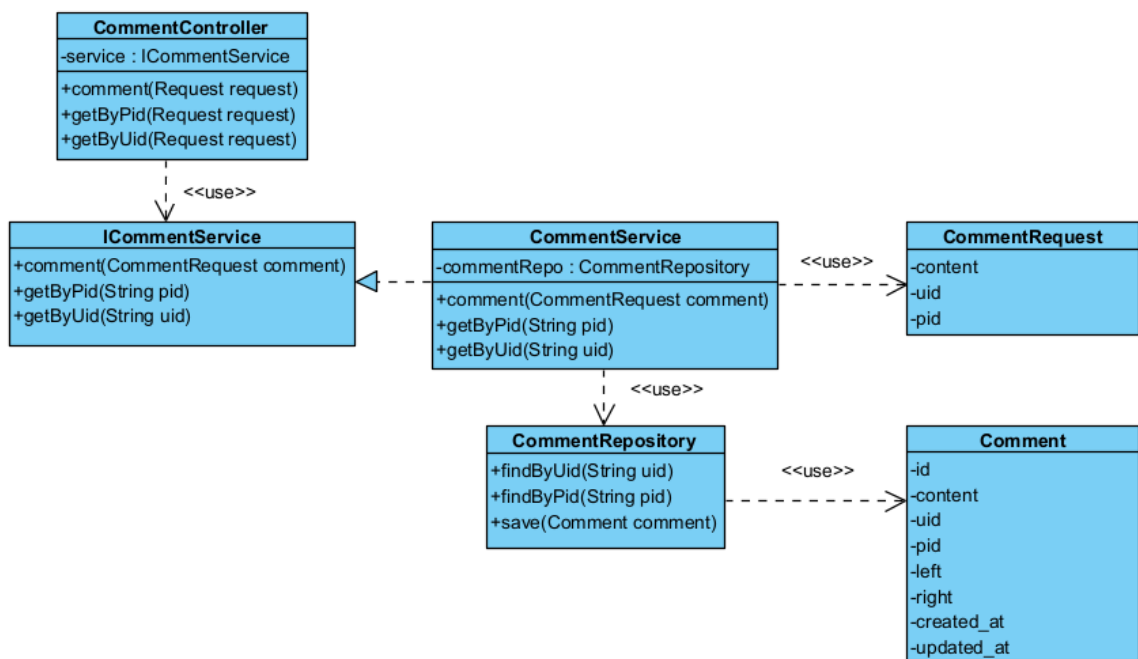


Hình 12: Biểu đồ lớp AuthService

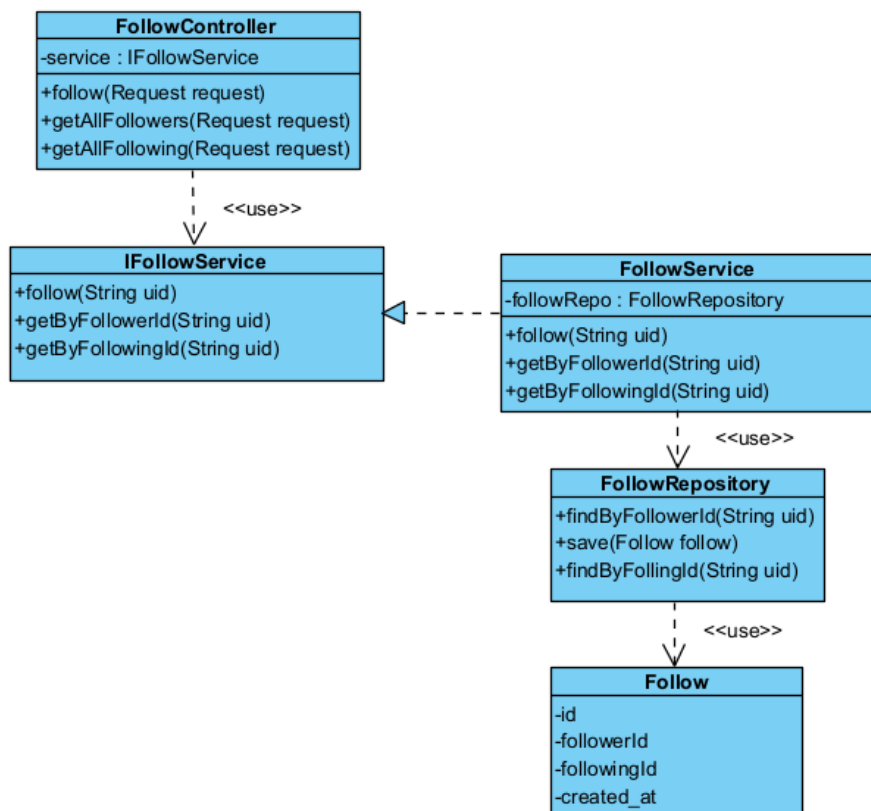
3.4.2 Biểu đồ lớp BlogService



Hình 13: Biểu đồ lớp PostService

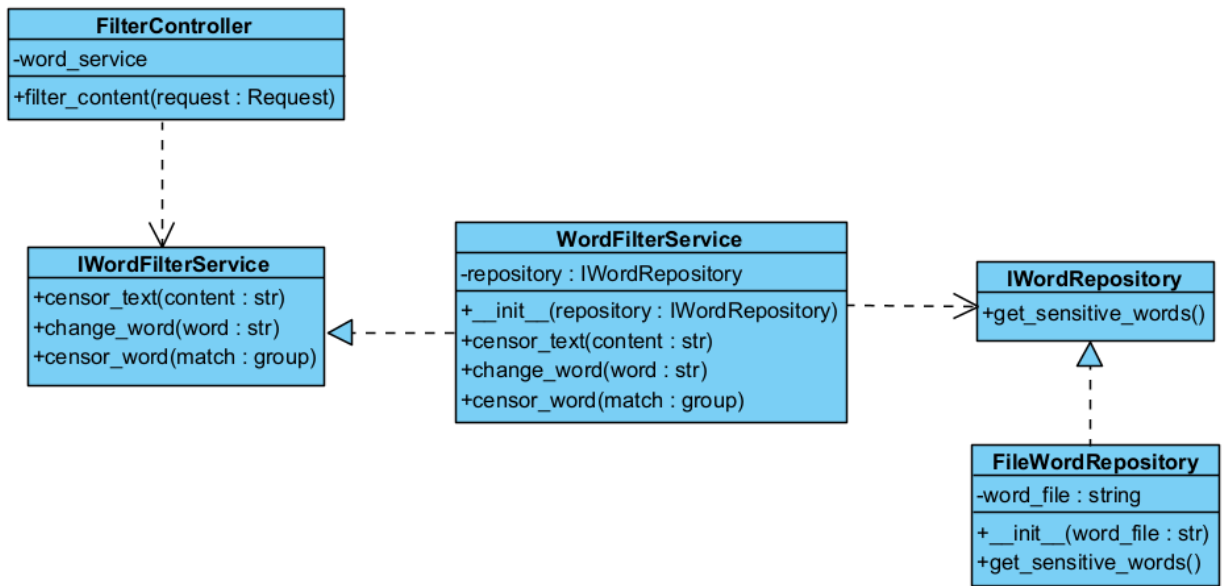


Hình 14: Biểu đồ lớp CommentService

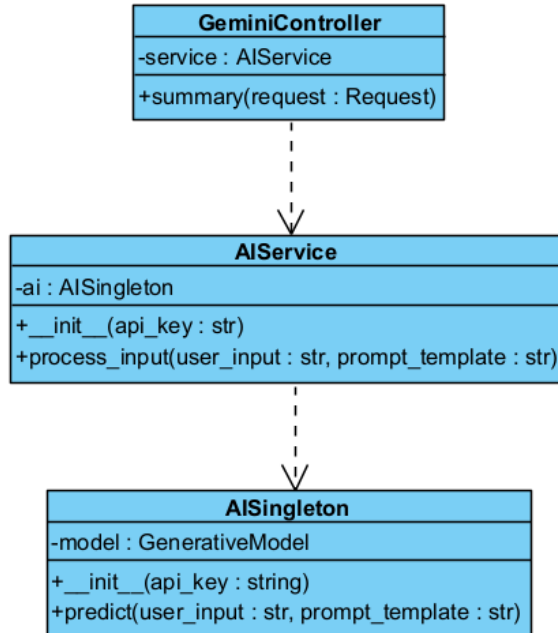


Hình 15: Biểu đồ lớp FollowService

3.4.3 Biểu đồ lớp SmartProcessing



Hình 16: Biểu đồ lớp FilterService



Hình 17: Biểu đồ lớp AIService

3.5 Thiết kế cơ sở dữ liệu

3.5.1 Sơ đồ quan hệ thực thể tổng quát



Hình 18: Sơ đồ quan hệ thực thể tổng quát

3.5.2 Bảng Users

Field Name	Data Type	Field Size	Constraint	Description
id	varchar	32	Primary Key	Mã định danh người dùng
name	nvarchar	100	Not Null	Tên người dùng
email	varchar	100	Not Null	Email
password	varchar	100	Not Null	Mật khẩu
role	enum('admin', 'user')		Not Null	Quyền
avatar	Varchar		Default 'active'	Hình ảnh đại diện

Bảng 11: Bảng Users

3.5.3 Bảng Post

Field Name	Data Type	Field Size	Constraint	Description
id	varchar	32	Primary Key	Mã định danh bài viết
uid	varchar	32	Not Null	Mã định danh chủ sở hữu
title	nvarchar	100	Not Null	Tiêu đề bài viết
content	text		Not Null	Nội dung lược bỏ nhạy cảm
raw_content	text		Not Null	Nội dung gốc
summary_ai	text		Not Null	Nội dung tóm tắt
has_sensitive_content	boolean		Default false	Chứa nội dung nhạy cảm
cover	nvarchar	100	Nullable	Ảnh bìa
view_count	int		Default 0	Số lượng người xem
comment_count	int		Default 0	Số lượng bình luận
updated_at	timestamp		Default Now()	Ngày sửa gần nhất
created_at	timestamp		Default Now()	Ngày tạo

Bảng 12: Bảng Post

3.5.4 Bảng Follow

Field Name	Data Type	Field Size	Constraint	Description
id	int		Primary Key	Mã định danh theo dõi
follower_id	nvarchar	32	Not Null	Mã định danh người dùng
following_id	nvarchar	32	Not Null	Mã định danh người dùng
created_at	timestamp		Default Now()	Ngày theo dõi

Bảng 13: Bảng Follow

3.5.5 Bảng Hashtag

Field Name	Data Type	Field Size	Constraint	Description
id	int		Primary Key	Mã định danh hashtag
pid	nvarchar	32	Foreign Key	Mã định danh bài viết
hashtag	nvarchar	100	Not Null	Nội dung hashtag

Bảng 14: Bảng Hashtag

3.5.6 Bảng Category

Field Name	Data Type	Field Size	Constraint	Description
id	int		Primary Key	Mã định danh chủ đề
cdesc	nvarchar	255	Not Null	Mô tả chủ đề
cname	nvarchar	255	Not Null	Chủ đề

Bảng 15: Bảng Category

3.5.7 Bảng Category_Blog

Field Name	Data Type	Field Size	Constraint	Description
id	int		Primary Key	Mã định danh
cid	nvarchar	32	Foreign Key	Mã định danh chủ đề
pid	nvarchar	32	Foreign Key	Mã định danh bài viết

Bảng 16: Bảng CategoryBlog

3.5.8 Bảng Comment

Field Name	Data Type	Field Size	Constraint	Description
id	int		Primary Key	Mã định danh bình luận
content	text		Not Null	Nội dung bình luận
left_value	int		Not Null	NSM Left
right_value	int			NSM Right
created_at	timestamp		Default Now()	Ngày tạo
updated_at	timestamp		Default Now()	Ngày sửa gần nhất

Bảng 17: Bảng Comment

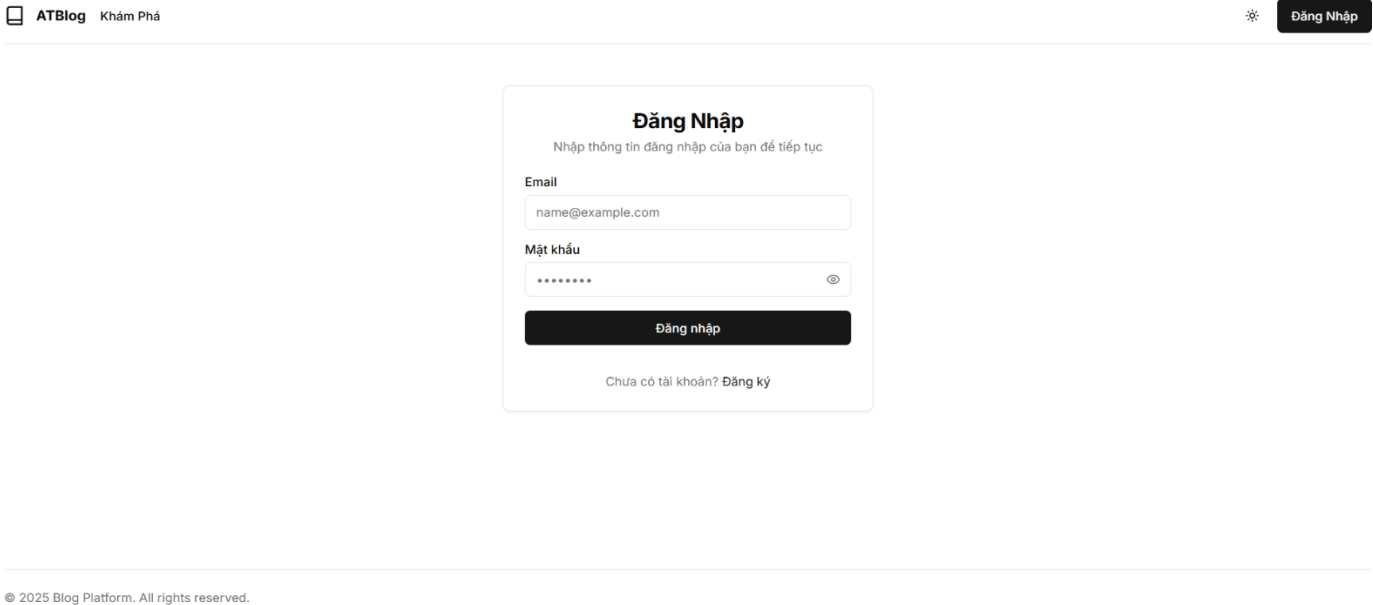
3.5.9 Bảng Notification

Field Name	Data Type	Field Size	Constraint	Description
id	int		Primary Key	Mã định danh thông báo
is_read	boolean		Default false	Trạng thái đọc
message_	text		Not Null	Nội dung thông báo
uid	nvarchar	32	Not Null	Mã định danh người dùng
pid	nvarchar	32	Foreign Key	Mã định danh bài viết
created_at	timestamp		Default Now()	Ngày tạo
updated_at	timestamp		Default Now()	Ngày sửa gần nhất

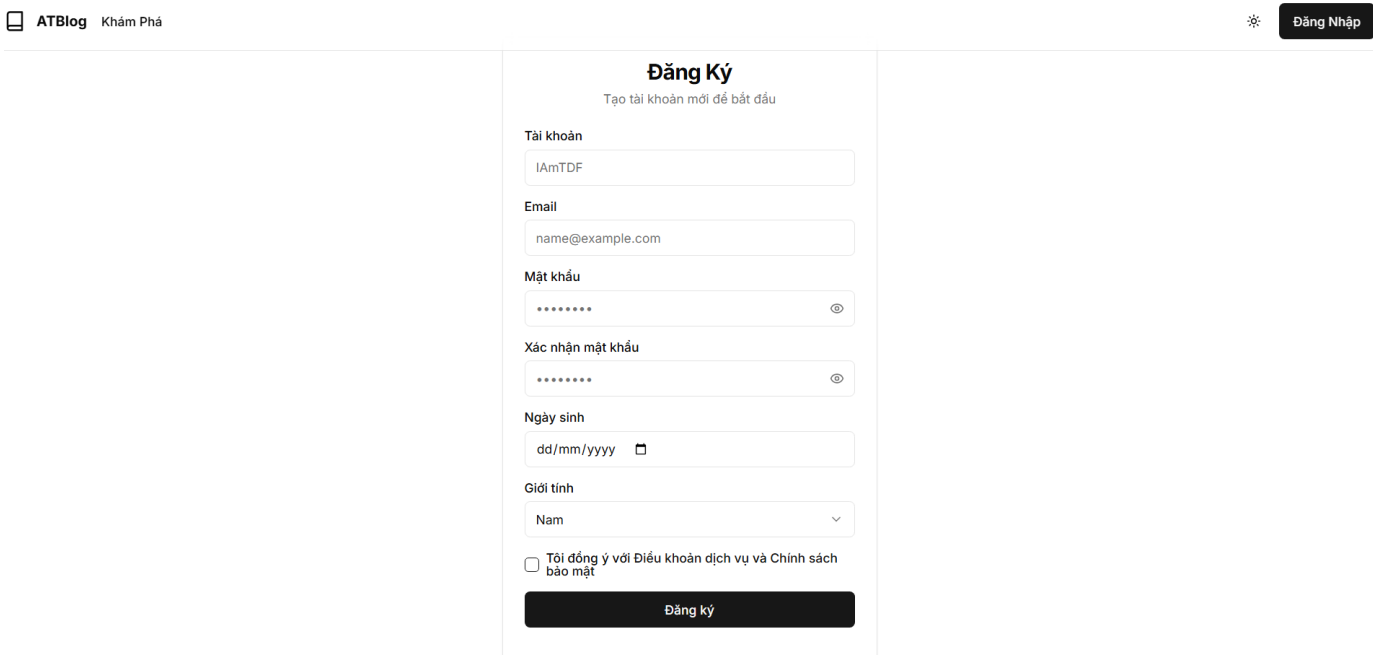
Bảng 18: Bảng Notification

3.6 Thiết kế giao diện

3.6.1 Giao diện người dùng



Hình 19: Giao diện đăng nhập người dùng



Hình 20: Giao diện đăng ký người dùng

Khám Phá Bài Viết

🔍 Tìm blogs...

≡



📁 Education

Toán Học – Ngôn Ngữ Bí Mật Của Vũ Trụ

Toán Học – Ngôn Ngữ Bí Mật Của Vũ Trụ <h2 data-start="206"

👁 1 lượt đọc 💬 0 bình luận ⌚ 25/05/2025

👤 Đặng Xuân Thái



⚠️ Nội dung nhạy cảm

📁 Heal 📁 Entertainment

Trái Đất Đang Kêu Cứu – Chúng Ta Đang Làm Gì Với Môi Trường?

Trái Đất Đang Kêu Cứu – Chúng Ta Đang Làm Gì Với Môi Trường?</h1

👁 1 lượt đọc 💬 0 bình luận ⌚ 25/05/2025

👤 Đặng Xuân Thái



📁 Technology

Trí Tuệ Nhân Tạo (AI) – Cuộc Cách Mạng Công Nghệ Đang Diễn Ra

Trí Tuệ Nhân Tạo (AI) – Cuộc Cách Mạng Công Nghệ Đang Diễn Ra</h

👁 1 lượt đọc 💬 0 bình luận ⌚ 25/05/2025

👤 Đặng Xuân Thái

Hình 21: Giao diện trang chủ

Tạo Bài Viết Mới

👤 Đặng Xuân Thái

Trình Soạn Thảo Cài Đặt

Tiêu Đề Bài Viết

📝 Nhập tiêu đề mô tả...

Ảnh Bìa (Tùy chọn)



Kéo và thả ảnh bìa hoặc nhấp để tải lên

📶 Tải ảnh lên

Nội Dung

File Edit View Insert Format Tools Table

↩️ ➡️ Paragraph System Font 12pt B I U ✖️ 🔗 📷 📺 📄 ...

Hình 22: Giao diện tạo bài viết

Tạo Bài Viết Mới

Đăng Xuân Thái

Trình Soạn Thảo

Cài Đặt

Chủ Đề

Chọn chủ đề mô tả bài viết của bạn (tối đa 3)

Technology Heal Travel Food Finance Education Entertainment Science

Thẻ

Thêm thẻ để giúp độc giả khám phá bài viết của bạn

Thêm thẻ và nhấn Enter


Thêm

Thông báo sẽ được gửi cho người theo dõi khi bạn đăng bài viết.

Xuất Bản Bài Viết

© 2025 Blog Platform. All rights reserved.

Hình 23: Giao diện tạo bài viết



Đặng Xuân Thái ♂

19/9/2003

Chỉnh Sửa Hồ Sơ

Thống Kê

Lượt Xem

3

Bài Viết

4

Follower

0

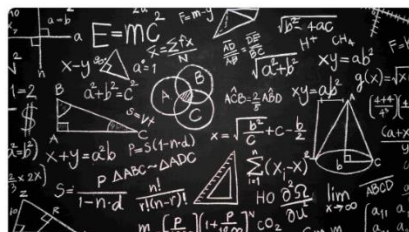
Following

1

Phân Tích

Bài Viết Người Theo Dõi Đang Theo Dõi

Bài Viết Của Bạn



Education

Toán Học – Ngôn Ngữ Bí Mật Của Vũ Trụ

Toán Học – Ngôn Ngữ Bí Mật Của Vũ Trụ <h2 data-
start="206"

1 lượt đọc

0 bình luận

25/05/2025



Technology

Trí Tuệ Nhân Tạo (AI) – Cuộc Cách Mạng Công Nghệ Đang Diễn Ra

Trí Tuệ Nhân Tạo (AI) – Cuộc Cách Mạng Công Nghệ Đang Diễn Ra</h

1 lượt đọc

0 bình luận

25/05/2025



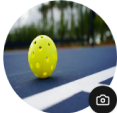
Hình 24: Giao diện trang cá nhân

Chỉnh sửa hồ sơ

Cập nhật thông tin cá nhân của bạn và cách người khác nhìn thấy bạn trên nền tảng.

Ảnh đại diện

Thông tin này sẽ được hiển thị trên trang cá nhân của bạn và trên toàn bộ trang web.



Thông tin cơ bản

Cập nhật thông tin cá nhân của bạn.

Username

Đặng Xuân Thái

Date of Birth

19/09/2003

Gender

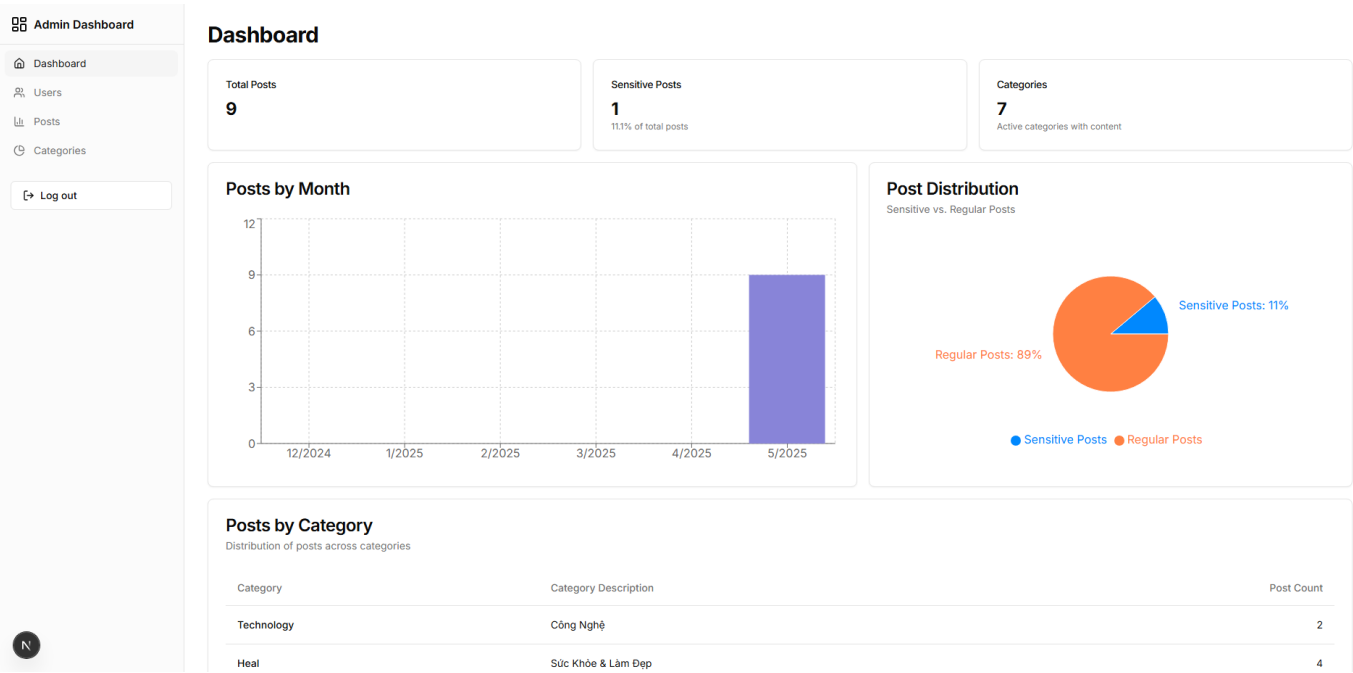
Male

Hủy

Cập nhật

Hình 25: Giao diện chỉnh sửa trang cá nhân

3.6.2 Giao diện quản trị viên



Admin Dashboard

Dashboard

Users

Posts

Categories

Log out

Post Management

Posts

Manage blog posts and content.

Q Search posts...

Title	Category	Author	Sensitive Content	Published	Views	Actions
213	TravelFinance	Đặng Xuân Thái		19/5/2025	0	...
321	TravelFinance	Đặng Xuân Thái		19/5/2025	0	...
tdf	TechnologyTravel	Đặng Xuân Thái	Sensitive	19/5/2025	1	...
heheheh	TravelFood	Đặng Xuân Thái		19/5/2025	1	...
213	HealTravel	Default User		19/5/2025	2	...
t	TravelFood	Default User		19/5/2025	2	...
Environment	Heal	Default User		12/5/2025	3	...
AI	TechnologyHeal	Default User		12/5/2025	1	...
1231233		Default User		12/5/2025	3	...

Hình 27: Giao diện Quản lý bài viết

Admin Dashboard

Dashboard

Users

Posts

Categories

Log out

User Management

Users

Q Search users...

Name	Email	Gender	Date Of Birth	Status	Actions
iamtdf	thaidangfa12@gmail.com	FEMALE	2003-09-19	Active	...
123	owner1@gmail.com	MALE	2003-09-19	Active	...
Default User	user@gmail.com	MALE	2000-01-11	Active	...
Đặng Xuân Thái	thaidangfa@gmail.com	MALE	2003-09-19	Active	...

Hình 28: Giao diện quản lý người dùng

Admin Dashboard

Dashboard

Users

Posts

Categories

Log out

N

Category Management

Categories

Manage content categories for your blog.

Q Search categories...

Name	Description	Posts
Technology	Công Nghệ	2
Heal	Sức Khỏe & Làm Đẹp	4
Travel	Du Lịch	8
Food	Ẩm Thực	3
Finance	Tài Chính	2
Education	Giáo Dục	1
Science	Khoa Học	1

Hình 29: Giao diện quản lý thể loại

37

CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

4.1 Môi trường triển khai

Hệ thống được triển khai trên môi trường cục bộ (local) với kiến trúc microservice, bao gồm nhiều service độc lập giao tiếp qua API. Môi trường triển khai được thiết lập như sau:

- Hệ điều hành: Hỗ trợ Windows 10/11, macOS, hoặc Linux (Ubuntu 20.04 trở lên), đảm bảo khả năng chạy nhiều service trên máy tính cá nhân.
- Cấu trúc microservice:
 - Service Java Spring Boot:
 - AuthService: quản lý thông tin người dùng.
 - NotificationService: quản lý việc gửi thông báo qua email cho người dùng.
 - BlogService: quản lý thông tin liên quan đến các bài viết.
 - API Gateway: định tuyến yêu cầu đến các service.
 - Service Python:
 - SmartProcessing: xử lý việc lọc từ ngữ nhạy cảm, giao tiếp với GPT.
 - Giao diện:
 - Admin Dashboard: Giao diện quản lý của quản trị viên.
 - User Frontend: Giao diện người dùng.

4.2 Cài đặt hệ thống

4.2.1 Yêu cầu hệ thống

Để cài đặt và vận hành hệ thống microservice trên môi trường cục bộ, máy tính cần đáp ứng các yêu cầu tối thiểu sau:

- Phần cứng:
 - CPU: Tối thiểu 4 core (khuyến nghị 8 core để xử lý nhiều service đồng thời).
 - RAM: Tối thiểu 8GB (khuyến nghị 16GB để đảm bảo hiệu suất khi chạy Next.js, Spring Boot, Python, và cơ sở dữ liệu).
 - Dung lượng lưu trữ: Tối thiểu 2GB trống (khuyến nghị 5GB để lưu trữ mã nguồn, cơ sở dữ liệu, và các phụ thuộc).
- Phần mềm:
 - Hệ điều hành: Windows, hoặc Linux
 - Công cụ phát triển:
 - Node.js và npm để chạy Next.js.
 - Java JDK để chạy Spring Boot.
 - Python (phiên bản 3.8 trở lên) để chạy Python Service.
 - Docker để khởi chạy các container.
 - Git để quản lý mã nguồn.
 - Maven (cho Spring Boot) để quản lý phụ thuộc Java.

- Mạng: Kết nối internet ổn định
- Khóa API: API từ Gemini

4.2.2 Các bước cài đặt chương trình

- Chạy file *docker-compose.yml* bằng lệnh `docker-compose up -d` để khởi tạo các container
- Thứ tự các service hoạt động:
 - Config-server
 - Service-registry
 - Notification-service
 - Auth-service
 - Python-service
 - Blog-service
 - Gateway-server
- Công hoạt động của các service:

Tên service	Cổng hoạt động	URL
Config-server	8071	http://localhost:8071
Sevice-registry	8070	http://localhost:8070
Notification-server	8130	http://localhost:8130
Auth-service	8080	http://localhost:8080
Python-service	1992	http://localhost:1992
Blog-service	8090	http://localhost:8090
Gateway-server	8888	http://localhost:8888

CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ

5.1 Kết quả thử nghiệm hệ thống

Sau quá trình xây dựng và hoàn thiện hệ thống website blog cá nhân tích hợp chức năng tóm tắt bài viết bằng AI, nhóm đã tiến hành thử nghiệm các chức năng chính để đánh giá mức độ hoàn thiện và khả năng vận hành của hệ thống. Cụ thể:

- Đăng ký/Đăng nhập: Người dùng có thể tạo tài khoản, đăng nhập và sử dụng hệ thống một cách dễ dàng. Hệ thống xử lý xác thực đơn giản nhưng an toàn.
- Tạo và quản lý bài viết: Người dùng có thể viết bài mới, chỉnh sửa hoặc xóa bài viết đã đăng. Giao diện viết bài thân thiện, hỗ trợ định dạng văn bản cơ bản.
- Tóm tắt bài viết bằng AI: Khi người dùng đăng bài, hệ thống sử dụng API AI để tạo ra một đoạn tóm tắt tự động. Kết quả tóm tắt khá chính xác, giúp người đọc nắm bắt nhanh nội dung chính.
- Xem bài viết và bình luận: Người dùng có thể xem danh sách bài viết, đọc nội dung chi tiết và để lại bình luận. Các thao tác này diễn ra mượt mà, ít gặp lỗi.
- Tìm kiếm và phân loại: Hệ thống hỗ trợ tìm kiếm bài viết theo tiêu đề hoặc phân loại theo chủ đề, giúp nâng cao trải nghiệm người dùng.

5.2 Đánh giá hiệu quả hệ thống

Thông qua quá trình thử nghiệm và sử dụng thử, nhóm nhận thấy hệ thống đạt được những điểm tích cực sau:

Ưu điểm:

- Giao diện trực quan, dễ sử dụng: Thiết kế đơn giản, phù hợp với người dùng phổ thông, không cần kiến thức kỹ thuật.
- Chức năng tóm tắt AI hữu ích: Giúp đáp ứng xu hướng "đọc nhanh, nắm ý chính" của nhiều bạn trẻ hiện nay.
- Khả năng mở rộng tốt: Với kiến trúc đã thiết kế, hệ thống có thể phát triển thêm các chức năng nâng cao trong tương lai.

Hạn chế:

- Khả năng tóm tắt còn phụ thuộc vào chất lượng đầu vào: Với những bài viết ngắn hoặc nhiều lỗi chính tả, AI tóm tắt chưa thực sự hiệu quả.
- Chưa tích hợp các tính năng mạng xã hội: Chức năng chia sẻ, theo dõi hoặc yêu thích bài viết vẫn chưa được triển khai.