**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

SINH VIÊN THỰC HIỆN:

Mai Đăng Khoa – 15520371

**ĐỒ ÁN MÔN ĐỒ ÁN 2**

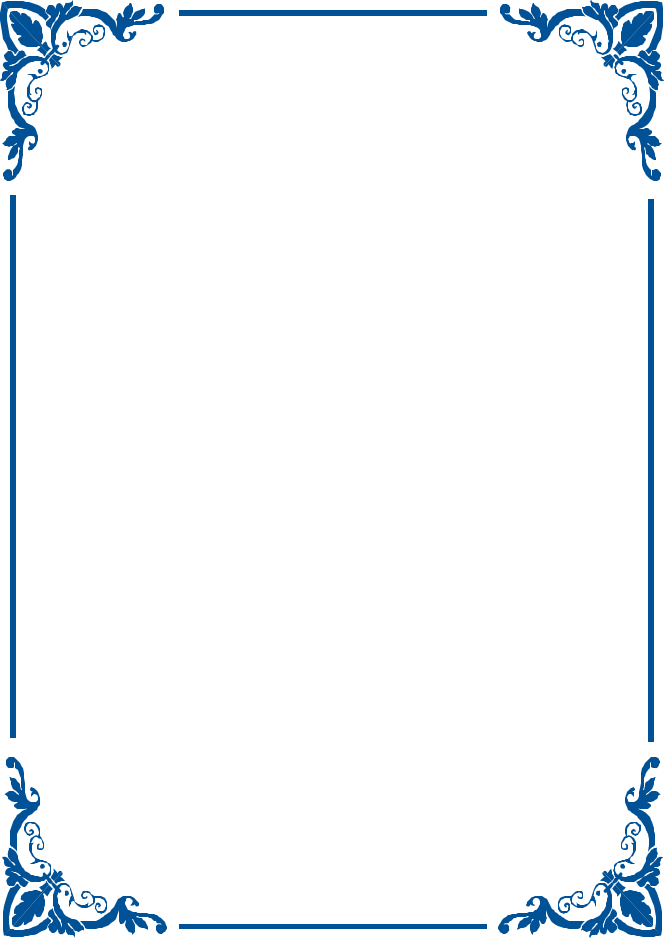
**(Lớp: SE122.L11.PMCL)**

**XÂY DỰNG WEBSITE CHIA SẺ**

**PHƯƠNG PHÁP CHĂM SÓC CÂY CẢNH**

GIẢNG VIÊN HƯỚNG DẪN:

Thầy Lê Thanh Trọng

 **LỜI CẢM ƠN**

Lời đầu tiên, em xin chân thành cảm ơn thầy phụ trách môn học đã hỗ trợ cho nhóm em trong quá trình thực hiện đề tài của đồ án.

Trong quá trình thực hiện đề tài, nhóm không tránh khỏi những thiếu sót, rất mong thầy (cô) có những ý kiến đóng góp nhằm hoàn thiện những kiến thức và kinh nghiệm để các thành viên trong nhóm có thể thực hiện các đề tài khác trong tương lai một cách có hiệu quả nhất.

**NHẬN XÉT CỦA GIẢNG VIÊN**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

**………………………………………………………………………………………………**

MỤC LỤC

[Chương 1 7](#_Toc61472264)

[THÔNG TIN CHUNG 7](#_Toc61472265)

[I. Tên đề tài: 7](#_Toc61472266)

[II. Môi trường phát triển ứng dụng 7](#_Toc61472267)

[III. Thông tin sinh viên: 7](#_Toc61472268)

[Chương 2 8](#_Toc61472269)

[PHÁT BIỂU BÀI TOÁN 8](#_Toc61472270)

[I. LÝ DO CHỌN ĐỀ TÀI: 8](#_Toc61472271)

[II. CÁC CHỨC NĂNG CỦA HỆ THỐNG: 8](#_Toc61472272)

[1. Với người dùng: 8](#_Toc61472273)

[2. Với admin 8](#_Toc61472274)

[Chương 3 9](#_Toc61472275)

[XÂY DỰNG HỆ THỐNG 9](#_Toc61472276)

[I. SƠ ĐỒ USE CASE: 9](#_Toc61472277)

[1. Sơ đồ use case cho trang web 9](#_Toc61472278)

[2. Sơ đồ use case cho web admin: 10](#_Toc61472279)

[II. SƠ ĐỒ PHÂN RÃ CHỨC NĂNG: 11](#_Toc61472280)

[1. Phân rã chức năng trang web chính: 11](#_Toc61472281)

[2. Phân rã chức năng trang web admin: 12](#_Toc61472282)

[III. DANH SÁCH CÁC ACTOR: 12](#_Toc61472283)

[IV. DANH SÁCH CÁC USE CASE: 12](#_Toc61472284)

[V. ĐẶC TẢ USE CASE: 13](#_Toc61472285)

[1. Xem tin tức: 13](#_Toc61472286)

[2. Xem bài viết của user: 14](#_Toc61472287)

[3. Xem ds user tiêu biểu 16](#_Toc61472288)

[4. Xem ds bài viết / tin tức theo tag: 16](#_Toc61472289)

[5. Theo dõi 1 user: 18](#_Toc61472290)

[6. Comment 1 bài viết: 18](#_Toc61472291)

[7. Lưu bài viết / tin tức: 19](#_Toc61472292)

[8. Viết bài: 20](#_Toc61472293)

[9. Quản lí tin tức: 21](#_Toc61472294)

[10. Quản lí bài viết: 25](#_Toc61472295)

[11. Quản lý comment: 26](#_Toc61472296)

[12. Quản lý user: 27](#_Toc61472297)

[13. Admin quản lý tag: 28](#_Toc61472298)

[14. Quản lý report: 29](#_Toc61472299)

[VI. SƠ ĐỒ LỚP: 30](#_Toc61472300)

[30](#_Toc61472301)

[VII. THIẾT KẾ CSDL: 31](#_Toc61472302)

[VIII. GIAO DIỆN: 32](#_Toc61472303)

[1. Giao diện người dùng: 32](#_Toc61472304)

[2. Giao diện admin: 35](#_Toc61472305)

[Chương 4 44](#_Toc61472306)

[TỔNG QUAN CÔNG NGHỆ 44](#_Toc61472307)

[I. Spring Boot: 44](#_Toc61472308)

[1. Khái niệm: 44](#_Toc61472309)

[2. Ưu điểm: 44](#_Toc61472310)

[II. Thymeleaf: 44](#_Toc61472311)

[1. Khái niệm: 44](#_Toc61472312)

[III. JPA: 45](#_Toc61472313)

[1. Khái niệm: 45](#_Toc61472314)

[2. Ưu điểm: 45](#_Toc61472315)

[IV. Bcrypt: 46](#_Toc61472316)

[1. Khái niệm: 46](#_Toc61472317)

[2. Ưu điểm: 46](#_Toc61472318)

[V. AdminLTE 47](#_Toc61472319)

[1. Khái niệm: 47](#_Toc61472320)

[VI. MySQL 47](#_Toc61472321)

[1. Khái niệm: 47](#_Toc61472322)

[Chương 5 48](#_Toc61472323)

[KẾT LUẬN, HƯỚNG PHÁT TRIỂN 48](#_Toc61472324)

[I. Ưu điểm: 48](#_Toc61472325)

[II. Nhược điểm: 48](#_Toc61472326)

[III. Hướng phát triển: 48](#_Toc61472327)

# 

# THÔNG TIN CHUNG

## Tên đề tài:

Website chia sẻ thông tin chăm sóc cây cảnh

## Môi trường phát triển ứng dụng

* Hệ điều hành: Microsoft Windows 10
* Công cụ xây dựng ứng dụng: Eclipse IDE for Java EE Developers
* Hệ quản trị cơ sở dữ liệu: MySQL Server
* Công cụ để vẽ sơ đồ: <https://app.diagrams.net>
* Các thư viện sử dụng:

## Thông tin sinh viên:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và tên** | **Email** |
| **1** | **15520371** | **Mai Đăng Khoa** | **15520371@gm.uit.edu.vn** |

# 

# PHÁT BIỂU BÀI TOÁN

## LÝ DO CHỌN ĐỀ TÀI:

###### Xuất phát từ mong muốn có 1 diễn đàn, 1 mạng xã hội thu nhỏ dành cho những người đam mê cây cảnh có thể giao tiếp, chia sẻ hình ảnh và bí quyết chăm sóc cây với nhau.

##### Nắm bắt việc hiện tại chưa có 1 nền tảng mxh thu nhỏ dành cho đề tài này, cũng như việc internet càng ngày càng phát triển và nền tảng web app ngày một lên ngôi. Em mong muốn có thể tạo ra 1 sản phẩm đáp ứng thị trường này.

## CÁC CHỨC NĂNG CỦA HỆ THỐNG:

### Với người dùng:

###### Viết bài viết, viết comment

###### Đăng nhập để theo dõi người dùng khác

###### Xem các bài viết của admin cũng như của người dùng khác

###### Xem các bài viết theo loại danh mục

### Với admin

###### Quản lý user

###### Quản lý bài viết

Quản lý loại danh mục

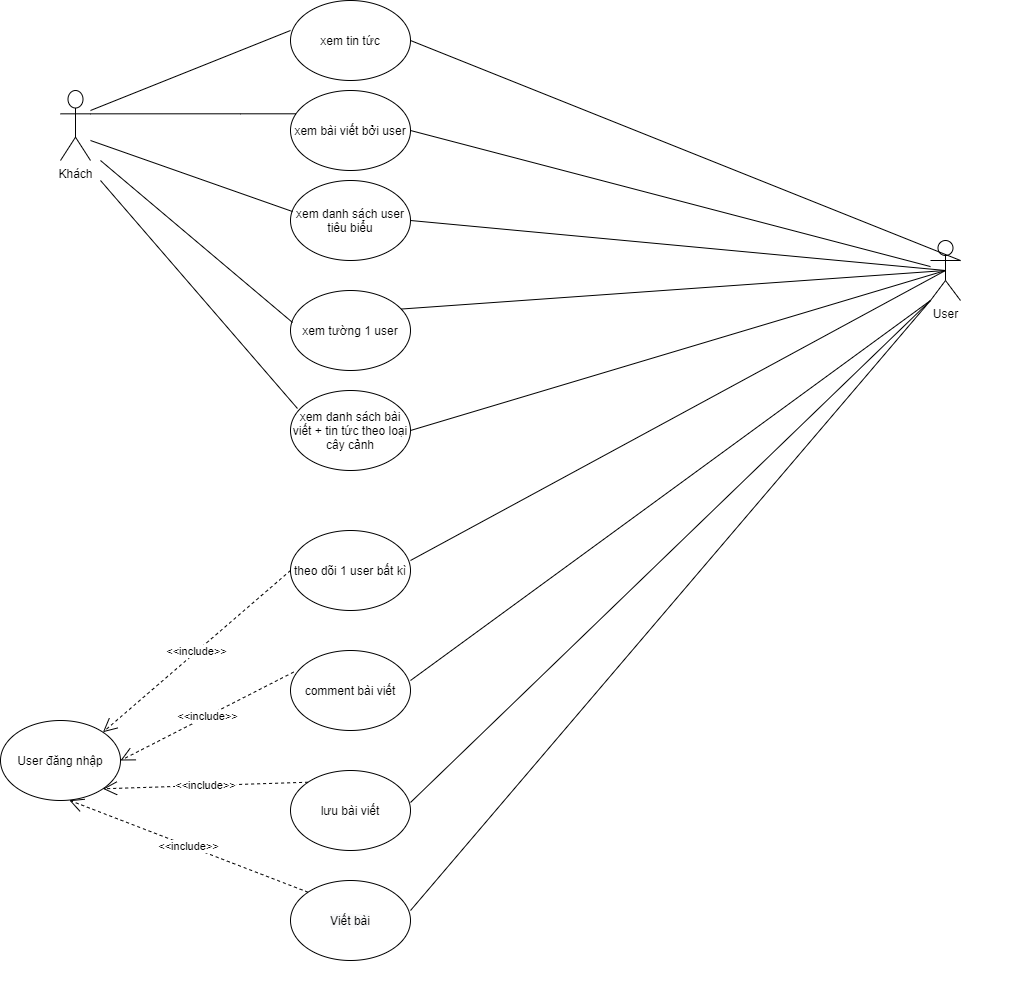
Quản lý comment

# 

# XÂY DỰNG HỆ THỐNG

## SƠ ĐỒ USE CASE:

### Sơ đồ use case cho trang web

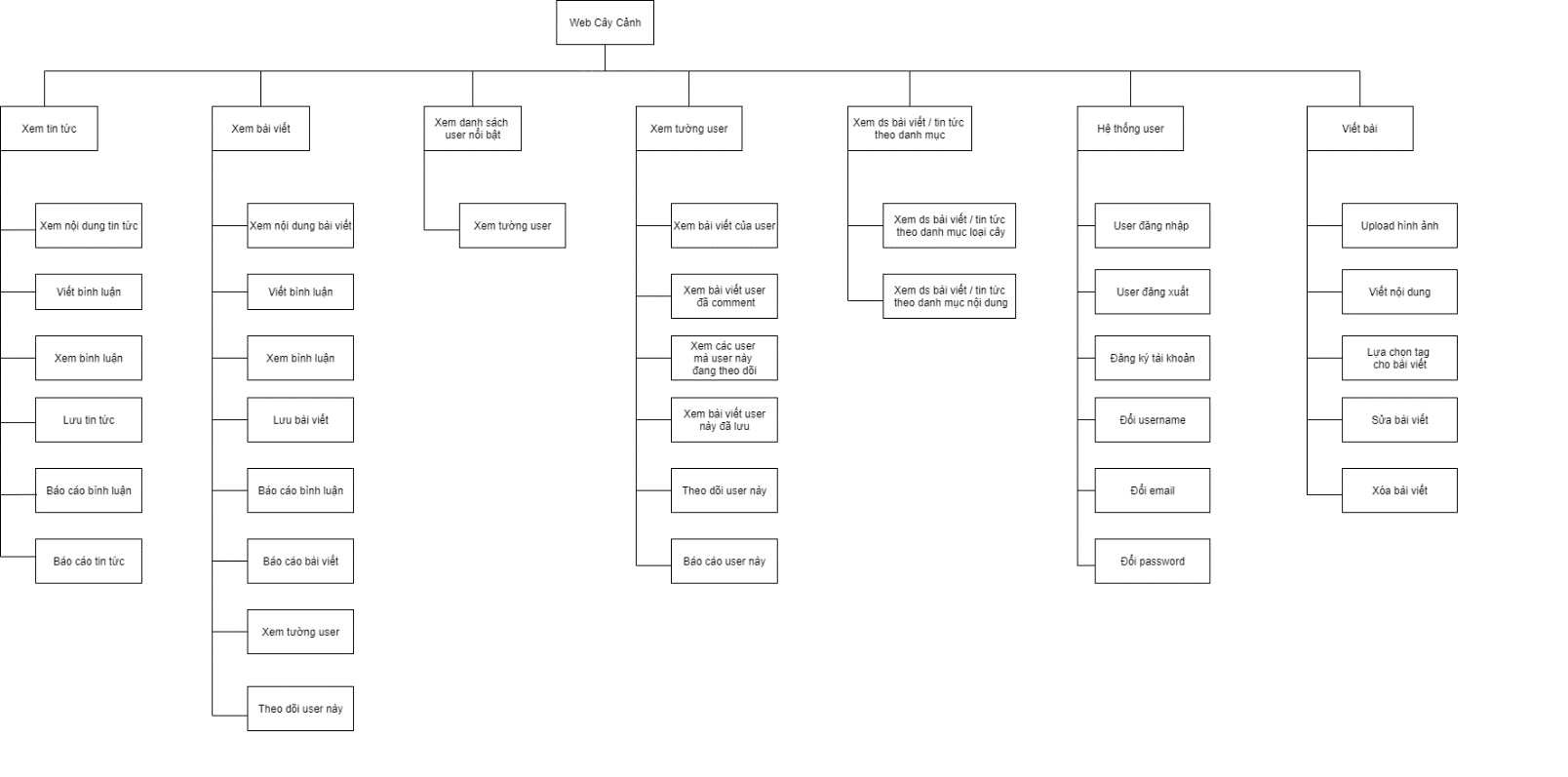


### Sơ đồ use case cho web admin:

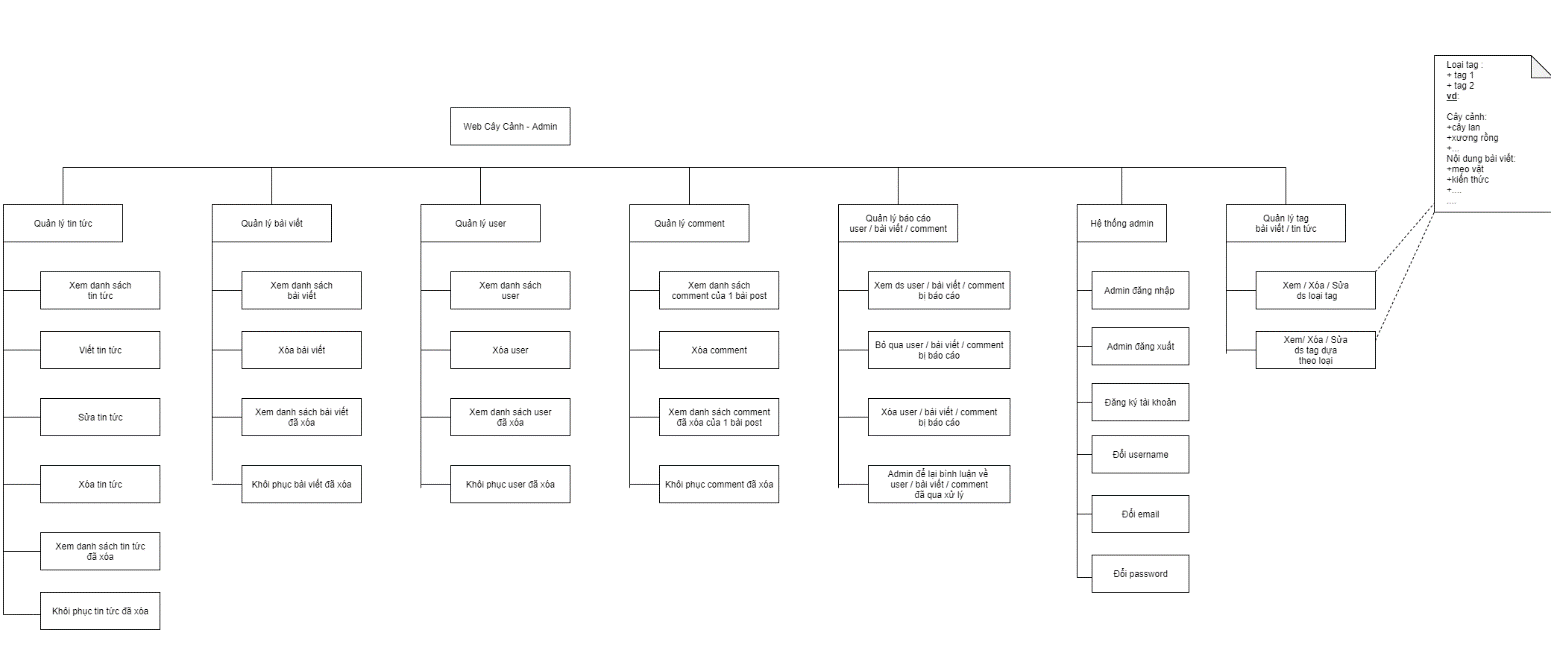


## SƠ ĐỒ PHÂN RÃ CHỨC NĂNG:

### Phân rã chức năng trang web chính:



### Phân rã chức năng trang web admin:



## DANH SÁCH CÁC ACTOR:

|  |  |  |
| --- | --- | --- |
| STT | Tên Actor | Mô tả |
| 1 | Khách | Người dùng bất kì có thể truy cập để xem nội dung được công bố tại trang web chính |
| 2 | User | Người dùng đã đăng nhập tài khoản của mình tại trang web. Có thể viết post, viết comment, theo dõi người dùng khác, … |
| 3 | Admin | Quản trị trang web, viết tin tức, thêm/xóa/sửa các post/user/comment/tag |

## DANH SÁCH CÁC USE CASE:

|  |  |  |
| --- | --- | --- |
| STT | Use Case | Mô tả |
|  | Xem tin tức |  |
|  | Xem bài viết của user |  |
|  | Xem danh sách user tiêu biểu |  |
|  | Xem tường 1 user |  |
|  | Xem tin tức / bài viết theo tag |  |
|  | Theo dõi 1 user |  |
|  | Comment 1 bài viết |  |
|  | Lưu 1 bài viết |  |
|  | Viết bài |  |
|  | Quản lí tin tức |  |
|  | Quản lí bài viết |  |
|  | Quản lí comment |  |
|  | Quản lí user |  |
|  | Quản lí tag |  |
|  | Quản lí report về user/comment/post | Xử lí các user/comment/post bị report |

## ĐẶC TẢ USE CASE:

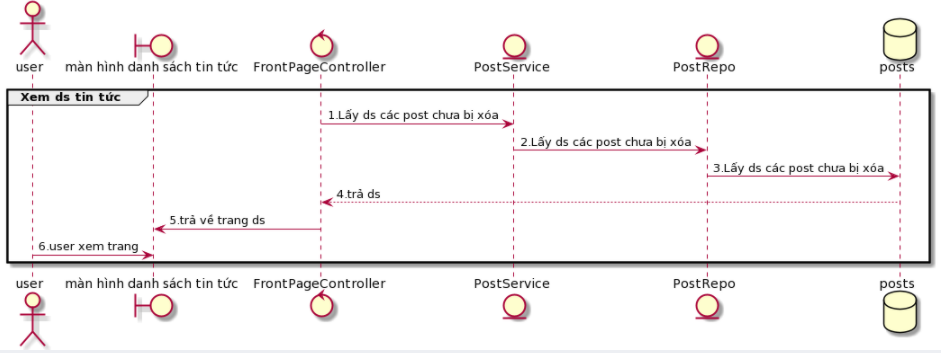
### Xem tin tức:

|  |  |
| --- | --- |
| **Use Case Name** | Xem tin tức |
| **Description** | Người xem trang web vào xem tin tức |
| **Actor(s)** | Khách, User |
| **Priority** | Phải có |
| **Trigger** | Actor muốn đọc tin tức |
| **Pre-Condition(s):** | Thiết bị người dùng đã kết nối internet |
| **Post-Condition(s):** | Hiển thị trang tin tức cho tin tức mà người dùng muốn xem |
| **Basic Flow** | 1. Người dùng đăng nhập trang web 2. Người dùng chọn tin tức muốn xem 3. Điều hướng đến trang chi tiết tin tức |
| **Alternative Flow** |  |

#### Activity diagram:



#### Sequence diagram:



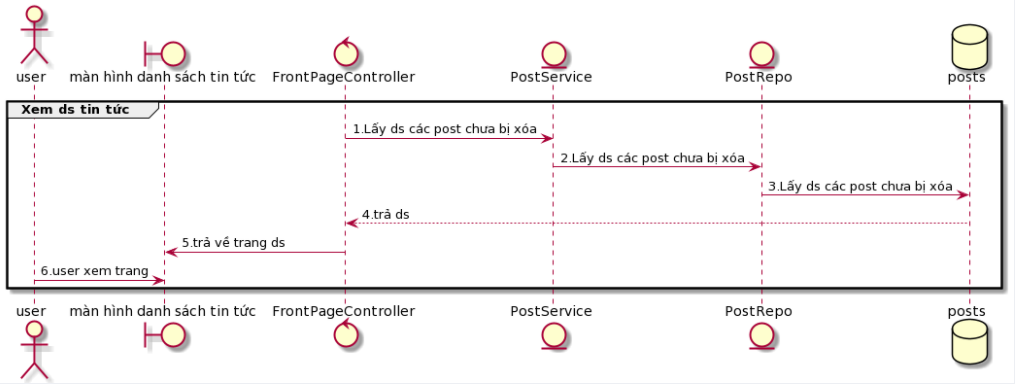
### Xem bài viết của user:

|  |  |
| --- | --- |
| **Use Case Name** | Xem bài viết của user |
| **Description** | Người xem trang web vào xem bài viết của user |
| **Actor(s)** | Khách, User |
| **Priority** | Phải có |
| **Trigger** | Actor muốn đọc bài viết của user |
| **Pre-Condition(s):** | Thiết bị người dùng đã kết nối internet |
| **Post-Condition(s):** | Hiển thị trang chi tiết bài viết của user mà người dùng muốn xem |
| **Basic Flow** | 1. Người dùng đăng nhập trang web 2. Người dùng chọn bài viết muốn xem 3. Điều hướng đến trang chi tiết bài viết của user |
| **Alternative Flow** |  |

#### Activity diagram:



#### Sequence diagram:



### Xem ds user tiêu biểu

|  |  |
| --- | --- |
| **Use Case Name** | Xem ds user tiêu biểu |
| **Description** | Xem ds những user có nhiều bài viết + nhiều cmt trên 1 bài viết |
| **Actor(s)** | Khách, User |
| **Priority** | Phải có |
| **Trigger** | Actor muốn xem ds user tiêu biểu |
| **Pre-Condition(s):** | Thiết bị người dùng đã kết nối internet |
| **Post-Condition(s):** | Show ds user tiêu biểu |
| **Basic Flow** | 1. Người dùng đăng nhập trang web 2. Người dùng chọn tab ds user tiêu biểu tại navbar 3. Điều hướng đến trang ds user tiêu biểu |
| **Alternative Flow** |  |

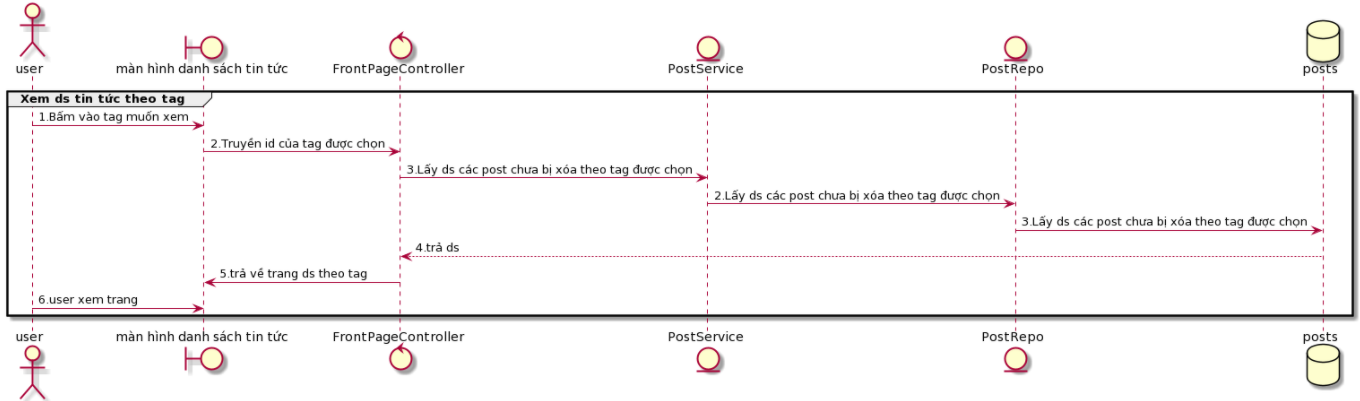
### Xem ds bài viết / tin tức theo tag:

|  |  |
| --- | --- |
| **Use Case Name** | Xem tin tức/bài viết theo tag |
| **Description** | Người dùng có thể xem danh sách các tin tức/bài viết dựa theo tag |
| **Actor(s)** | Khách, User |
| **Priority** | Phải có |
| **Trigger** | Actor muốn xem danh sách các tin tức/bài viết dựa theo tag |
| **Pre-Condition(s):** | Thiết bị người dùng đã kết nối internet |
| **Post-Condition(s):** | Show danh sách các tin tức/bài viết dựa theo tag |
| **Basic Flow** | 1. Người dùng đăng nhập trang web 2. Người dùng chọn tab danh mục 3. Điều hướng đến trang danh sách các tag 4. Người dùng chọn tag mình muốn 5. Điều hướng đến trang danh sách các tin tức/bài viết dựa theo tag |
| **Alternative Flow** |  |

#### Activity diagram:



#### Sequence diagram:



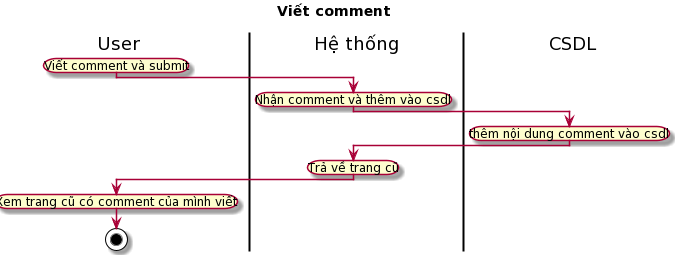
### Theo dõi 1 user:

|  |  |
| --- | --- |
| **Use Case Name** | Theo dõi 1 user |
| **Description** | User A sau khi đăng nhập có thể theo dõi 1 user B, sau khi theo dõi, user A có thể xem các bài viết bởi user B ở mục bài viết của các user đang theo dõi |
| **Actor(s)** | User |
| **Priority** | Phải có |
| **Trigger** | User muốn theo dõi 1 user khác |
| **Pre-Condition(s):** | Thiết bị người dùng đã kết nối internet |
| **Post-Condition(s):** | User được theo dõi sẽ được thêm vào danh sách đang theo dõi của user đang đăng nhập |
| **Basic Flow** | 1. User bấm nút theo dõi 1 user bất kì 2. User được theo dõi sẽ được thêm vảo ds theo dõi |
| **Alternative Flow** |  |

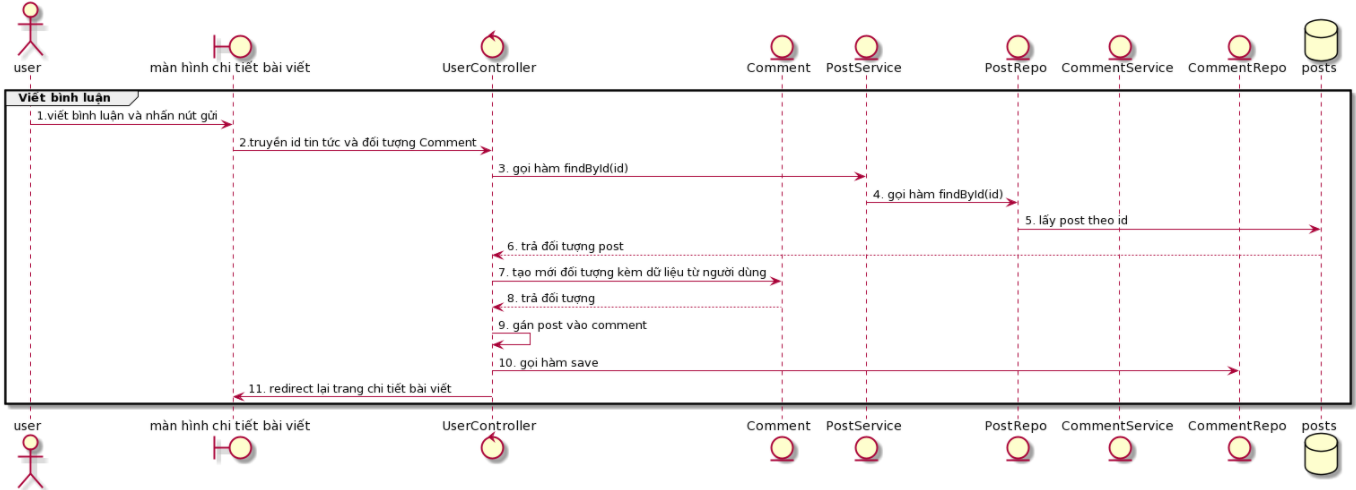
### Comment 1 bài viết:

|  |  |
| --- | --- |
| **Use Case Name** | Comment 1 bài viết |
| **Description** | Khi xem 1 bài viết hoặc tin tức, user có thể viết comment cho bài viết / tin tức đó |
| **Actor(s)** | User |
| **Priority** | Phải có |
| **Trigger** | User muốn viết comment |
| **Pre-Condition(s):** | User đã đăng nhập  User đang ở trang nội dung tin tức / bài viết |
| **Post-Condition(s):** | Comment sẽ được cập nhật vào ds các comment của bài viết / tin tức đó |
| **Basic Flow** | 1. User nhập nội dung comment 2. User nhấn nút submit comment |
| **Alternative Flow** |  |

#### Activity diagram:



#### Sequence diagram:



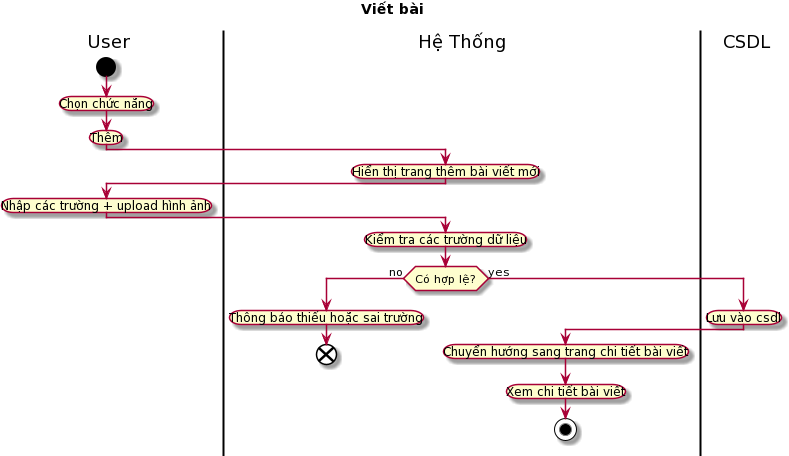
### Lưu bài viết / tin tức:

|  |  |
| --- | --- |
| **Use Case Name** | Lưu 1 bài viết |
| **Description** | Khi xem 1 bài viết hoặc tin tức, user có thể lưu lại bài viết / tin tức đó |
| **Actor(s)** | User |
| **Priority** | Phải có |
| **Trigger** | User muốn viết lưu bài viết / tin tức |
| **Pre-Condition(s):** | User đã đăng nhập |
| **Post-Condition(s):** | Tin tức / bài viết sẽ được cập nhật vào danh sách các tin tức / bài viết đã lưu của user đó |
| **Basic Flow** | 1. User vào trang chi tiết tin tức / bài viết 2. User nhấn nút lưu tin tức / bài viết |
| **Alternative Flow** |  |

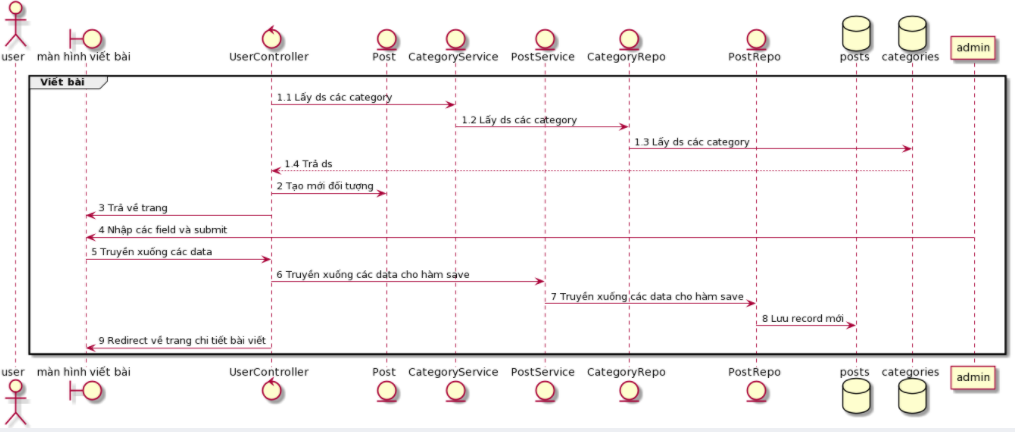
### User viết bài:

|  |  |
| --- | --- |
| **Use Case Name** | Viết bài |
| **Description** | User viết bài viết để đăng lên trang |
| **Actor(s)** | User |
| **Priority** | Phải có |
| **Trigger** | User muốn viết bài viết |
| **Pre-Condition(s):** | User đã đăng nhập |
| **Post-Condition(s):** | Bài viết của user sẽ được cập nhật vào danh sách các bài viết |
| **Basic Flow** | 1. User vào trang viết bài 2. User viết tựa đề / nội dung / chọn tag / upload hình ảnh 3. User nhấn nút submit |
| **Alternative Flow** |  |

#### Activity diagram:



#### Sequence diagram:

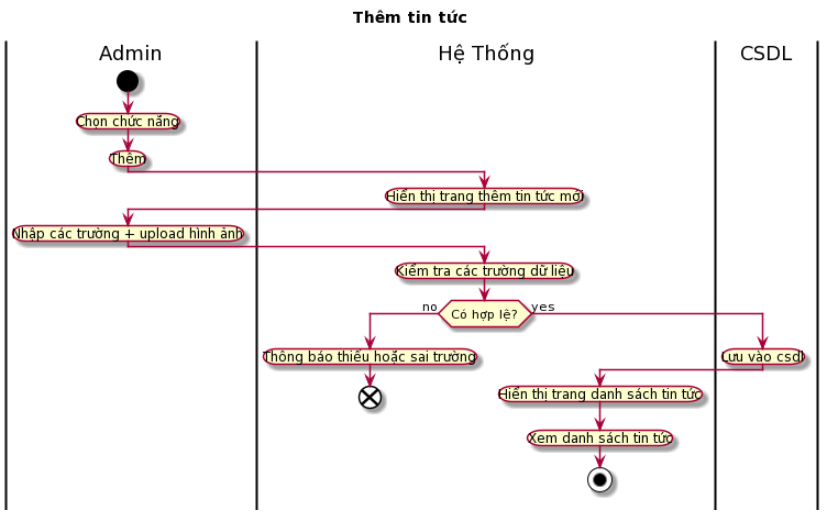


### Quản lí tin tức:

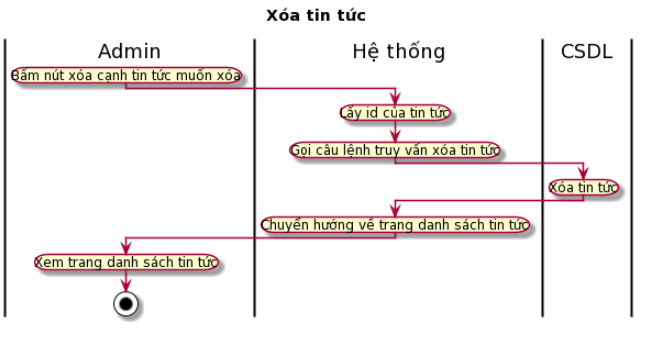
#### Đặc tả

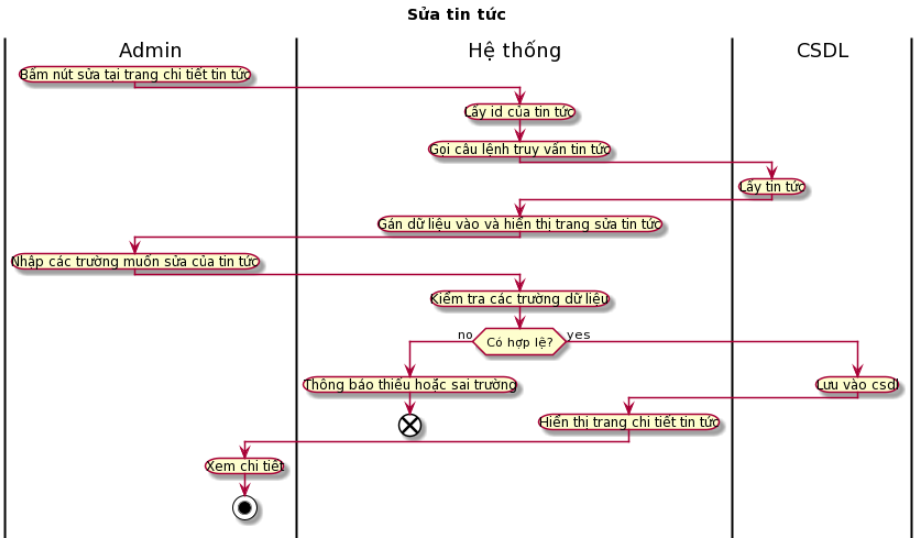
|  |  |
| --- | --- |
| **Use Case Name** | Quản lí tin tức |
| **Description** | Admin xem / thêm / xóa / sửa tin tức |
| **Actor(s)** | Admin |
| **Priority** | Phải có |
| **Trigger** | Admin quản lí tin tức |
| **Pre-Condition(s):** | Admin đã đăng nhập User đang ở trang nội dung tin tức / bài viết |
| **Post-Condition(s):** | Tin tức cập nhật (nếu có) hoặc thêm (nếu có) |
| **Basic Flow** | 1. Admin vào trang quản lý tin tức 2. Admin vào trang viết tin tức 3. Admin viết tựa đề / nội dung / upload hình ảnh / chọn tag 4. Admin nhấn nút submit |
| **Alternative Flow** | 2a. Admin nhìn vào phần danh sách tin tức 3a. Admin nhấn nút xóa cạnh tin tức muốn xóa |
| **Alternative Flow** | 2b. Admin nhìn vào phần danh sách tin tức 3b. Admin nhấn nút sửa cạnh tin tức muốn sửa 4b. Admin vào trang sửa tin tức 5b. Admin sửa tiêu đề / nội dung / tag của tin tức 6b. Admin nhấn submit |

#### Activity Diagram

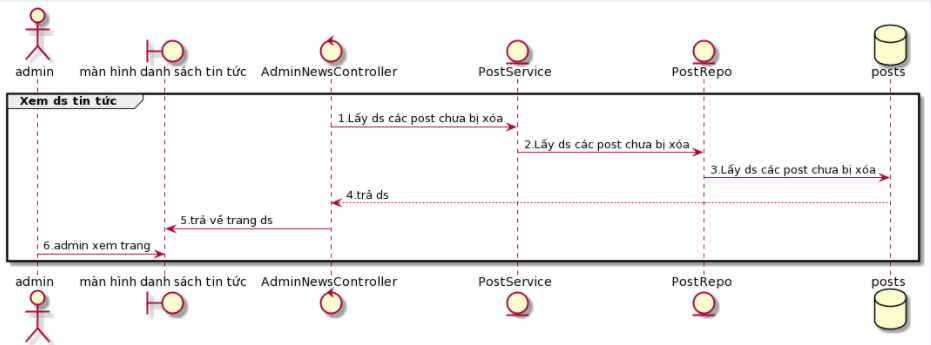


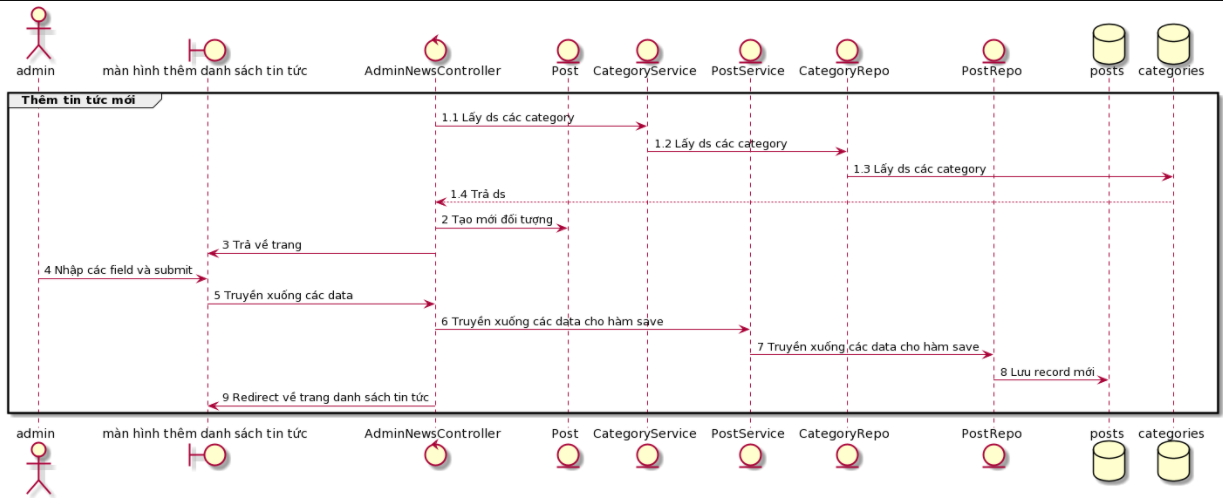


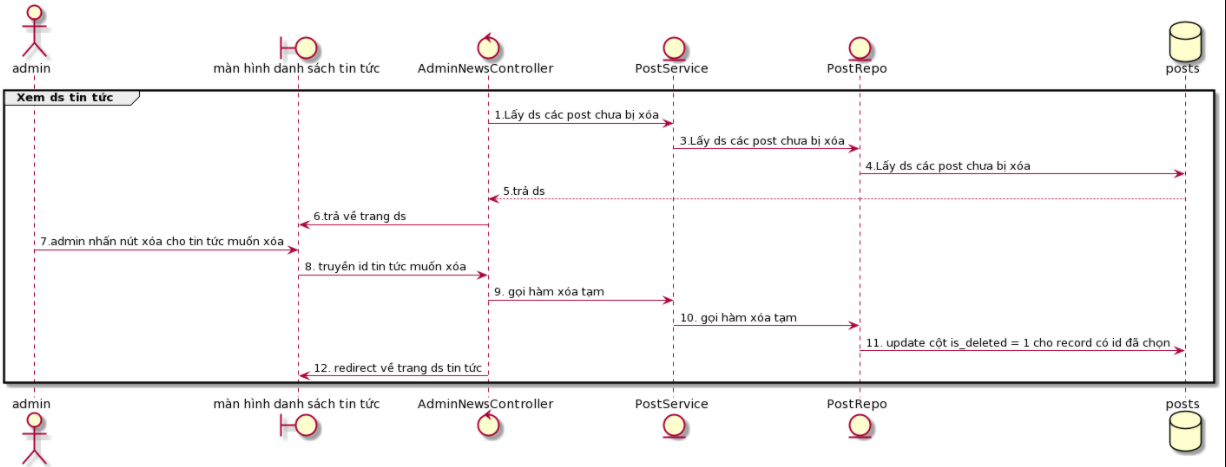


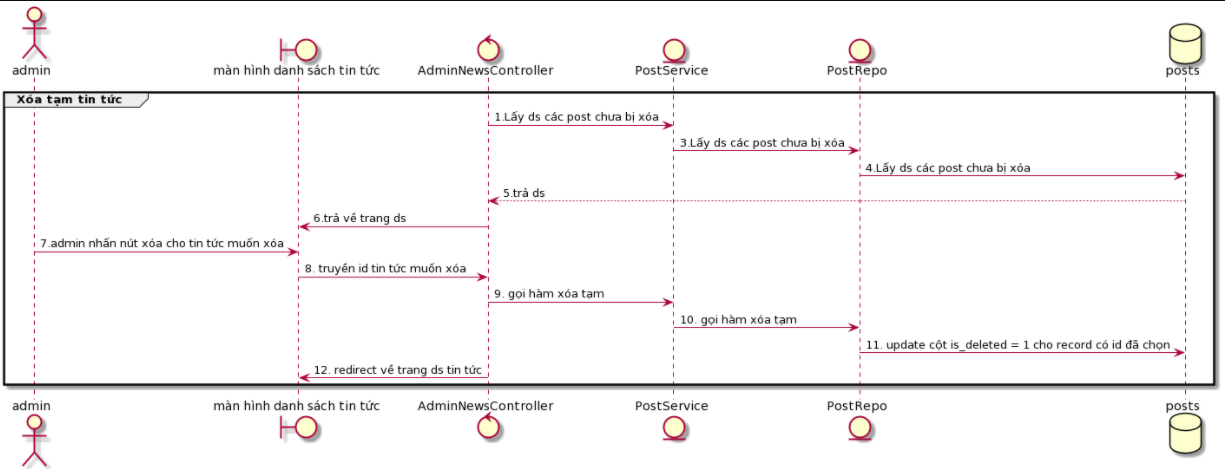


#### Sequence diagram:







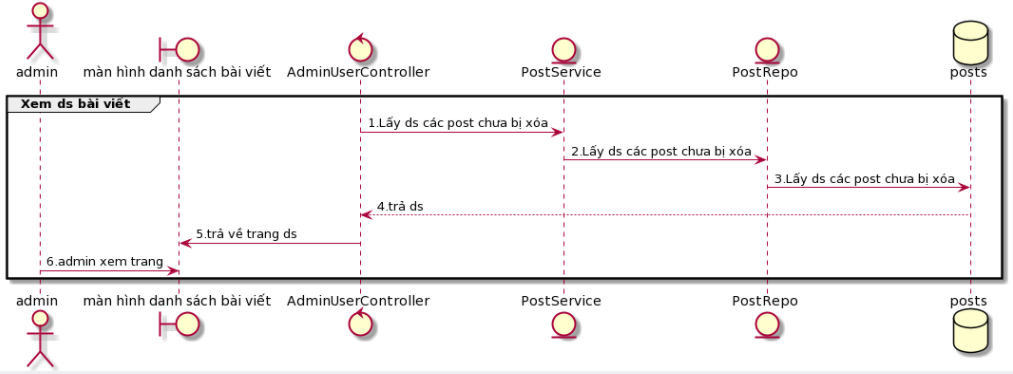


### Quản lí bài viết:

|  |  |
| --- | --- |
| **Use Case Name** | Quản lí bài viết |
| **Description** | Admin xem / xóa / khôi phục bài viết của user |
| **Actor(s)** | Admin |
| **Priority** | Phải có |
| **Trigger** | Admin quản lí bài viết của user |
| **Pre-Condition(s):** | Admin đã đăng nhập User đang ở trang nội dung tin tức / bài viết |
| **Post-Condition(s):** | Bài viết của user được khôi phục / xóa /(nếu có) |
| **Basic Flow** | 1. Admin vào trang quản lý bài viết 2. Admin xem danh sách bài viết 3. Admin nhấn nút xóa bài viết đã chọn |
| **Alternative Flow** | 2a. Admin vào nhấn vào danh mục các bài viết đã xóa 3a. Admin xem danh sách các bài viết đã xóa 4a. Admin nhấn nút khôi phục bài viết đã chọn |

#### Activity diagram:

#### Sequence diagram:



### Quản lý comment:

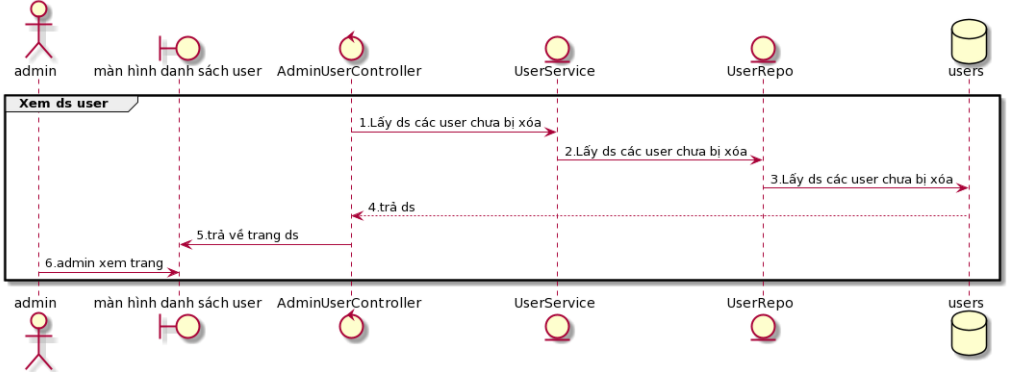
|  |  |
| --- | --- |
| **Use Case Name** | Quản lí comment |
| **Description** | Admin xem / xóa / khôi phục comment của user |
| **Actor(s)** | Admin |
| **Priority** | Phải có |
| **Trigger** | Admin quản lí comment của user |
| **Pre-Condition(s):** | Admin đã đăng nhập User đang ở trang nội dung tin tức / bài viết |
| **Post-Condition(s):** | Comment của user được khôi phục / xóa (nếu có) |
| **Basic Flow** | 1. Admin vào trang quản lý comment 2. Admin xem danh sách comment 3. Admin nhấn nút xóa comment đã chọn |
| **Alternative Flow** | 2a. Admin vào nhấn vào danh mục các comment đã xóa 3a. Admin xem danh sách các comment đã xóa 4a. Admin nhấn nút khôi phục comment đã chọn |

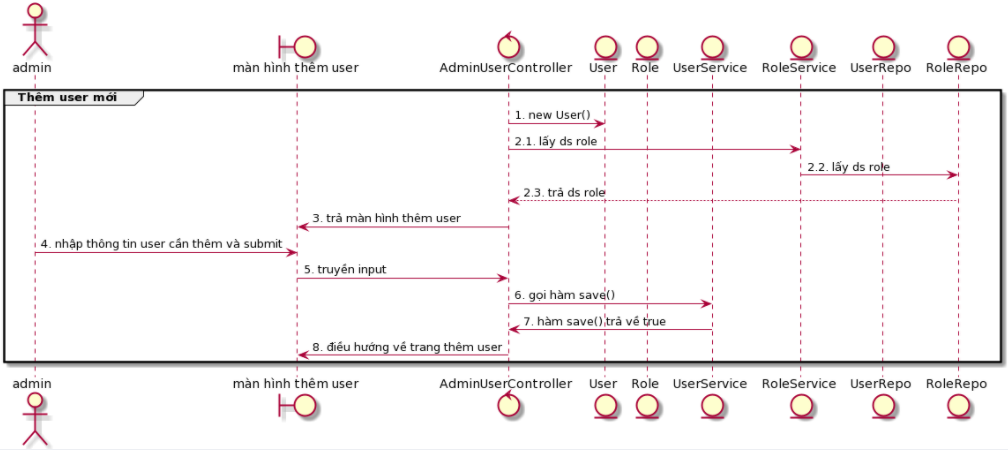
### Quản lý user:

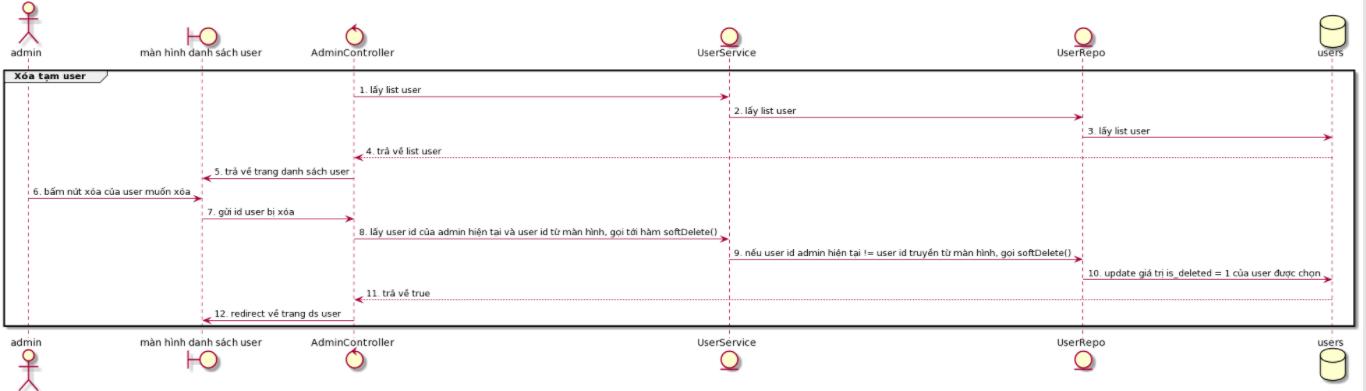
#### Đặc tả

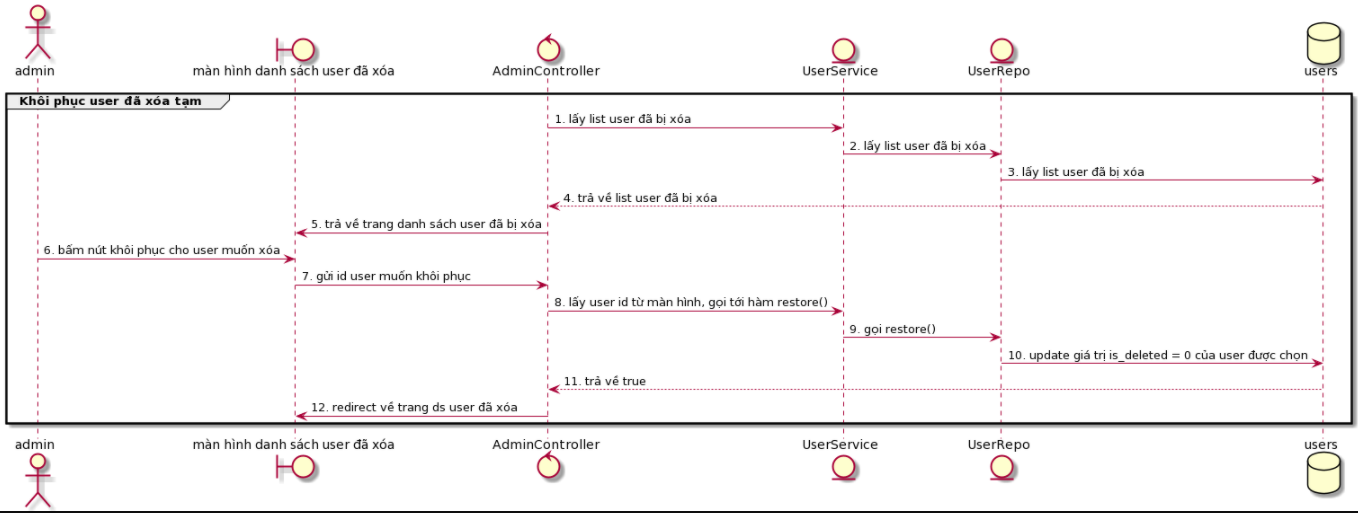
|  |  |
| --- | --- |
| **Use Case Name** | Quản lí user |
| **Description** | Admin xem / xóa / khôi phục user |
| **Actor(s)** | Admin |
| **Priority** | Phải có |
| **Trigger** | Admin quản lí user |
| **Pre-Condition(s):** | Admin đã đăng nhập User đang ở trang nội dung tin tức / bài viết |
| **Post-Condition(s):** | User được khôi phục / xóa (nếu có) |
| **Basic Flow** | 1. Admin vào trang quản lý user 2. Admin xem danh sách user 3. Admin nhấn nút xóa user đã chọn |
| **Alternative Flow** | 2a. Admin vào nhấn vào danh mục các user đã xóa 3a. Admin xem danh sách các user đã xóa 4a. Admin nhấn nút khôi phục user đã chọn |

#### Sequence diagram:









### Admin quản lý tag:

|  |  |
| --- | --- |
| **Use Case Name** | Quản lí tag |
| **Description** | Admin xem / xóa / thêm / khôi phục tag |
| **Actor(s)** | Admin |
| **Priority** | Phải có |
| **Trigger** | Admin quản lí tag |
| **Pre-Condition(s):** | Admin đã đăng nhập User đang ở trang nội dung tin tức / bài viết |
| **Post-Condition(s):** | Tag được thêm / khôi phục / xóa (nếu có) |
| **Basic Flow** | 1. Admin vào trang quản lý tag 2. Admin xem danh sách tag 3. Admin nhấn nút xóa tag đã chọn |
| **Alternative Flow** | 2a. Admin vào nhấn vào danh mục các tag đã xóa 3a. Admin xem danh sách các tag đã xóa 4a. Admin nhấn nút khôi phục tag đã chọn |
| **Alternative Flow** | 2b. Admin vào nhấn vào nút thêm mới tag 3b. Admin vào trang thêm mới tag 4b. Admin chọn loại tag 5b. Admin nhập tên tag mới 6b. Admin nhấn nút thêm mới |

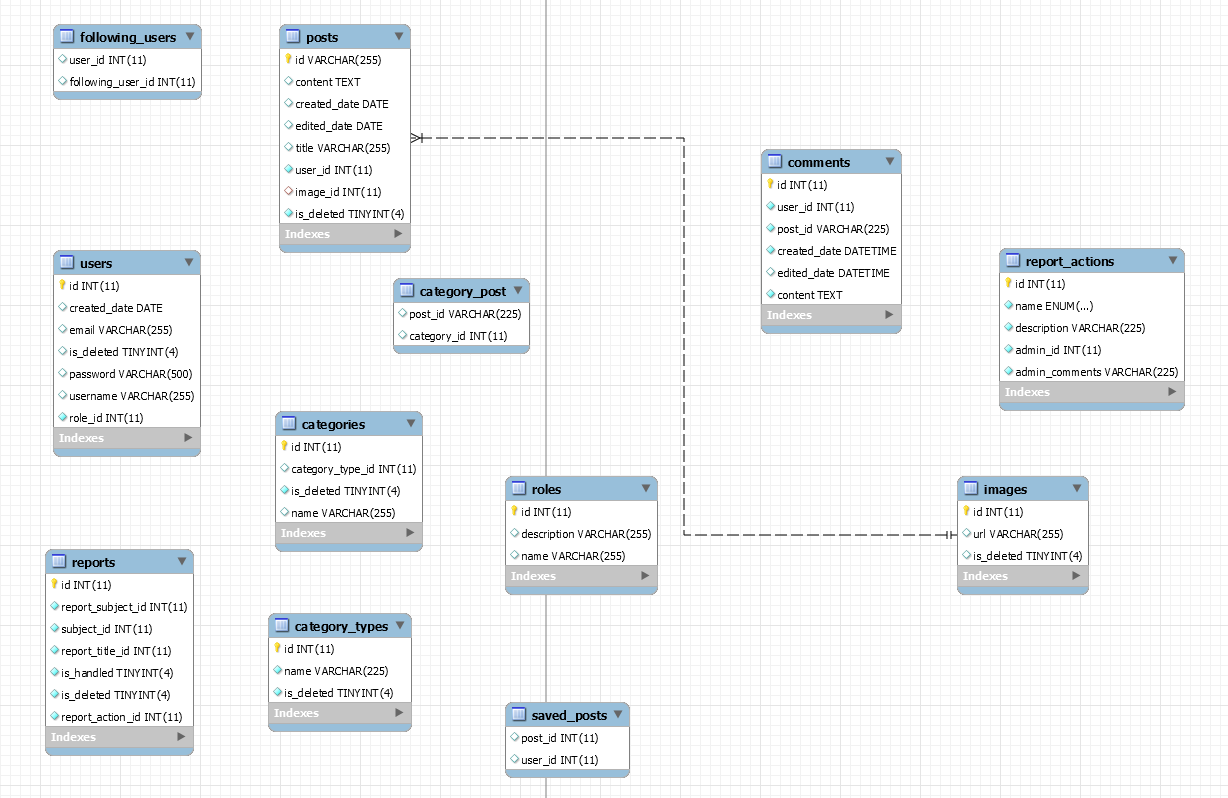
### Quản lý report:

|  |  |
| --- | --- |
| **Use Case Name** | Quản lí report |
| **Description** | Admin quản lý report về bài viết / comment / user |
| **Actor(s)** | Admin |
| **Priority** | Phải có |
| **Trigger** | Admin quản lý report về bài viết / comment / user |
| **Pre-Condition(s):** | Admin đã đăng nhập User đang ở trang nội dung tin tức / bài viết |
| **Post-Condition(s):** | Bài viết / comment / user bị xóa hoặc được bỏ qua |
| **Basic Flow** | 1. Admin vào trang quản lý report 2. Admin xem mục report bài viết 3. Admin xem danh sách các bài viết bị report 4. Admin để lại comment về xử lý 5. Admin chọn nút xóa hoặc bỏ qua |
| **Alternative Flow** | 2a. Admin xem mục report user 3a. Admin xem danh sách các user bị report |
| **Alternative Flow** | 2b. Admin xem mục report comment 3b. Admin xem danh sách các comment bị report |

## SƠ ĐỒ LỚP:

### 

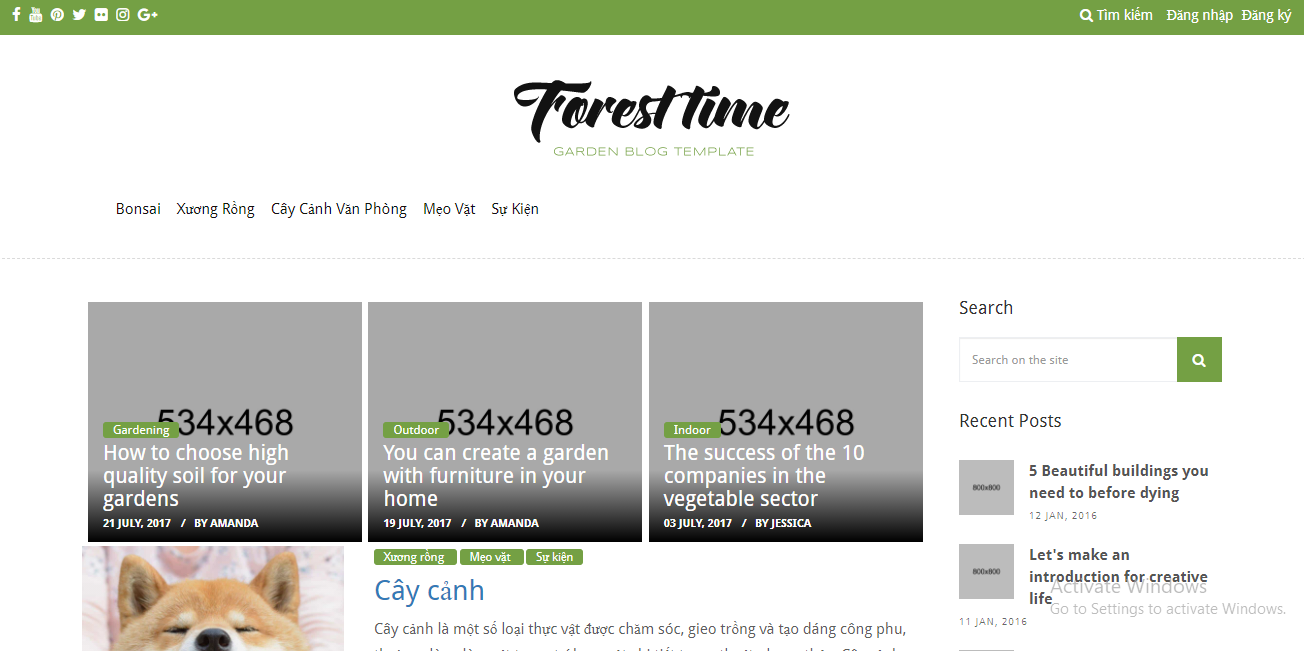
## THIẾT KẾ CSDL:



## GIAO DIỆN:

### Giao diện người dùng:

#### Trang chủ:



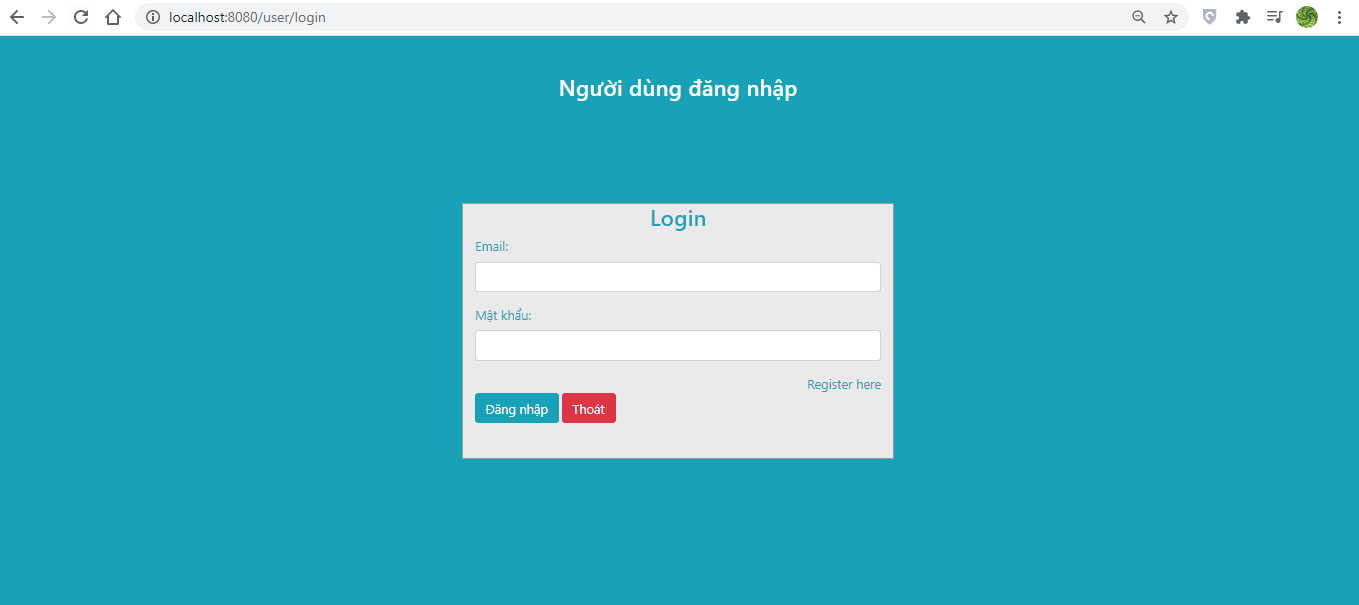
#### Chi tiết bài viết:

Trang chi tiết bài viết

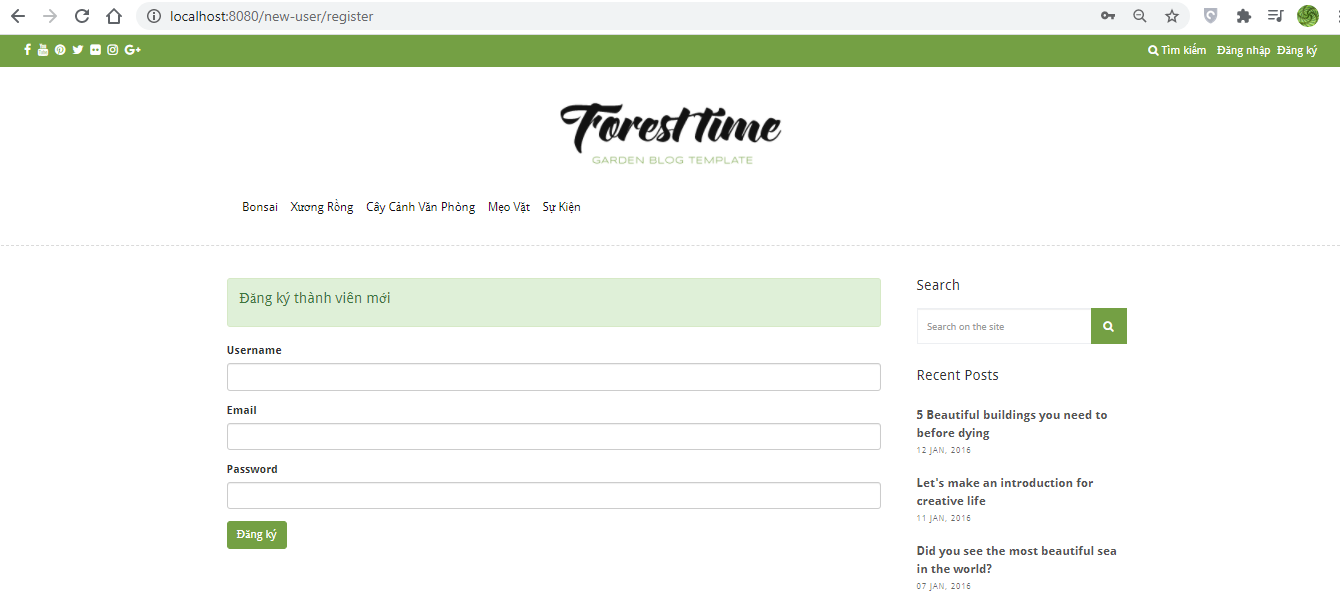
#### 

Mục comment trong chi tiết bài viết

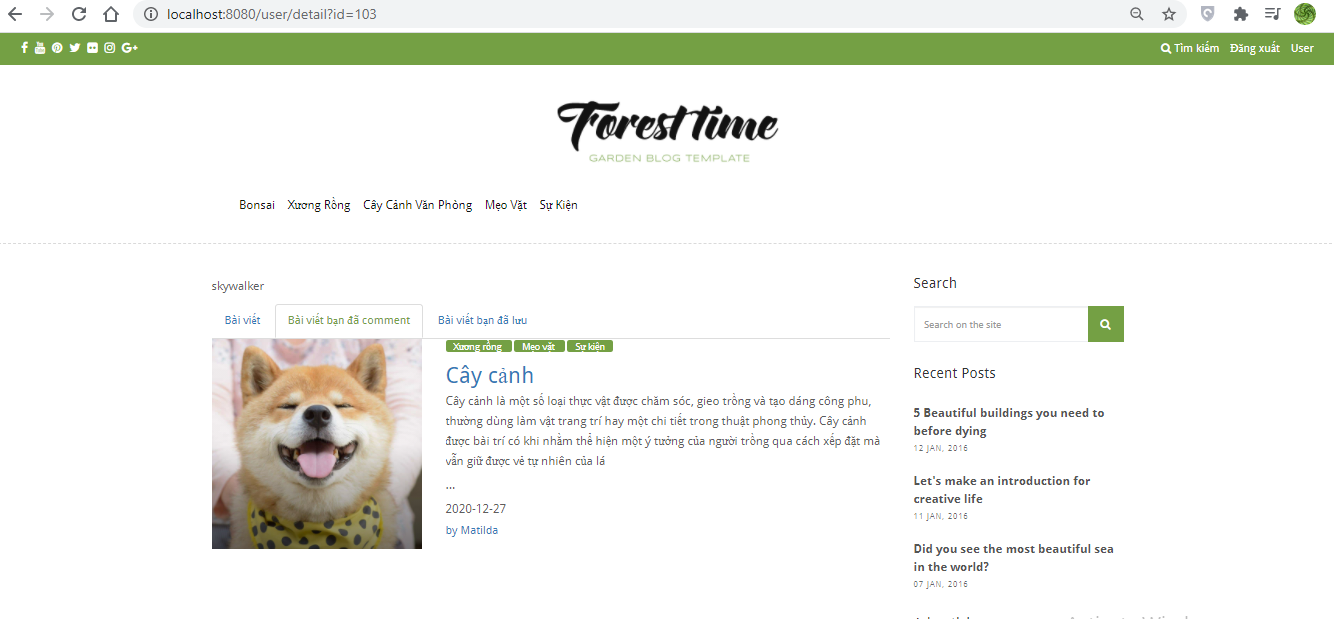
#### Trang đăng nhập:



#### Trang đăng ký tài khoản:



#### Tường nhà user:

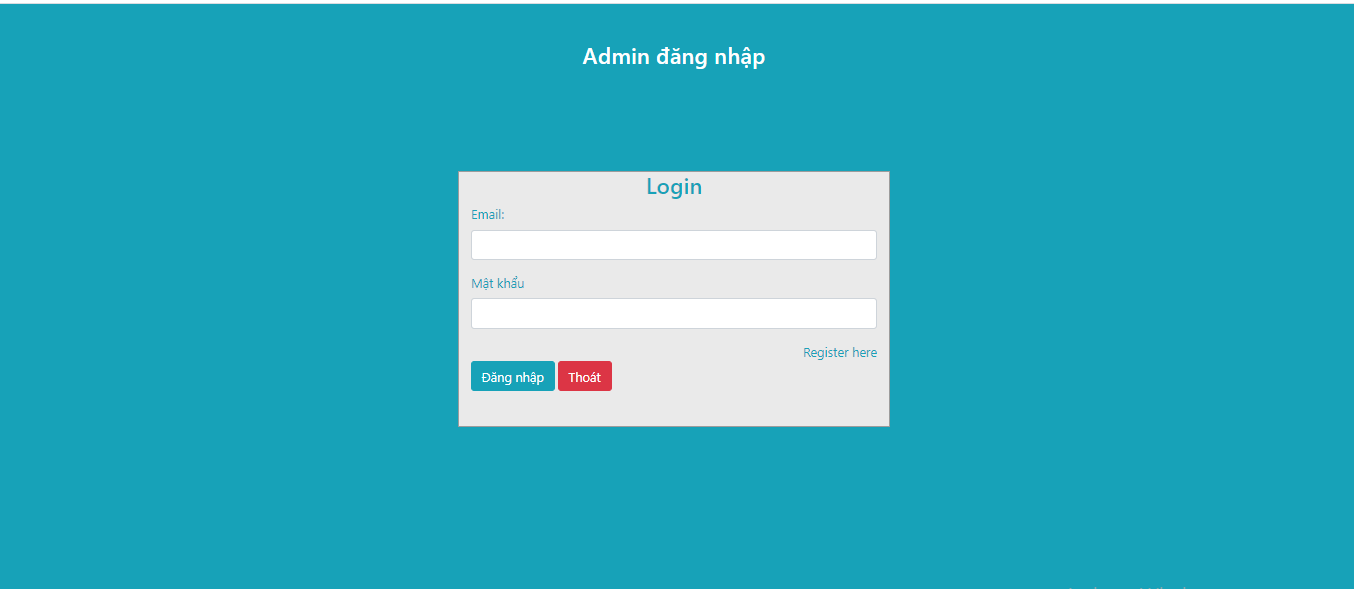


#### Trang viết bài:

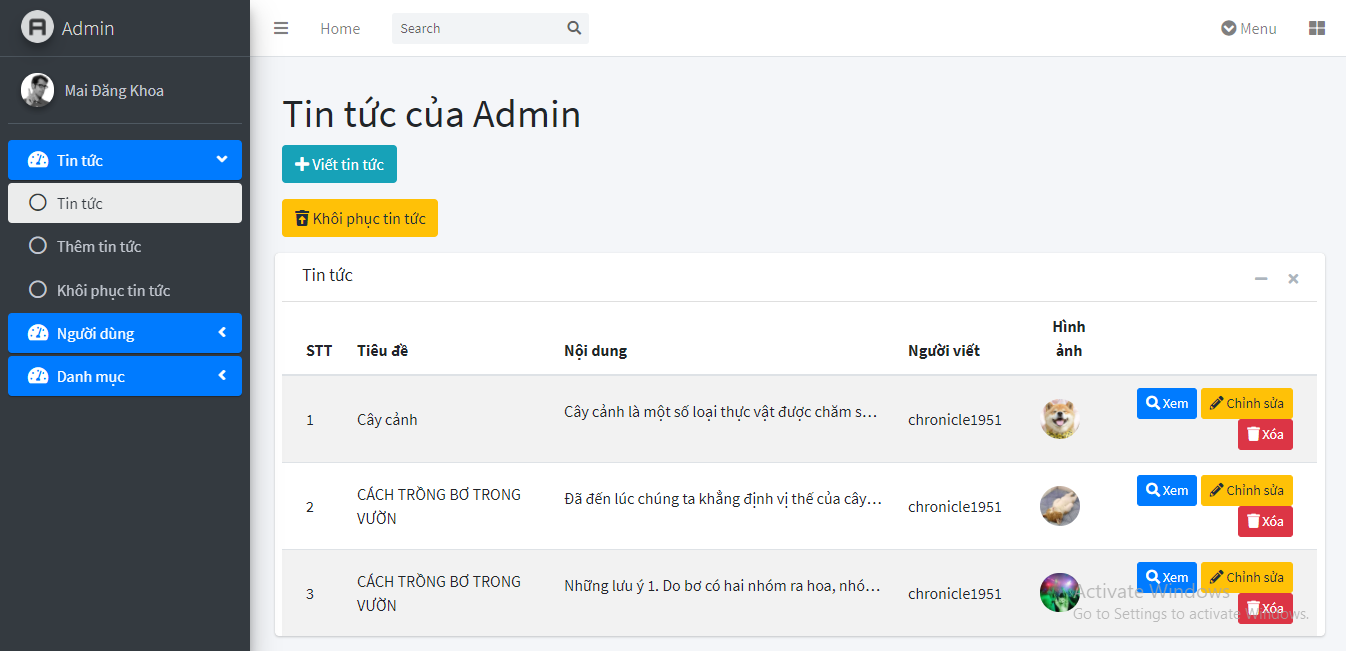


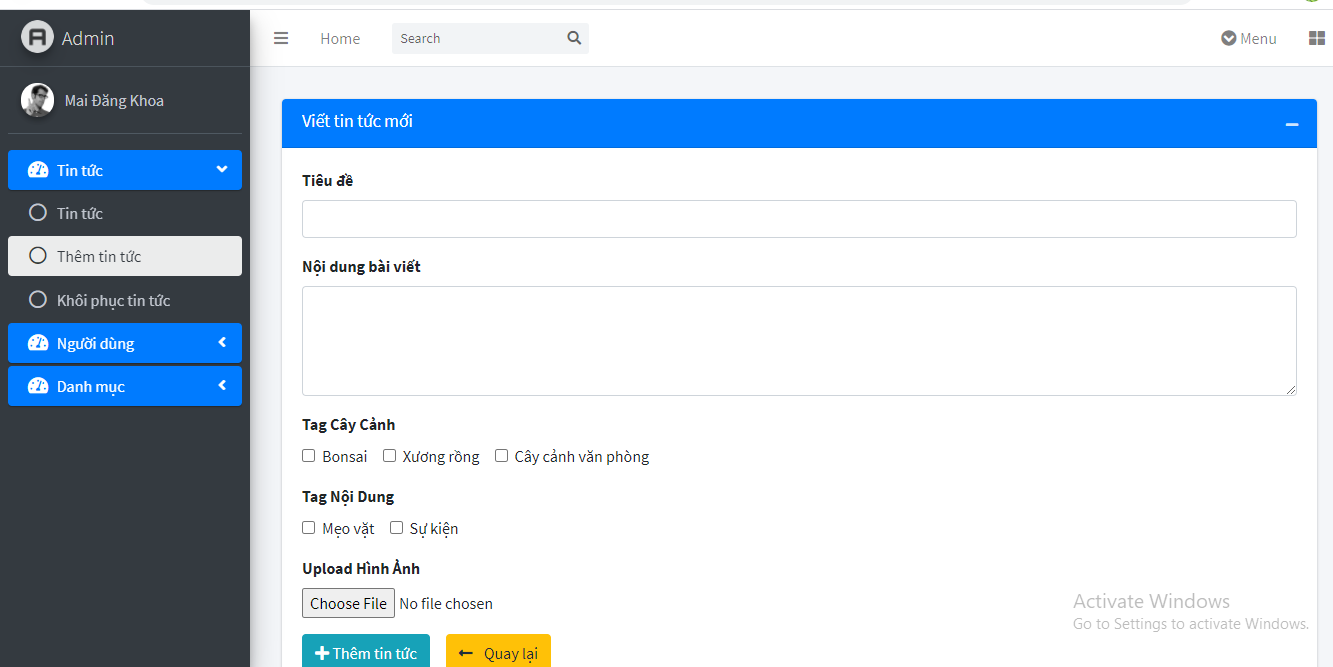
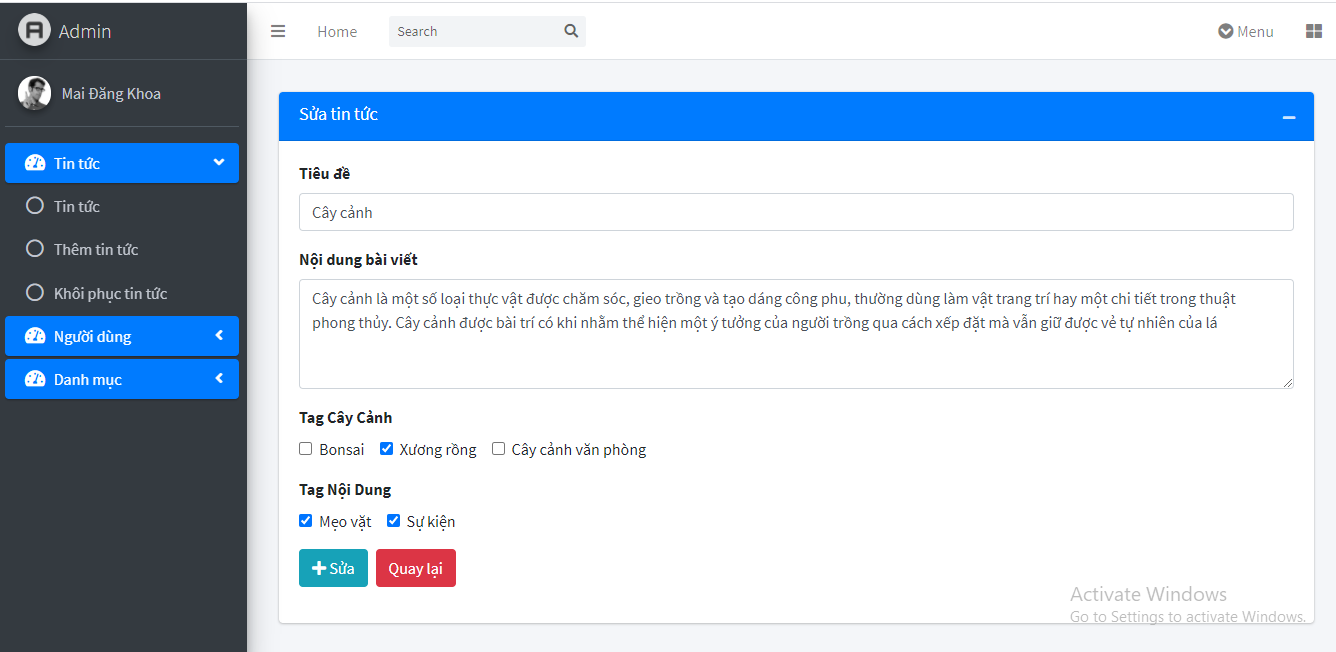
### Giao diện admin:

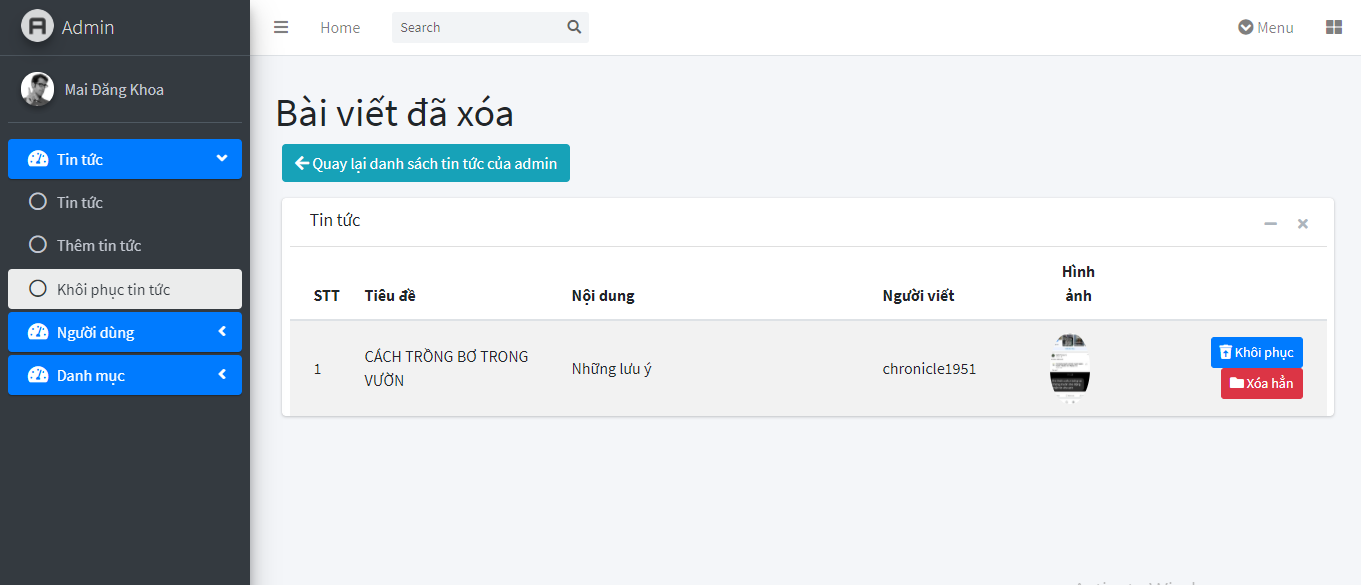
#### Trang đăng nhập:



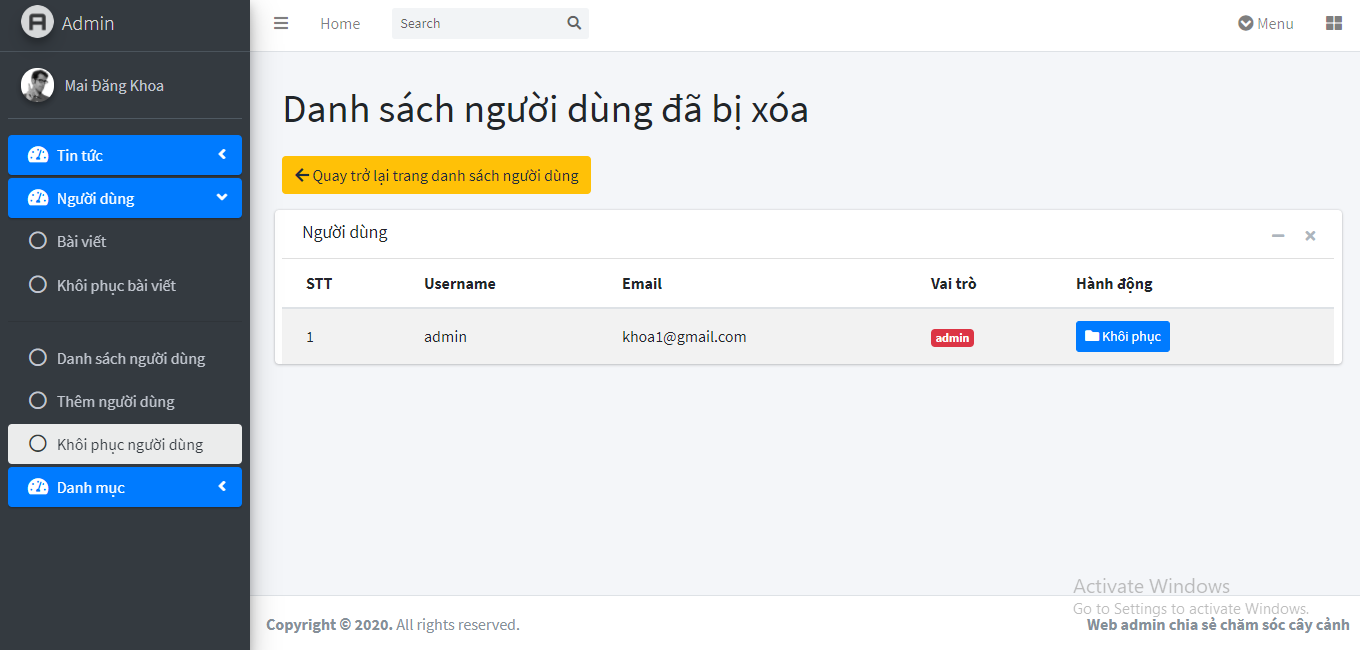
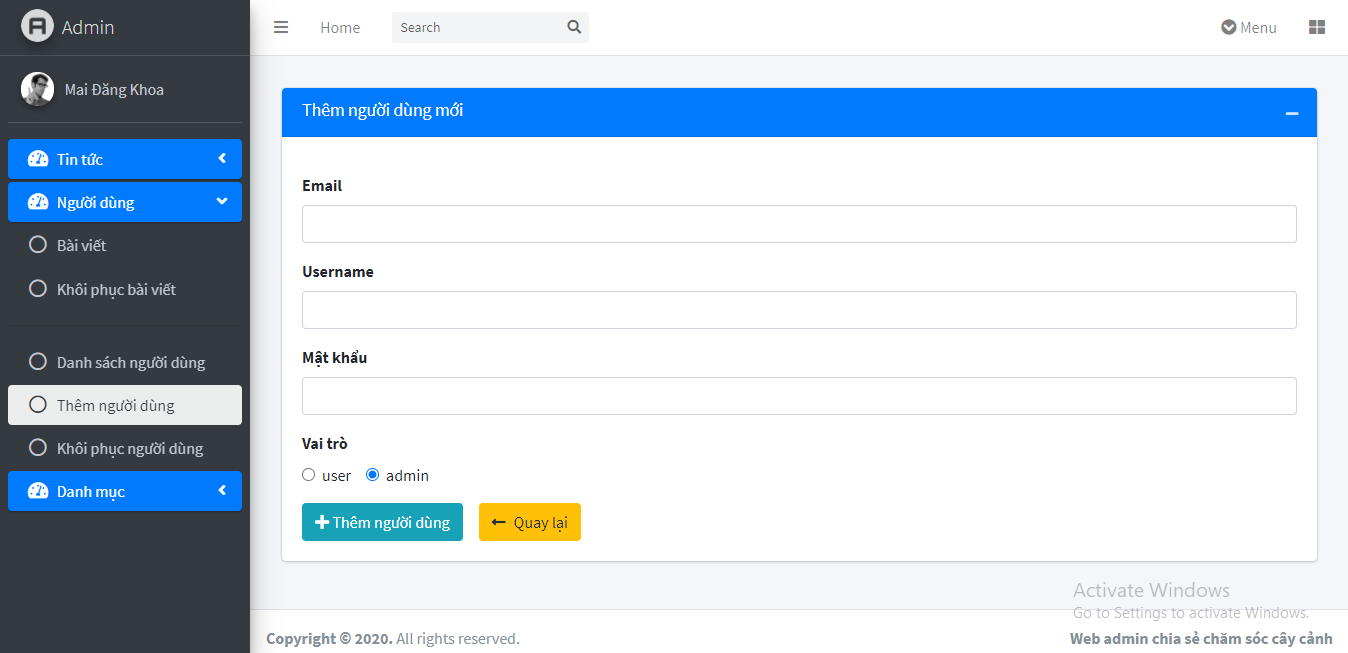
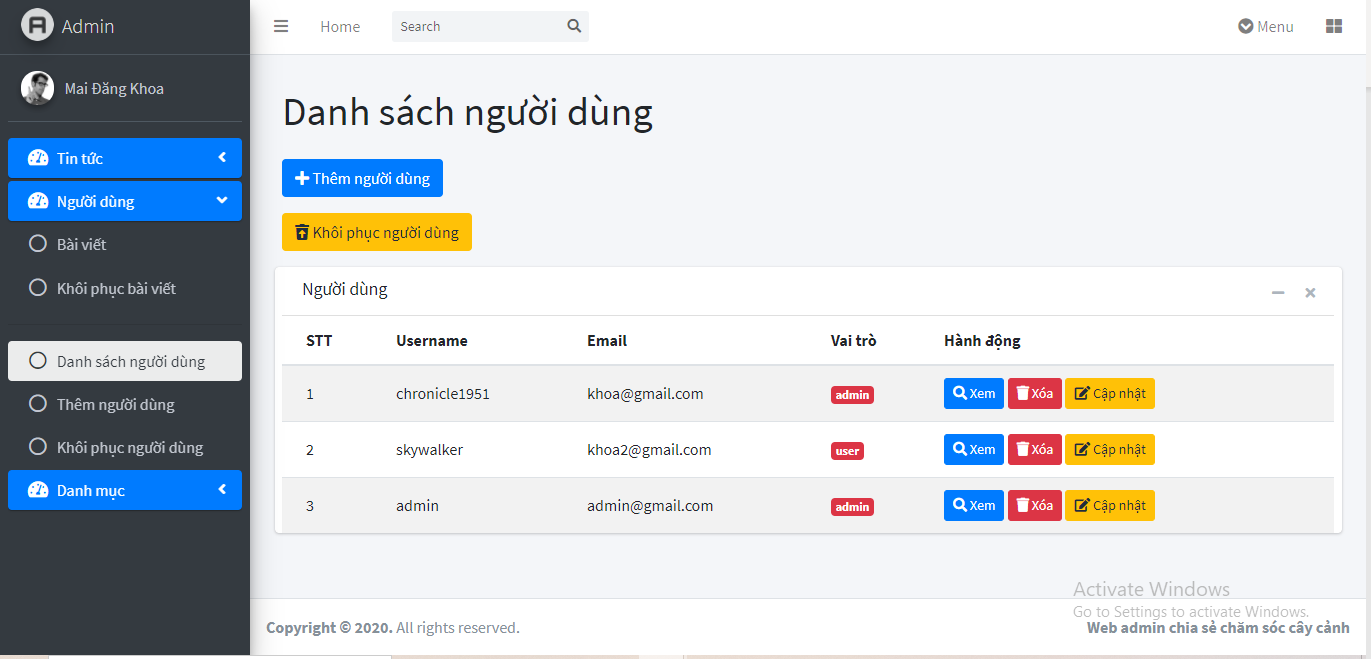
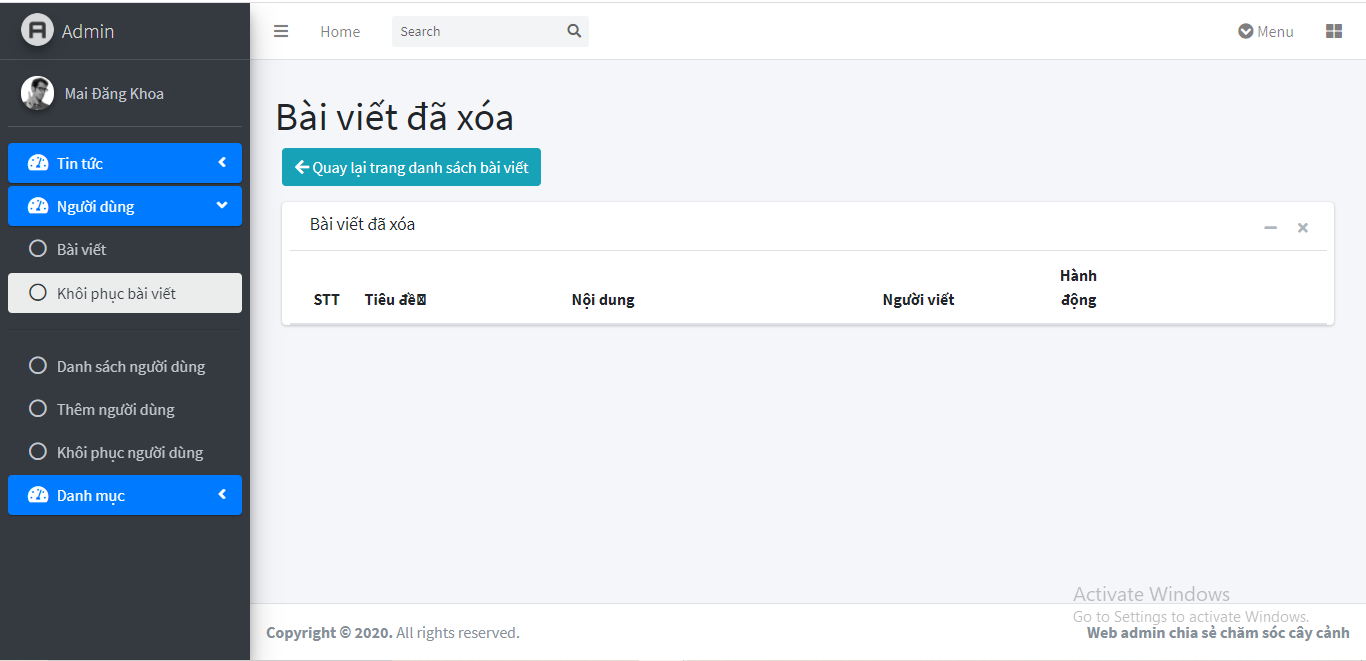
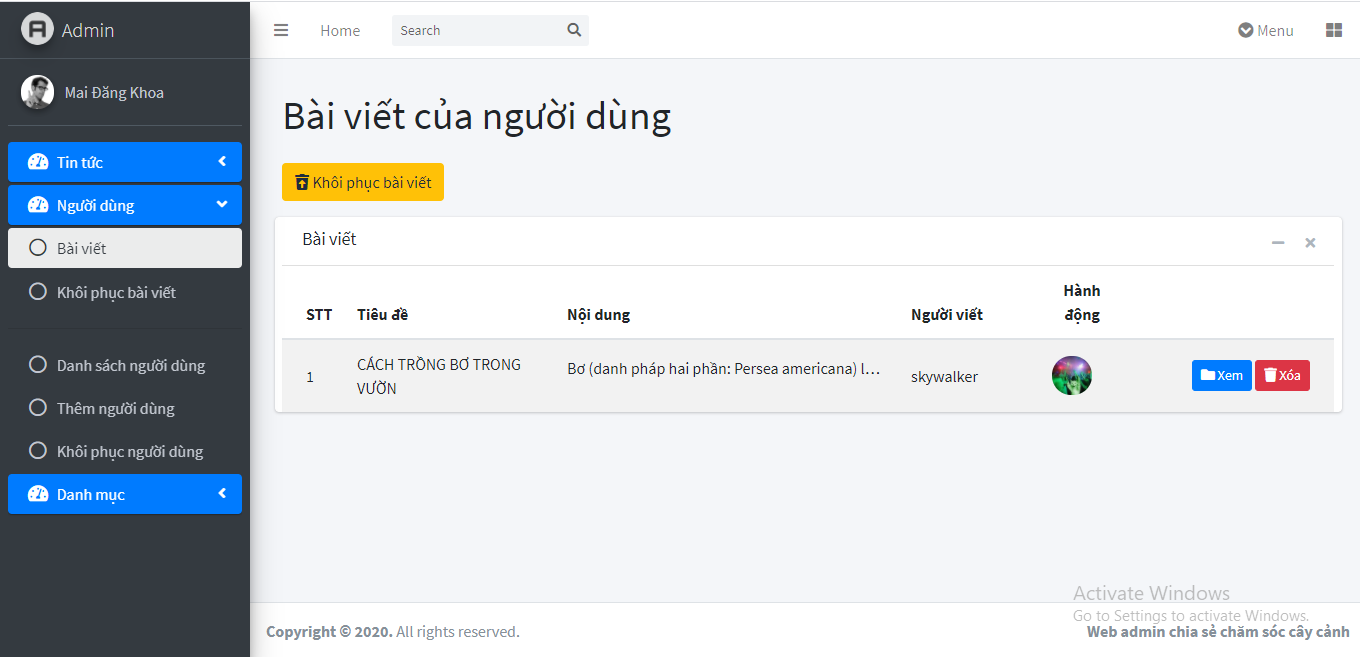
#### Quản lý tin tức:



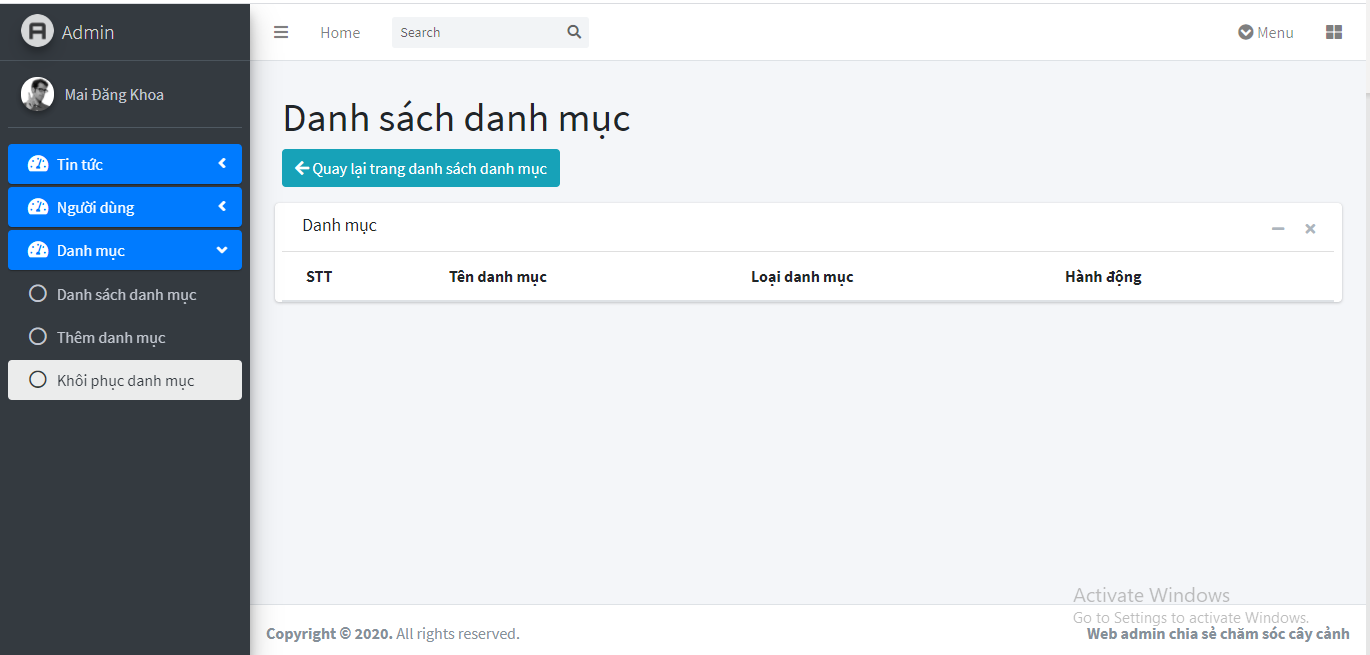




#### Quản lý người dùng:



#### Quản lý danh mục:



# 

# TỔNG QUAN CÔNG NGHỆ

## Spring Boot:

### Khái niệm:

* Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) – Phát triển ứng dụng nhanh.
* Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring.
* Spring Boot không yêu cầu cấu hình XML
* Nó là một chuẩn cho cấu hình thiết kế phần mềm, tăng cao năng suất cho developer.

### Ưu điểm:

* Có các tính năng của Spring Framework.
* Tạo ứng dụng độc lập, có thể chạy bằng java -jar (cho cả java web)
* Nhúng trực tiếp các ứng dụng server (Tomcat, Jetty…) do đó không cần phải triển khai file WAR
* Cấu hình ít, tự động cậu hình bất kì khi nào có thể (Giảm thời gian viết code, tăng năng suất)
* Không yêu cầu XML config…
* Cung cấp nhiều plugin

## Thymeleaf:

### Khái niệm:

Thymeleaf là một Java XML/XHTML/HTML5 Template Engine nó có thể làm việc với cả hai môi trường Web và môi trường không phải Web. Nó phù hợp hơn khi được sử dụng để phục vụ XHTML/HTML5 trên tầng View (View Layer) của ứng dụng Web dựa trên kiến trúc MVC. Nó có thể xử lý bất kỳ một file XML nào, thậm trí trên các môi trường offline (Không trực tuyến). Nó hỗ trợ đầy đủ để tương tác với Spring Framework.

## JPA:

### Khái niệm:

**JPA** là viết tắt của **Java Persistence API**, nó là một đặc tả Java cho việc ánh xạ giữa các đối tượng Java với cơ sở dữ liệu quan hệ sử dụng công nghệ phổ biến là **ORM** (Object Relational Mapping).

JPA cung cấp đầy đủ các công cụ cho phép chúng ta có thể thao tác với cơ sở dữ liệu một cách đơn giản và nhanh chóng. JPA có thể dùng để persist một đối tượng Java (POJO – Plain Old Java Object) vào trong cơ sở dữ liệu hoặc lấy dữ liệu từ cơ sở dữ liệu và ánh xạ (mapping) ra các đối tượng Java một cách đơn giản.

JPA chỉ là một API định nghĩa các đặc tả cần thiết và không có code hiện thực từ những đặc tả đó. Nó chỉ chứa những hướng dẫn để hiện thực ORM. Do đó cần phải có một cài đặt ORM để hoạt động và persist các đối tượng Java. Các ORM Framework có thể sử dụng cho JPA như: Hibernate, iBatis, Eclipse Link, OpenJPA,…

### Ưu điểm:

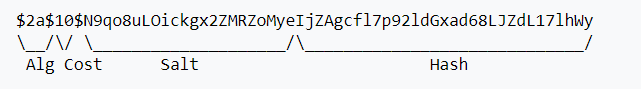
* Viết ít code hơn.
* Performance tốt.
* Độc lập về database.
* Không phải làm việc với SQL.
* Hỗ trợ cấu hình triển khai bằng annotation và xml.
* Có nhiều framework ORM miễn phí hỗ trợ có thể dùng để phát triển nhiều loại ứng dụng khác nhau.
* JPA là một đặc tả đã được chuẩn hóa và là một thành phần trong đặc tả EJB 3.
* Dễ dàng chuyển từ một ORM này sang một ORM khác. Ví dụ từ iBatis sang Hibernate.

## Bcrypt:

### Khái niệm:

Bcrypt là 1 hàm mã hóa mật khẩu thiết kế bởi Niels Provos và David Mazieres, dựa trên thuật toán mã hóa Blowfish.

1 chuỗi sau khi mã hóa bởi Bcrypt sẽ có định dạng:



$2a$: 4 kí tự đầu khai báo phiên bản thuật toán bcrypt

10: cost factor (210 ==> 1,024 rounds), tức thuật toán EksBlowfishSetup sẽ chạy 1024 vòng

9qo8uLOickgx2ZMRZoMye : 16-byte (128-bit) salt, salt là 1 chuỗi random được sinh ra

IjZAgcfl7p92ldGxad68LJZdL17lhWy : mật khẩu sau khi đã hash

Em sử dụng thư viện lấy tại <https://mvnrepository.com/artifact/at.favre.lib/bcrypt/0.2.0> để ứng dụng mã hóa mật khẩu người dùng

### Ưu điểm:

Vì số vòng hash rất nhiều nên sẽ ngăn chặn được việc hack mật khẩu bằng rainbow table nhờ thời gian lớn

Vì là mã hóa 1 chiều nên cách duy nhất so sánh được mật khẩu database và mật khẩu input, đó là dùng salt + mật khẩu input cho ra chuỗi mã hóa và so sánh với database

## AdminLTE

### Khái niệm:

Là 1 template boostrap để nhanh chóng triển khai 1 trang admin

Có chứa rất nhiều dependency và plugin cơ bản của frontend như bootstrap 4, jquery 3.3.1+, chartjs, fontawesome 5, …

## MySQL

### Khái niệm:

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

# 

# KẾT LUẬN, HƯỚNG PHÁT TRIỂN

## Ưu điểm:

− Hoàn thành yêu cầu đặt ra ban đầu, có đầy đủ tính năng cơ bản cần thiết của một ứng dụng chia sẻ chăm sóc cây cảnh.

− Giao diện đơn giản, bố cục hợp lý, người dùng dễ dàng theo tác trên ứng dụng.

− Sinh viên thực hiện đã nắm được cách thức làm một ứng dụng mạng xã hội thu nhỏ,

hiểu biết về ngôn ngữ Java, công nghệ Spring boot, Thymeleaf.

− Ứng dụng đã được triển khai trên local.

## Nhược điểm:

− Còn hạn chế một số tính năng:

+ Chưa tích hợp được chức năng báo cáo comment

+ Chưa phát triển được nhiều phần động cho trang web

+ Tối ưu code chưa tốt nên còn nhiều chỗ có thể phát triển trong tương lai.

## Hướng phát triển:

− Hoàn thiện thêm nhiều tính năng đặt ra ban đầu.

− Phát triển thành server api phục vụ cho nhiều sản phẩm về sau, không chỉ là web.