**BỘ LAO ĐỘNG - THƯƠNG BINH VÀ XÃ HỘI**

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ BÁCH KHOA HÀ NỘI**



**BÁO CÁO THỰC TẬP CHUYÊN ĐỀ**

**THIẾT KẾ BLOG ẨM THỰC**

**SV (nhóm) thực hiện:**Nguyễn Quang Tùng

**MSSV:2309620034**

**Lớp:2623CNT01**

**GVHD:Trần Thị Huê**

**Hà Nội, 4 tháng 1 năm 2025**



**BÁO CÁO THỰC TẬP CHUYÊN ĐỀ**

**THIẾT KẾ BLOG ẨM THỰC**

**Hà Nội, 4 tháng 1 năm 2025**

**BỘ LAO ĐỘNG - THƯƠNG BINH VÀ XÃ HỘI**

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ BÁCH KHOA HÀ NỘI**

**KẾ HOẠCH THỰC TẬP**

Họ và tên SV:Nguyễn Quang Tùng Lớp-khóa: K23

MSSV:2309620034

Số ĐT:0342493715 Email:tungfck0987@gmail.com

**Danh sách thành viên nhóm:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Họ và tên** | **Điện thoại** | **Ghi chú** |
| 1 | Nguyễn Quang Tùng | 0342493715 |  |
|  |  |  |  |
|  |  |  |  |

**1. Nội dung thực tập: Thiết kế blog ẩm thực**

**2. Nhiệm vụ cá nhân:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Họ và tên** | **Nhiệm vụ** | **Ghi chú** |
| 1 | Nguyễn Quang Tùng | Phụ trách toàn bộ dự án |  |
|  |  |  |  |
|  |  |  |  |

**3. Kế hoạch chi tiết**

|  |  |  |  |
| --- | --- | --- | --- |
| **Buổi/ngày thực hiện** | **Công việc** | **Kết quả đạt được *(theo tỷ lệ %)*** | **Ghi chú** |
| B.1/ | Phân tích yêu cầu |  |  |
| B.2/ | Thiết kế hệ thống |  |  |
| B.3/ | Phát triển |  |  |
| B.4/ | Kiểm thử |  |  |
| B.5/ | Triển khai |  |  |
| B.6/ |  |  |  |
|  |  |  |  |

*Hà Nội, ngày tháng năm 20*

**Xác nhận của GV hướng dẫn Trưởng nhóm**

*(ký, ghi rõ họ và tên) (ký, ghi rõ họ và tên)*

# **LỜI NÓI ĐẦU**

Trong thời đại công nghệ phát triển mạnh mẽ, nhu cầu kết nối, chia sẻ và tìm kiếm thông tin qua các nền tảng trực tuyến ngày càng gia tăng. Lĩnh vực ẩm thực cũng không ngoại lệ, khi ngày càng có nhiều người tìm kiếm các công thức nấu ăn độc đáo, những chia sẻ hữu ích từ cộng đồng yêu ẩm thực, cũng như cơ hội để giới thiệu bản thân và xây dựng thương hiệu cá nhân trong lĩnh vực này. Hiểu được tầm quan trọng của việc đáp ứng những nhu cầu trên, dự án "Blog Ẩm Thực" được ra đời với mục tiêu xây dựng một không gian trực tuyến chuyên nghiệp và thân thiện, nơi hội tụ những giá trị văn hóa ẩm thực và kết nối cộng đồng.

Dự án không chỉ đơn thuần là một blog mà còn là một nền tảng đa dạng chức năng, cung cấp cơ hội để người dùng chia sẻ kiến thức, tìm kiếm công thức mới, khám phá các trải nghiệm ẩm thực độc đáo, và tương tác với những người cùng đam mê. Bằng việc áp dụng các công nghệ hiện đại cùng quy trình phát triển bài bản, dự án hướng đến việc trở thành một trong những nền tảng đáng tin cậy nhất cho cộng đồng yêu ẩm thực.

Với sự nỗ lực không ngừng, chúng tôi cam kết mang lại một sản phẩm chất lượng, đáp ứng các tiêu chuẩn khắt khe về bảo mật, hiệu suất và trải nghiệm người dùng. Trong tài liệu này, bạn sẽ tìm thấy chi tiết về kế hoạch phân tích, thiết kế và triển khai dự án "Blog Ẩm Thực", được thực hiện bởi một đội ngũ đầy đam mê và tận tâm, hoặc trong trường hợp này, một cá nhân duy nhất đảm nhiệm tất cả các vai trò: Nguyễn Quang Tùng.

# **CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN**

## Đặt vấn đề

Trong thời đại số hóa, việc tìm kiếm thông tin và chia sẻ kiến thức ngày càng trở nên phổ biến. Tuy nhiên, không phải tất cả các nền tảng trực tuyến đều đáp ứng tốt nhu cầu của cộng đồng người dùng. Đặc biệt trong lĩnh vực ẩm thực, mặc dù có rất nhiều trang web chia sẻ công thức nấu ăn, nhưng đa số lại không mang đến sự kết nối giữa người dùng và thiếu các công cụ để hỗ trợ phát triển thương hiệu cá nhân hoặc tổ chức. Điều này đặt ra nhu cầu cần thiết cho một hệ thống blog ẩm thực hoàn chỉnh, nơi tích hợp cả việc chia sẻ, học hỏi và giao lưu trong một nền tảng duy nhất. "Blog Ẩm Thực" ra đời nhằm giải quyết các vấn đề trên, với mục tiêu không chỉ đáp ứng nhu cầu cơ bản mà còn mở rộng giá trị tương tác, kết nối và phát triển cộng đồng.

## Mô tả bài toán

### **2.1 Khảo sát hệ thống**

**Bước 1**: Người dùng đăng ký tài khoản để sử dụng các tính năng của blog ẩm thực, bao gồm việc chia sẻ, tìm kiếm và lưu trữ công thức nấu ăn.

**Bước 2**: Người dùng đăng nhập vào hệ thống để truy cập vào các công thức nấu ăn yêu thích, xem thông tin chi tiết và đăng tải công thức của mình.

**Bước 3**: Người dùng có thể tìm kiếm, đánh giá và bình luận về các công thức, đồng thời lưu các công thức vào danh sách yêu thích để sử dụng sau.

**Vấn đề**: Khi blog có lượng người dùng lớn, việc quản lý thông tin khách hàng, công thức và các tương tác sẽ tốn rất nhiều thời gian và công sức nếu làm thủ công. Hệ thống truyền thống có thể gây ra khó khăn trong việc lưu trữ và tra cứu dữ liệu.

**2.2 Phát biểu bài toán**

“Hệ thống quản lý blog ẩm thực cung cấp cho người quản lý và người sử dụng các chức năng cần thiết để theo dõi và quản lý thông tin người dùng, các công thức nấu ăn, lịch sử tìm kiếm và tương tác của người dùng với các món ăn. Các chức năng của hệ thống bao gồm:

* **Đăng ký và đăng nhập**: Người dùng cần tạo tài khoản để đăng nhập vào hệ thống. Sau khi đăng nhập thành công, người dùng có thể xem thông tin cá nhân của mình (tên, email, danh sách công thức yêu thích, lịch sử tìm kiếm, v.v.).
* **Công thức nấu ăn**: Các công thức nấu ăn sẽ được người dùng đăng tải lên hệ thống, bao gồm tên món ăn, nguyên liệu, hướng dẫn chế biến, và hình ảnh hoặc video minh họa. Các công thức này có thể được phân loại theo thể loại món ăn, nguyên liệu chính, thời gian chế biến, v.v.
* **Tìm kiếm và lọc công thức**: Người dùng có thể tìm kiếm công thức theo các tiêu chí khác nhau (theo tên món ăn, nguyên liệu, thời gian, độ khó, v.v.), giúp họ nhanh chóng tìm được món ăn phù hợp.
* **Đánh giá và bình luận**: Người dùng có thể đánh giá các công thức và để lại bình luận về món ăn, chia sẻ kinh nghiệm nấu nướng hoặc mẹo chế biến.
* **Danh sách yêu thích**: Người dùng có thể lưu lại các công thức yêu thích vào tài khoản của mình để dễ dàng truy cập và sử dụng sau này.
* **Thông tin người dùng**: Hệ thống lưu trữ thông tin của người dùng như tên, email, sở thích ăn uống, và công thức đã lưu. Người dùng có thể chỉnh sửa thông tin cá nhân khi cần.

### **Hạn chế của việc quản lí truyền thống**

* **Chi phí lưu trữ cao**: Việc quản lý thông tin công thức nấu ăn và người dùng bằng giấy tờ hoặc các phương pháp truyền thống sẽ yêu cầu một lượng lớn giấy tờ và tài liệu, gây tốn kém về chi phí lưu trữ và bảo mật.
* **Khó khăn trong việc tra cứu và cập nhật**: Việc tìm kiếm công thức nấu ăn hoặc chỉnh sửa thông tin sẽ mất nhiều thời gian và công sức nếu thực hiện thủ công. Khó khăn trong việc duy trì tính chính xác và cập nhật thông tin sẽ ảnh hưởng đến trải nghiệm người dùng.
* **Thiếu tính linh hoạt và hiệu quả**: Việc quản lý hồ sơ người dùng và công thức nấu ăn bằng phương pháp truyền thống có thể dẫn đến tình trạng quá tải khi số lượng người dùng và công thức ngày càng tăng.

## Đề xuất giải pháp

Khi phát triển hệ thống quản lý blog ẩm thực trên môi trường internet, người dùng có thể dễ dàng tra cứu và chia sẻ công thức nấu ăn, đồng thời theo dõi các công thức yêu thích và đánh giá món ăn.

**Xuất phát điểm** với sự phát triển của công nghệ và số lượng người dùng internet ngày càng tăng, việc chia sẻ và tìm kiếm công thức nấu ăn trở nên thuận tiện và nhanh chóng. Sự phổ biến của các thiết bị di động giúp người dùng có thể truy cập, chia sẻ và lưu trữ công thức mọi lúc, mọi nơi.

## Giải pháp dự kiến

### **4.2 Cách sử dụng hệ thống**

**4.1 Mô tả**

* Người dùng muốn xem, chia sẻ công thức, đánh giá hoặc chỉnh sửa thông tin về công thức nấu ăn.
* Hệ thống sẽ tiếp nhận, xử lý và lưu trữ thông tin các công thức nấu ăn, đánh giá và tương tác của người dùng.

**4.2 Cách sử dụng hệ thống**

**Khách hàng (Người dùng):**

* Truy cập vào hệ thống.
* Đăng ký tài khoản (bỏ qua nếu đã có).
* Đăng nhập vào hệ thống.
* Tìm kiếm và xem các công thức nấu ăn.
* Đánh giá và bình luận về các công thức.
* Lưu các công thức yêu thích vào danh sách cá nhân.

**Nhân viên, quản lý:**

* Đăng nhập vào trang quản trị bằng tài khoản admin.
* Kiểm tra, duyệt và quản lý các công thức nấu ăn đăng tải.
* Lưu trữ và chỉnh sửa thông tin công thức nếu cần thiết.

**Blogger/Chuyên gia ẩm thực:**

* Đăng nhập vào hệ thống bằng tài khoản chuyên gia.
* Chia sẻ công thức nấu ăn mới, cập nhật nội dung và tương tác với người dùng.

**4.3 Lợi ích của hệ thống**

* **Hoạt động liên tục**: Hệ thống blog ẩm thực hoạt động 24/7, giúp người dùng dễ dàng tìm kiếm, chia sẻ và lưu trữ công thức nấu ăn mọi lúc, mọi nơi mà không cần phải di chuyển.
* **Tối ưu nhân lực**: Hệ thống giúp giảm thiểu nhân lực cần thiết để quản lý nội dung, đồng thời cho phép tăng cường nhân lực tại các bộ phận cần sự hỗ trợ trong quá trình duyệt và chỉnh sửa công thức.
* **Lưu trữ dữ liệu lâu dài**: Các công thức nấu ăn và thông tin người dùng được lưu trữ vĩnh viễn trên hệ thống, giúp dễ dàng truy xuất, quản lý và phục vụ cho việc nghiên cứu, phân tích hoặc cập nhật trong tương lai.

# **CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG**

## Biểu đồ ca sử dụng

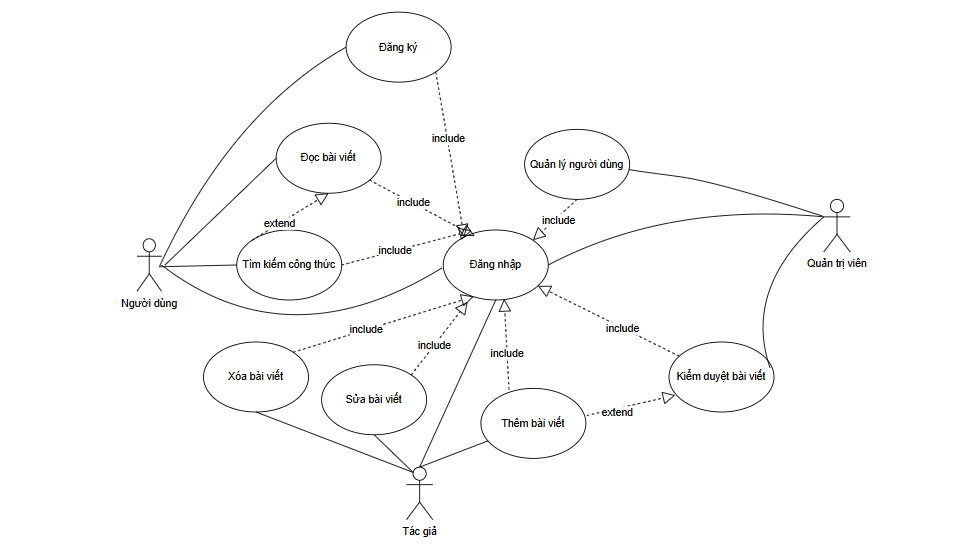
### **1.1 Xác định các actor**

|  |  |
| --- | --- |
| * Người dùng * Tác giả | * admin |

### **1.2 Xác định use case**

|  |  |
| --- | --- |
| * Người dùng * **Đăng ký** * **Đăng nhập** * **Tìm kiếm công thức** * **Đọc bài viết** * Tác giả * **Đăng nhập** * **Thêm bài viết** * **Sửa bài viết** * **Xóa bài** | * Quản trị viên * Đăng nhập * Quản lý người dùng * Kiểm duyệt bài viết |

### **1.3 Biểu đồ use case**

****

### **1.4 Biểu đồ phân rã**

#### 

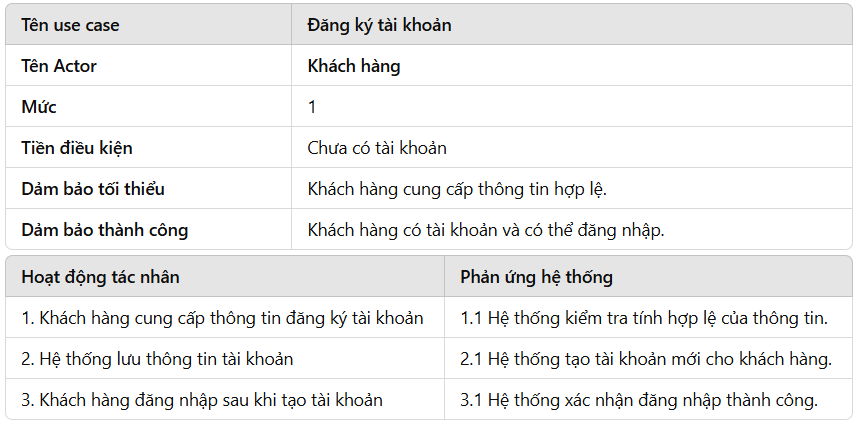
#### 

#### 

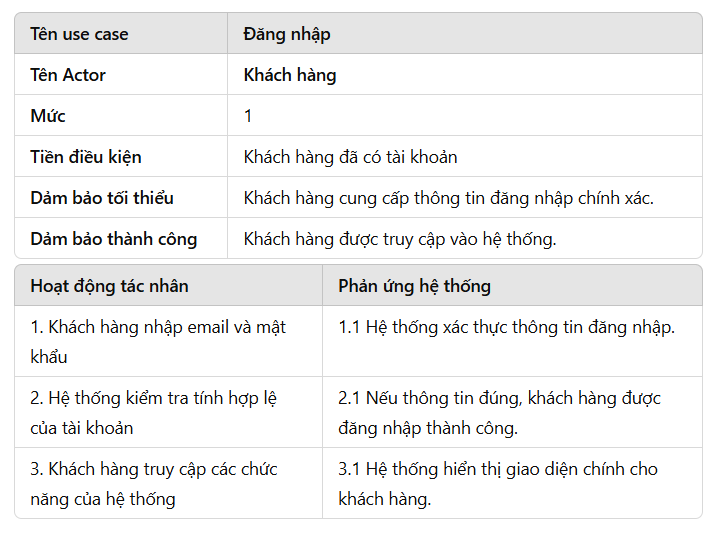
#### **1.4.1 Actor người dùng**

##### **Kịch bản cho Actor\_User**

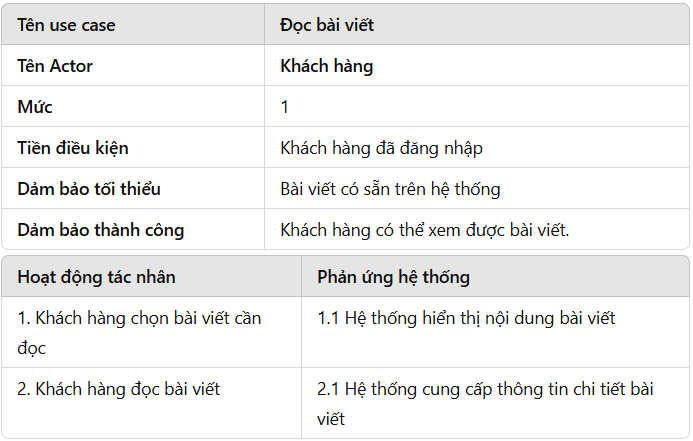
**Đăng ký**



**Đăng nhập**



**Đọc bài viết**



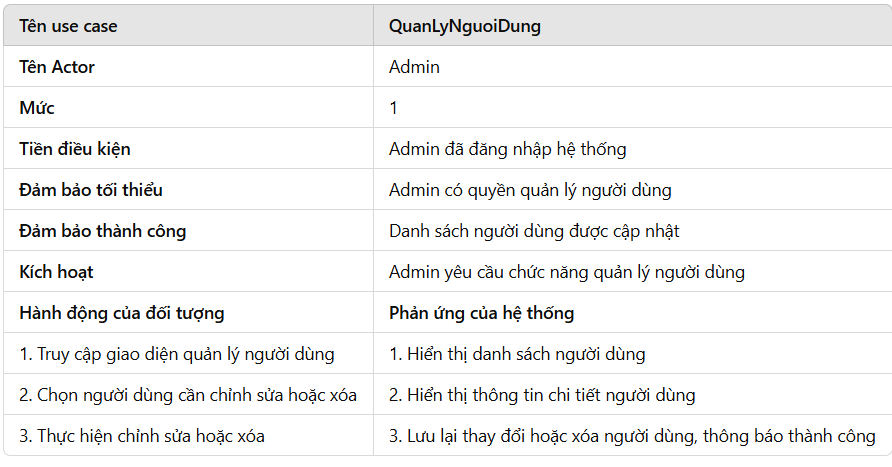
#### **1.4.2 Actor admin**

##### **Kịch bản cho actor admin**

**Đăng nhập**



**Quản lí người dùng**

****

**Kiểm duyệt bài viết**



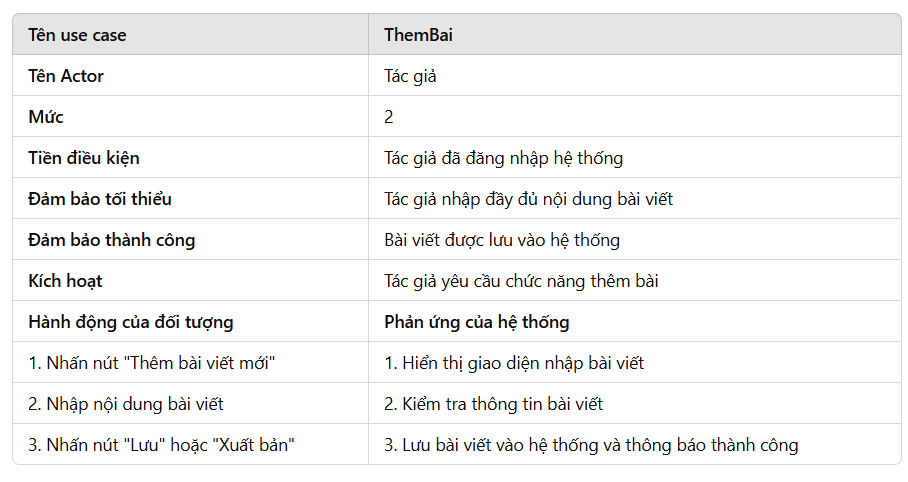
#### **1.4.3 Actor tác giả**

##### **Kịch bản cho actor tác giả**

**Đăng nhập**

****

**Thêm bài viết**

****

**Sửa bài**

****

**Xóa bài**

****

## 2. Biểu đồ hoạt động

### **2.1 Đăng ký**

**- Tên Use Case**: Đăng ký

**- Mô tả ngắn gọn**: Người dùng mới thực hiện đăng ký tài khoản bằng cách cung cấp thông tin cá nhân. Hệ thống kiểm tra và tạo tài khoản mới.

**- Actor**: Sinh viên, Giáo viên

**- Điều kiện tiên quyết**:

Người dùng đang ở **giao diện đăng ký**.

**- Điều kiện kết thúc**:

Tài khoản mới được tạo thành công và hệ thống thông báo kết quả.

**- Luồng sự kiện chính (Main Flow)**: **- Luồng sự kiện phụ (Alternative Flows)**:

**Tên đăng nhập đã tồn tại**:

Hệ thống thông báo: “Tên đăng nhập đã được sử dụng. Vui lòng chọn tên khác.”

Quay lại Bước 2.

**Xác nhận mật khẩu không khớp**:

Hệ thống thông báo: “Mật khẩu xác nhận không khớp. Vui lòng thử lại.”

Quay lại Bước 2.

**Email không hợp lệ**:

Hệ thống thông báo: “Email không hợp lệ. Vui lòng nhập đúng định dạng email.”

Quay lại Bước 2.

**- Kết quả đầu ra**:

Tài khoản mới được tạo thành công.

Người dùng được thông báo kết quả và chuyển đến trang **Đăng nhập**.

### **2.2 Đăng nhập**

**- Tên Use Case**: Đăng nhập

**- Mô tả ngắn gọn**: Người dùng nhập thông tin tài khoản để truy cập vào hệ thống. Hệ thống kiểm tra thông tin và phản hồi kết quả.

**- Actor**: Người dùng, admin, tác giả

**- Điều kiện tiên quyết**:

Người dùng đang ở **giao diện đăng nhập**.

**- Điều kiện kết thúc**:

Người dùng được chuyển đến **trang chủ hệ thống** (nếu đăng nhập thành công).

**- Luồng sự kiện chính (Main Flow)**:

**- Luồng sự kiện phụ (Alternative Flows)**:

**Sai tên đăng nhập hoặc mật khẩu**:

Hệ thống hiển thị thông báo: “Tên đăng nhập hoặc mật khẩu không đúng. Vui lòng thử lại.”

Quay lại bước nhập thông tin (Bước 1).

**- Kết quả đầu ra**:

Người dùng được chuyển đến **trang chủ hệ thống** khi đăng nhập thành công.

Thông báo lỗi nếu đăng nhập thất bại.

### **2.3 Xem bài viết**

**- Tên Use Case**: Xem bài viết

**- Mô tả ngắn gọn**: Người dùng xem các bài viết có sẵn trên hệ thống.

**- Actor**: người dùng, admin, tác giả

**- Điều kiện tiên quyết**:

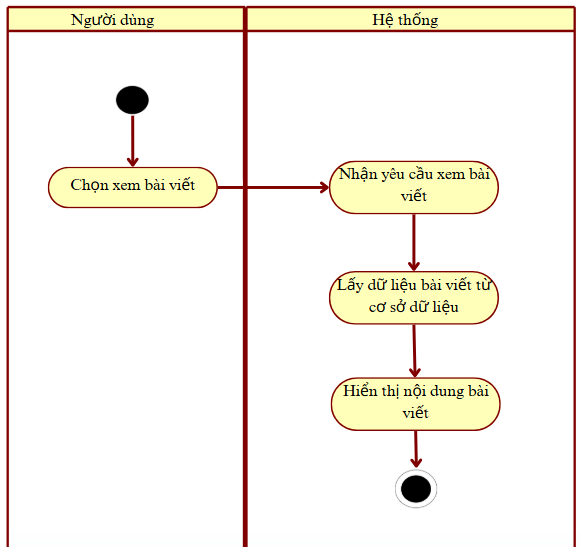
Người dùng đã đăng nhập vào hệ thống.

**- Điều kiện kết thúc**:

Người dùng xem bài viết thành công.

**- Luồng sự kiện chính (Main Flow)**:

* Người dùng truy cập vào trang chủ của blog ẩm thực.
* Hệ thống hiển thị danh sách các bài viết (bao gồm tiêu đề, ảnh minh họa, tóm tắt nội dung, và thông tin tác giả).
* Người dùng chọn bài viết muốn xem.
* Hệ thống hiển thị chi tiết bài viết, bao gồm:
* Tiêu đề bài viết.
* Nội dung chi tiết (văn bản, hình ảnh, video nếu có).
* Ngày đăng bài viết.
* Tên tác giả bài viết.
* Người dùng đọc nội dung bài viết.

****

### **2.4 Tìm kiếm**

**- Tên Use Case**: Tìm kiếm

**- Mô tả ngắn gọn**: Người dùng tìm kiếm bài viết theo từ khóa.

- **Actor**: người dùng, admin, tác giả

**- Điều kiện tiên quyết**:

Người dùng đã đăng nhập vào hệ thống.

**- Điều kiện kết thúc**:

Hệ thống hiển thị kết quả tìm kiếm.

**- Luồng sự kiện chính (Main Flow)**:

 Người dùng truy cập vào giao diện blog ẩm thực.

 Người dùng nhập từ khóa vào ô tìm kiếm (ví dụ: tên món ăn, nguyên liệu, tác giả bài viết, v.v.).

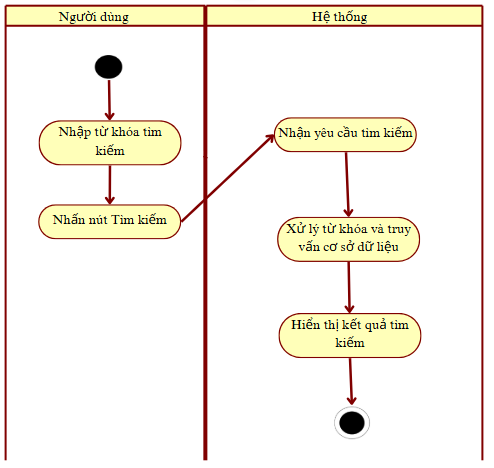
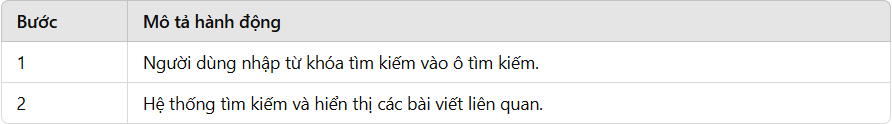
 Người dùng nhấn nút **"Tìm kiếm"**.

 Hệ thống kiểm tra và tìm kiếm bài viết dựa trên từ khóa đã nhập.

 Hệ thống hiển thị danh sách bài viết phù hợp với từ khóa, bao gồm:

* Tiêu đề bài viết.
* Tóm tắt nội dung.
* Ngày đăng bài viết.
* Tên tác giả.

 Người dùng chọn bài viết mong muốn để xem chi tiết (kết nối với Use Case **Xem bài viết**).

****

### **2.5 Xóa bài viết**

**- Tên Use Case**: Xóa bài viết

**- Mô tả ngắn gọn**: Admin xóa bài viết khỏi hệ thống.

**- Actor**: admin, Tác giả

**- Điều kiện tiên quyết**:

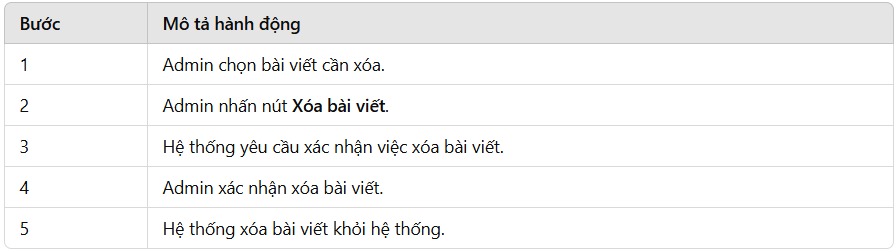
Admin đã đăng nhập vào hệ thống.

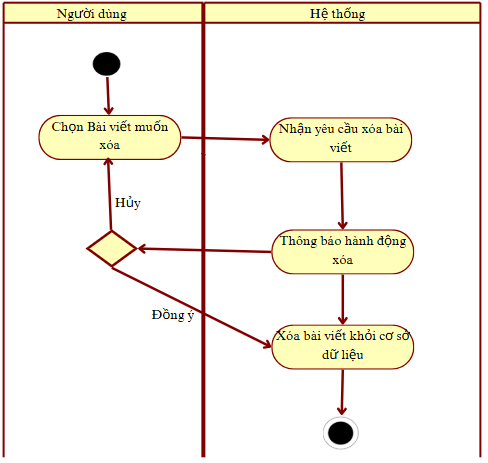
**- Điều kiện kết thúc**:

Bài viết được xóa khỏi hệ thống.

**- Luồng sự kiện chính (Main Flow)**:

* Người dùng (Admin hoặc Tác giả) truy cập vào danh sách bài viết.
* Người dùng tìm và chọn bài viết cần xóa.
* Hệ thống hiển thị hộp thoại xác nhận: **"Bạn có chắc chắn muốn xóa bài viết này không?"**
* Người dùng chọn **"Đồng ý"** để tiếp tục.
* Hệ thống xóa bài viết khỏi cơ sở dữ liệu.
* Hệ thống hiển thị thông báo: **"Bài viết đã được xóa thành công."**
* Danh sách bài viết được cập nhật, không còn hiển thị bài viết vừa bị xóa.

****

****

### **2.6 Đăng bài**

**- Tên Use Case**: Đăng bài

**- Mô tả ngắn gọn**: Người dùng đăng bài mới lên hệ thống.

**- Actor**: Tác giả

**- Điều kiện tiên quyết**:

Người dùng đã đăng nhập vào hệ thống.

**- Điều kiện kết thúc**:

Bài viết mới được đăng thành công.

**- Luồng sự kiện chính (Main Flow)**:

 Người dùng (Tác giả hoặc Admin) truy cập vào giao diện **“Đăng bài viết”**.

 Người dùng nhập các thông tin cần thiết, bao gồm:

* Tiêu đề bài viết.
* Nội dung bài viết (công thức, hướng dẫn, v.v.).
* Hình ảnh minh họa (nếu có).
* Từ khóa liên quan (tags).

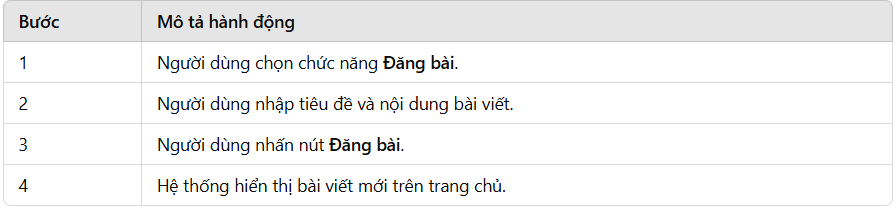
 Người dùng nhấn nút **"Đăng bài"**.

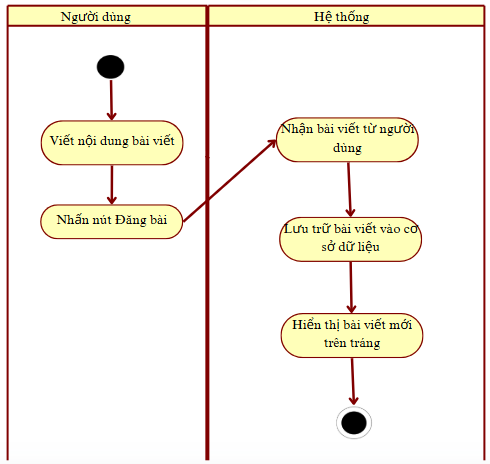
 Hệ thống kiểm tra dữ liệu đầu vào:

* Đảm bảo các trường thông tin bắt buộc được điền.

 Nếu hợp lệ, hệ thống lưu bài viết vào cơ sở dữ liệu và hiển thị thông báo: **"Bài viết đã được đăng tải thành công."**

 Bài viết xuất hiện trong danh sách bài viết công khai trên blog.

****



**2.7 Chỉnh sửa bài viết**

**- Tên Use Case**: Chỉnh sửa bài viết

**- Mô tả ngắn gọn**: Người dùng chỉnh sửa bài viết đã đăng.

**- Actor**: Tác giả

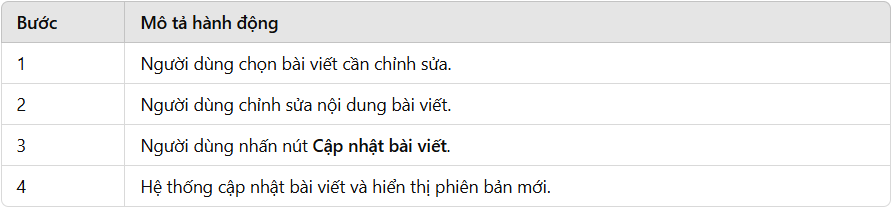
**- Điều kiện tiên quyết**:

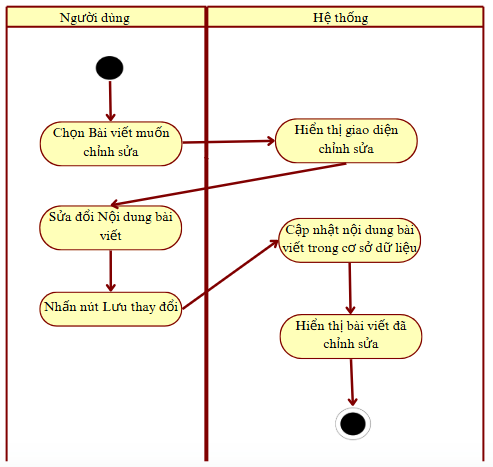
Người dùng đã đăng nhập vào hệ thống và sở hữu bài viết cần chỉnh sửa.

**- Điều kiện kết thúc**:

Bài viết được chỉnh sửa và cập nhật thành công.

**- Luồng sự kiện chính (Main Flow)**:

****



**2.8 Kiểm duyệt bài viết**

**- Tên Use Case**: Kiểm duyệt bài đăng

**- Mô tả ngắn gọn**: Admin kiểm tra và phê duyệt bài viết trước khi nó được hiển thị công khai.

**- Actor**: Admin

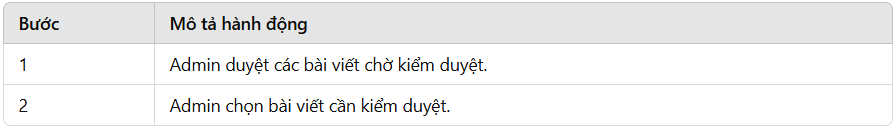
**- Điều kiện tiên quyết**:

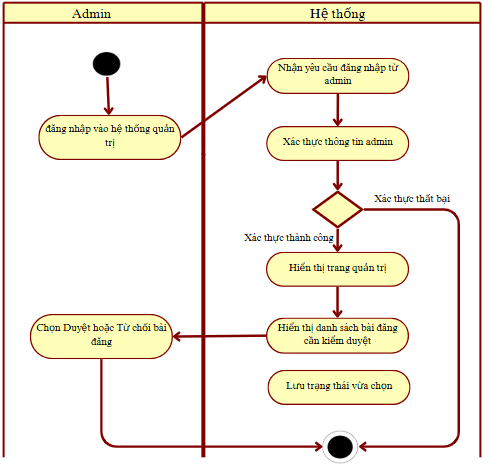
Admin đã đăng nhập vào hệ thống.

**- Điều kiện kết thúc**:

Bài viết được phê duyệt hoặc từ chối.

**- Luồng sự kiện chính (Main Flow)**:





## 3. Biểu đồ trạng thái

#### **3.1. Biểu đồ trạng thái đăng kí**

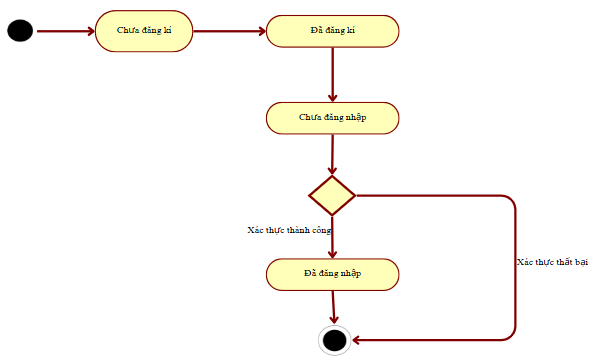
**3.1.1. Tên biểu đồ**: Biểu đồ trạng thái cho quy trình đăng ký và đăng nhập hệ thống

**3.1.2. Mục đích**: Biểu đồ trạng thái này mô tả quy trình chuyển đổi trạng thái của người dùng khi tương tác với hệ thống, từ lúc chưa đăng ký đến khi đăng nhập thành công hoặc thất bại.

**3.1.3. Các trạng thái chính**

**- Trạng thái ban đầu (Initial State)**:

* + Trạng thái khởi đầu của người dùng trước khi thực hiện bất kỳ hành động nào.
  + Ký hiệu: Hình tròn màu đen.
* **Chưa đăng ký**:
  + Người dùng chưa có tài khoản trong hệ thống.
  + Hành động khả dụng: Đi đăng ký.
* **đăng ký**:
  + Người dùng thực hiện quá trình đăng ký tài khoản.
  + Sau khi đăng ký xong, chuyển sang trạng thái "Chưa đăng nhập".
* **Chưa đăng nhập**:
  + Người dùng đã có tài khoản nhưng chưa đăng nhập.
  + Hành động khả dụng: Đi đăng nhập.
* **Đi đăng nhập**:
  + Người dùng thực hiện quá trình đăng nhập.
* **Xác thực**:
  + Hệ thống kiểm tra thông tin đăng nhập:
    - **Xác thực thành công**: Chuyển sang trạng thái "Đã đăng nhập".
    - **Xác thực thất bại**: Quay lại trạng thái "Chưa đăng nhập".
* **Đã đăng nhập**:
  + Trạng thái người dùng đã đăng nhập thành công.
* **Trạng thái kết thúc (Final State)**:
  + Quy trình kết thúc khi người dùng đã đăng nhập thành công.
  + Ký hiệu: Hình tròn đen viền trắng.



#### **3.2. Biểu đồ trạng thái đăng bài viết**

**3.2.1. Tên biểu đồ**

- Biểu đồ trạng thái cho quy trình đăng bài viết.

**3.2.2. Mục đích**

- Biểu đồ trạng thái này mô tả quy trình chuyển đổi trạng thái của bài viết khi được đăng lên hệ thống, từ lúc chưa đăng đến khi hoàn thành.

**3.2.3. Các trạng thái chính**

- **Trạng thái ban đầu (Initial State)**

* Mô tả: Trạng thái khởi đầu trước khi bài viết được thực hiện bất kỳ hành động nào.
* Ký hiệu: Hình tròn màu đen.

**- Chưa đăng bài viết**

* Mô tả: Trạng thái khi bài viết chưa được đăng lên hệ thống.
* Hành động khả dụng: "Đăng bài".

- **Đăng bài**

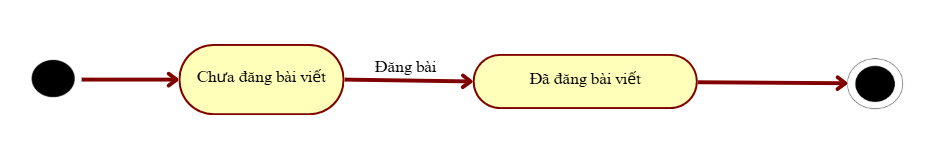
* Mô tả: Quá trình thực hiện đăng bài viết lên hệ thống.
* Sau khi hoàn tất, chuyển sang trạng thái "Đã đăng bài viết".

- **Đã đăng bài viết**

* Mô tả: Trạng thái khi bài viết đã được đăng thành công lên hệ thống.
* Hành động khả dụng: Không còn cần thiết.

 **Trạng thái kết thúc (Final State)**

* Mô tả: Quy trình đăng bài viết kết thúc khi bài viết đã được đăng thành công.
* Ký hiệu: Hình tròn đen viền trắng.



## 4. Biểu đồ lớp (Class)

### **4. 1. Các lớp dự kiến:**

**Admin**

**Tác giả**

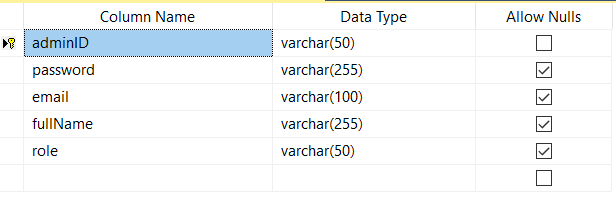
**Người dùng**

**Bài viết**

### **4.2. Các thuộc tính và phương thức:**

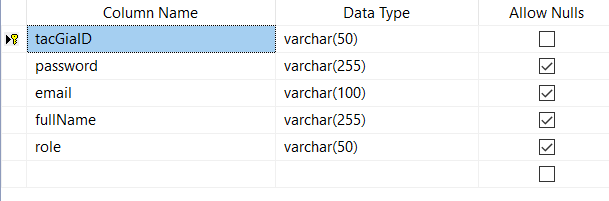
#### **4.2.1 Lớp Admin:**

* **Thuộc tính**:
  + adminID: String (Mã định danh của Admin)
  + password: String (Mật khẩu của Admin)
  + email: String (Email của Admin)
  + fullName: String (Tên đầy đủ của Admin)
  + role: String (Quyền hạn: Admin)
* **Phương thức**:
  + kiemDuyetBaiViet(baiViet: BaiViet): void – Kiểm duyệt bài viết (phê duyệt hoặc từ chối).
  + xoaBaiViet(baiViet: BaiViet): void – Xóa bài viết không phù hợp.
  + xoaTaiKhoan(nguoiDung: NguoiDung): void – Xóa tài khoản của tác giả hoặc người dùng.
  + dangNhap(email: String, password: String): boolean – Đăng nhập vào hệ thống.
  + dangKy(taiKhoan: NguoiDung): void – Đăng ký tài khoản cho người dùng mới.



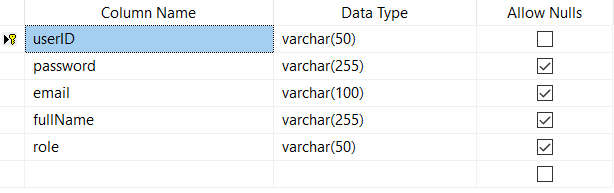
#### **4.2.2 Lớp Tác giả:**

* **Thuộc tính**:
  + tacGiaID: String (Mã định danh của tác giả)
  + password: String (Mật khẩu)
  + email: String (Email của tác giả)
  + fullName: String (Tên tác giả)
  + role: String (Quyền hạn: Tác giả)
* **Phương thức**:
  + dangNhap(email: String, password: String): boolean – Đăng nhập vào hệ thống.
  + xemBaiViet(): List<BaiViet> – Xem các bài viết trên hệ thống.
  + dangBai(tieuDe: String, noiDung: String, hinhAnh: List<Image>, tags: List<String>): BaiViet – Đăng bài viết mới.
  + suaBaiViet(baiViet: BaiViet, tieuDe: String, noiDung: String, hinhAnh: List<Image>): void – Chỉnh sửa bài viết của tác giả.
  + xoaBaiViet(baiViet: BaiViet): void – Xóa bài viết của mình.



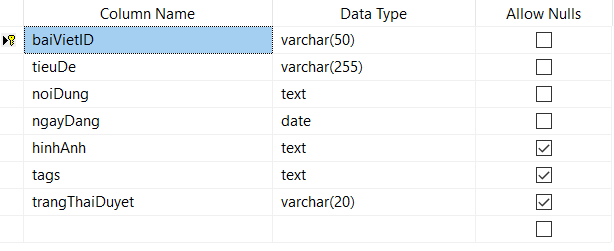
#### **4.2.3 Lớp Người dùng:**

* **Thuộc tính**:
  + userID: String (Mã định danh của người dùng)
  + password: String (Mật khẩu)
  + email: String (Email của người dùng)
  + fullName: String (Tên đầy đủ của người dùng)
  + role: String (Quyền hạn: Người dùng thông thường)
* **Phương thức**:
  + dangNhap(email: String, password: String): boolean – Đăng nhập vào hệ thống.
  + xemBaiViet(): List<BaiViet> – Xem các bài viết có sẵn.



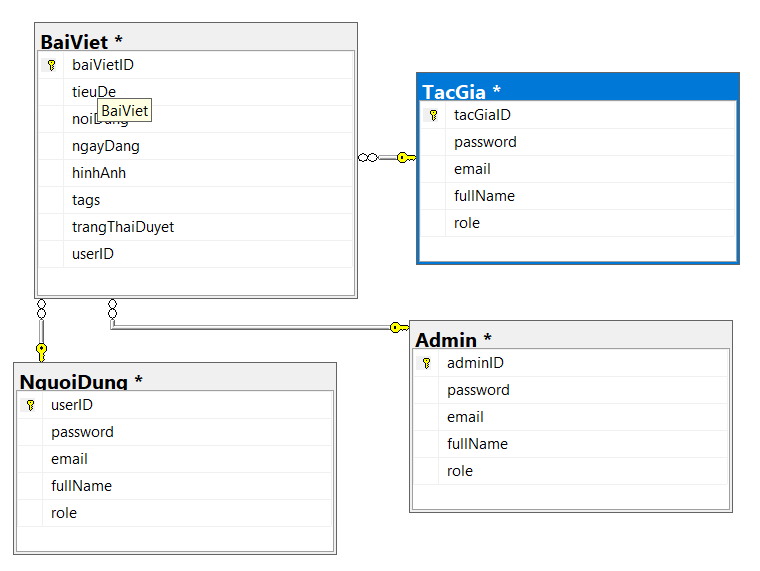
#### **4.2.4 Lớp Bài viết:**

* **Thuộc tính**:
  + baiVietID: String (Mã bài viết)
  + tieuDe: String (Tiêu đề bài viết)
  + noiDung: String (Nội dung bài viết)
  + ngayDang: Date (Ngày đăng bài viết)
  + hinhAnh: List<Image> (Danh sách hình ảnh minh họa)
  + tags: List<String> (Danh sách từ khóa liên quan)
  + trangThaiDuyet: boolean (Trạng thái duyệt: true/false)
* **Phương thức**:
  + xemBaiViet(): String – Xem chi tiết bài viết.



### **4.3. Các mối quan hệ:**

* **Admin** có quyền quản lý **Tác giả** và **Người dùng**.
* **Admin** có thể xóa bài viết và tài khoản người dùng.
* **Tác giả** và **Người dùng** có thể **xem bài viết**
* **Tác giả** có quyền đăng, chỉnh sửa, và xóa bài viết của mình.



#### **5. Biểu đồ trình tự**

#### **5.1. Đăng ký**

- Chức năng: Giúp người dùng đăng kí tài khoản để đăng nhập vào hệ thống

- Các bước thực hiện:

+ Người dùng nhập các thông tin cá nhân để đăng kí

+ Hệ thống lưu trữ thông tin và tự động đăng nhập tài khoản vừa đăng kí. Hiển thị dao diện trang chủ hệ thống

- Các đối tượng thực hiện như sau:

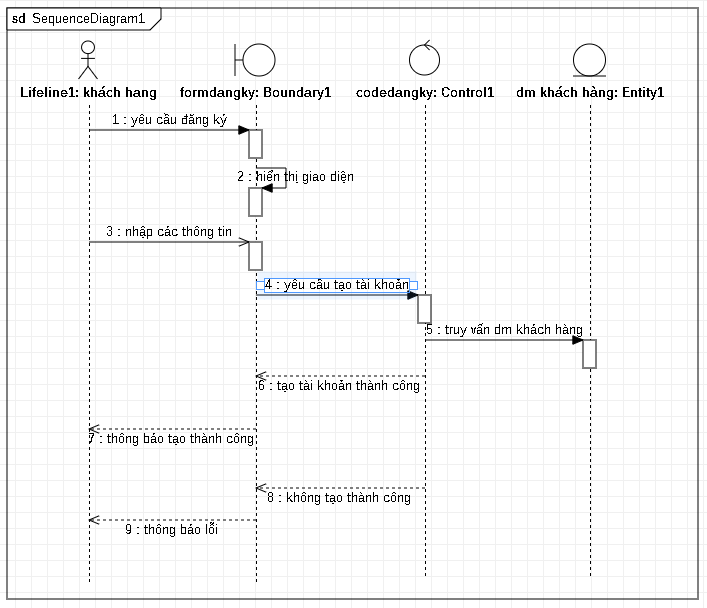
+Người dùng: Nhập thông tin đăng kí vào dao diện đăng kí của hệ thống

+ Hệ thống: Kiểm tra thông tin

+ Hệ thống: Lưu trữ thông tin

+ Hệ thống: Tự động đăng nhập tài khoản vào trang chủ hệ thống

+ Người dùng: Xem dao diện trang chủ



Hình 14. Biểu đồ tuần tự usecase Đăng ký

#### **5.2. Đăng nhập**

- Chức năng: Giúp người dùng đăng nhập tài khoản để sử dụng hệ thống

- Các bước thực hiện:

+ Người dùng nhập thông tin để đăng nhập

+ Hệ thống kiểm tra thông tin đăng nhập và đăng nhập

- Các đối tượng thực hiện như sau:

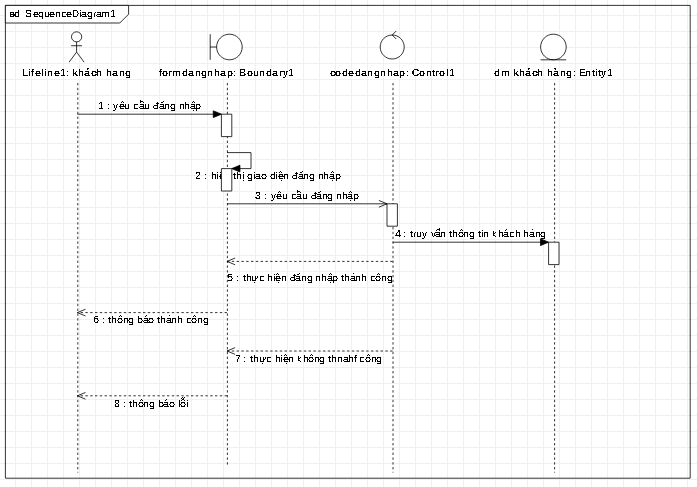
+ Người dùng: nhập usename và pass word

+ Hệ thống: Kiểm tra thông tin đăng nhập

Nếu đúng thì hiện thị giao diện trang chủ

Nếu sai hệ thống sẽ báo lỗi và yêu cầu đăng nhập lại

+ Người dùng: Xem dao diện trang chủ



Hình 15. Biểu đồ tuần tự usecase Đăng nhập

#### **5.3. Xóa tài khoản**

- Chức năng: Giúp người dùng xóa tài khoản khỏi hệ thống

- Các bước thực hiện:

+ Người dùng yêu cầu xóa

+ Hệ thống kiểm tra thông tin đăng nhập và phản hồi

- Các đối tượng thực hiện như sau:

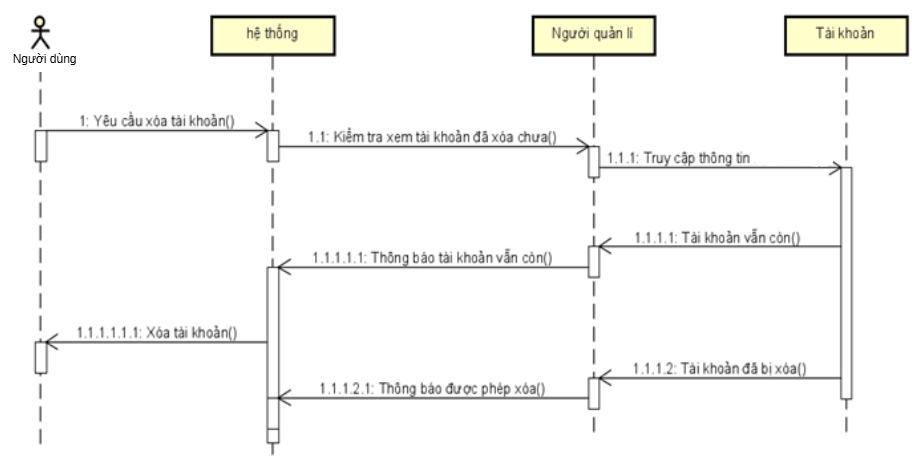
+ Người dùng: Yêu cầu xóa

+ Hệ thống: Kiểm tra thông tin xem đã được xóa chưa

Nếu đã được người quản lí đóng tài khoản thì thông báo cho binh sĩ được phép xóa

Nếu chưa được người quản lí đóng tài khoản thì thông báo không được phép xóa

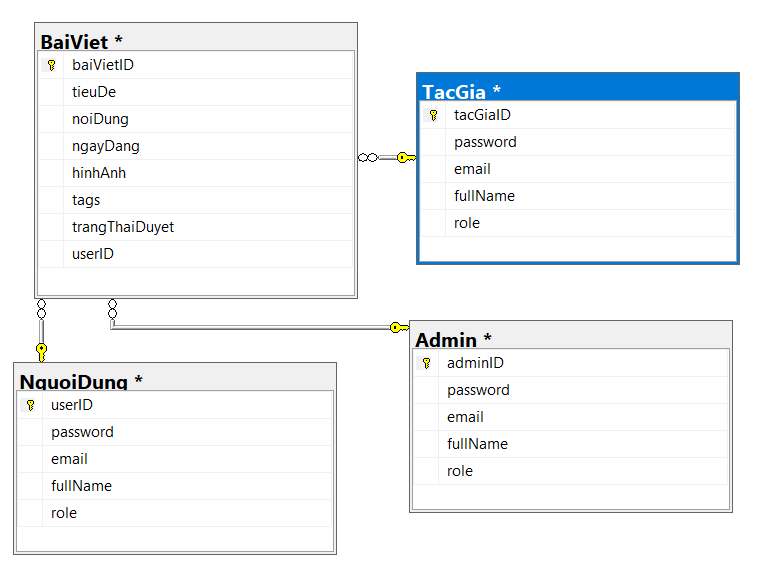
+ Người quản lí: Cho phép truy vấn thông tin tài khoản



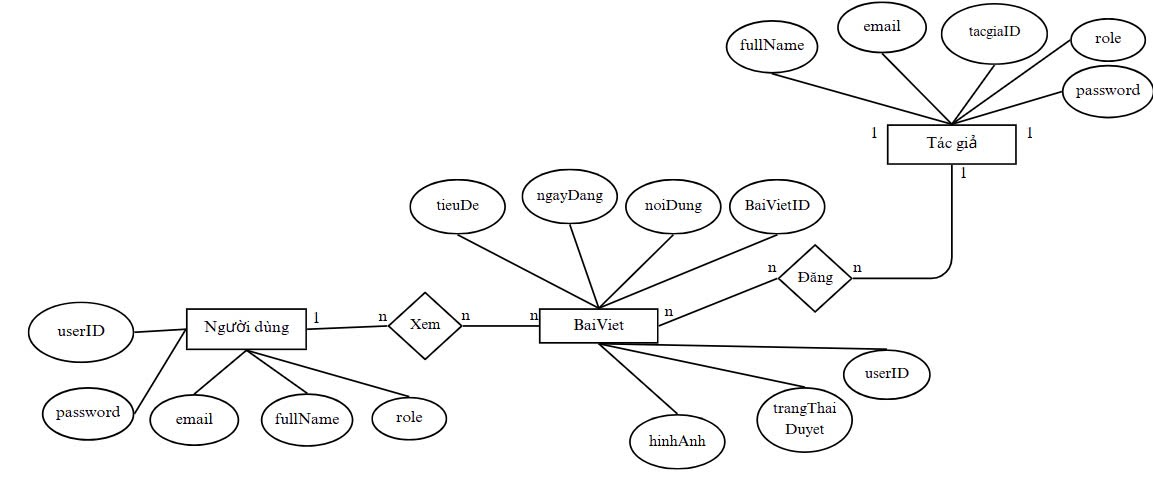
Hình 16. Biểu đồ tuần tự usecase Xóa tài khoản

# **CHƯƠNG 3. THIẾT KẾ HỆ THỐNG**

## 1.Cơ sở dữ liệu mức logic



## Ánh xạ



# **Kết luận**

**Những gì đã làm được:**

- Trong quá trình thiết kế và triển khai hệ thống quản lý em đã thu được những thành tựu sau:

**+ Xây dựng cơ sở dữ liệu**:

* + Xây dựng cơ sở dữ liệu một cách chặt chẽ
  + Liên kết giữa các bảng qua các trường riêng biệt giúp đảm bảo tính toàn vẹn dữ liệu và giảm thiểu sự dư thừa dữ liệu.

**+ Thiết kế giao diện người dùng bằng C#**:

* + Giao diện người dùng thân thiện và dễ sử dụng đã được thiết kế để hỗ trợ người dùng trong việc quản lý thông tin người dùng, các dụng cụ.

**Hạn chế:**

- Mặc dù đã đạt được nhiều thành tựu, hệ thống vẫn còn một số hạn chế cần được khắc phục trong tương lai:

**+ Chức năng giao diện người dùng**:

* + Giao diện người dùng hiện tại chỉ hỗ trợ các chức năng cơ bản. Các chức năng nâng cao như báo cáo thống kê, biểu đồ trực quan, và tìm kiếm nâng cao chưa được triển khai đầy đủ.

**+ Bảo mật và phân quyền**:

* + Hệ thống bảo mật và phân quyền hiện tại còn đơn giản. Cần nâng cấp tường lửa cho hệ thống trong tương lai

**+ Hiệu năng hệ thống**:

* + Với số lượng lớn dữ liệu và số lượng người dùng tăng cao, hệ thống có thể gặp vấn đề về hiệu năng. Hiện tại, chưa có cơ chế tối ưu hóa truy vấn và cân bằng tải để đảm bảo hiệu năng của hệ thống.

**=> Tổng kết :**

- Hệ thống đã tương đối hoàn thiện, đáp ứng tương đối tất cả các chức năng của người sử dụng , đặt nền móng vững chắc cho hệ thống và có thể phát triển thành quy mô lớn hơn trong tương lai. Góp phần quản lý phòng tập một cách logic và hiệu quả.

Em xin chân thành cảm ơn!

*Hà Nội, ngày tháng năm 20*

*(sinh viên ký, ghi đầy đủ họ và tên)*