

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC ĐẠI NAM**  
---  ---



## **ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG ỨNG DỤNG THỰC HIỆN QUY TRÌNH**  
**XUẤT BẢN CHO TẠP CHÍ KHOA HỌC**

**SINH VIÊN THỰC HIỆN : NGÔ THÀNH ĐỨC**

**MÃ SINH VIÊN : 1451020065**

**KHOA : CÔNG NGHỆ THÔNG TIN**

**HÀ NỘI - 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC ĐẠI NAM**

---



**NGÔ THÀNH ĐỨC**

**XÂY DỰNG ỨNG DỤNG THỰC HIỆN QUY TRÌNH  
XUẤT BẢN CHO TẠP CHÍ KHOA HỌC**

**SINH VIÊN THỰC HIỆN : NGÔ THÀNH ĐỨC**

**MÃ SINH VIÊN : 1451020065**

**NGƯỜI HƯỚNG DẪN: TS.TRẦN QUÝ NAM**

**HÀ NỘI - 2024**

## LỜI CAM ĐOAN

Tôi xin cam đoan đồ án tốt nghiệp với đề tài “Xây dựng ứng dụng thực hiện quy trình xuất bản cho tạp chí khoa học ” này được hoàn thành dựa trên các kết quả học tập của tôi dưới sự hướng dẫn của “TS TRẦN QUÝ NAM”. Các số liệu, kết quả nghiên cứu, và tài liệu tham khảo trong đồ án này là trung thực và đã được ghi nhận rõ ràng, chính xác. Tôi đã tuân thủ đầy đủ các quy định về bản quyền và trích dẫn tài liệu theo quy định của trường “Đại học Đại Nam”.

Tôi xin chịu hoàn toàn trách nhiệm về tính chính xác và trung thực của nội dung trong đồ án này.

Nếu phát hiện bất kỳ sự gian lận hay sai sót nào, tôi xin chịu mọi hình thức xử lý theo quy định của trường và pháp luật hiện hành.

Hà Nội, ngày      tháng      năm 2024

Ý kiến của giáo viên hướng dẫn

Người thực hiện

## LỜI CẢM ƠN

Lời đầu tiên, em muốn gửi những lời cảm ơn và biết ơn chân thành nhất của mình tới tất cả những người đã hỗ trợ, giúp đỡ em về kiến thức và tinh thần trong quá trình thực hiện thực đồ án tốt nghiệp.

Em xin chân thành gửi lời cảm ơn tới các giảng viên trong “Trường Đại Học Đại Nam” nói chung và các thầy/cô giáo trong “khoa Công Nghệ Thông Tin” nói riêng đã tận tình giảng dạy, truyền đạt cho chúng em những kiến thức cũng như kinh nghiệm quý báu trong suốt quá trình học tập.

Đặc biệt, em xin gửi lời cảm ơn đến giảng viên – “TS TRẦN QUÝ NAM”. Thầy đã tận tình theo sát giúp đỡ, trực tiếp chỉ bảo, hướng dẫn trong suốt quá trình nghiên cứu và làm đồ án tốt nghiệp. Trong thời gian được thầy chỉ bảo, em không những tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả từ thầy.

Do thời gian thực hiện có hạn, kiến thức còn nhiều hạn chế nên bài làm của em chắc chắn không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được ý kiến đóng góp của thầy/cô giáo và các bạn để em có thêm kinh nghiệm và tiếp tục hoàn thiện hơn trong tương lai sắp tới của mình.

Em xin chân thành cảm ơn!

## LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển như vũ bão, việc ra đời các website xuất bản tạp chí online nhằm đáp ứng nhu cầu ngày càng tăng của người tiêu dùng trong việc quản lý quy trình trực tuyến một cách thuận tiện và an toàn là rất cần thiết. Việc truy cập nhanh chóng và dễ dàng tiếp cận đến các sản phẩm và mua sắm ngay tại nhà giúp con người tiết kiệm rất nhiều thời gian và chi phí đi lại.

Đề tài " Xây dựng ứng dụng thực hiện quy trình xuất bản cho tạp chí khoa học" ra đời nhằm đáp ứng nhu cầu ngày càng tăng của người dùng trong việc xuất bản tạp chí trực tuyến một cách thuận tiện và an toàn. Hệ thống này được thiết kế để cung cấp một nền tảng dễ sử dụng, cho phép người dùng đăng tải, xem các trang tạp chí một cách nhanh chóng và tiện lợi.

Bằng cách áp dụng các công nghệ web tiên tiến và phương pháp quản lý bán hàng hiện đại, đề tài mong muốn xây dựng một nền tảng đáng tin cậy cho việc xuất bản tạp chí khoa học trực tuyến.

Trong quá trình thực hiện đề tài, em sẽ tập trung vào các mục tiêu chính như: thiết kế giao diện người dùng thân thiện, đảm bảo tính bảo mật và an toàn cho thông tin khách hàng, và tối ưu hóa hiệu suất của hệ thống. Đề tài cũng đặt ra mục tiêu đóng góp tích cực vào việc nâng cao chất lượng dịch vụ và sự hài lòng của khách hàng.

Em hy vọng rằng, thông qua việc hoàn thành đề tài này, em có thể đóng góp vào sự phát triển của doanh nghiệp và mang lại giá trị thực tiễn cho người tiêu dùng trong lĩnh vực xuất bản tạp chí.

Trong quá trình thực hiện, em đã tiến hành nghiên cứu về thị trường xuất bản tạp chí, tìm hiểu nhu cầu và hành vi của khách hàng, cũng như các quy trình quản lý quy trình xuất bản hiệu quả. Sau đó, em đã áp dụng kiến thức này để thiết kế và phát triển một hệ thống quản lý quy trình xuất bản tạp chí khoa học trực tuyến có khả năng đáp ứng các yêu cầu và mong muốn của người dùng.

Báo cáo này sẽ trình bày chi tiết về quá trình nghiên cứu, phân tích yêu cầu, thiết kế, và triển khai của đề tài tốt nghiệp. Em hy vọng rằng kết quả của đề tài này sẽ mang lại giá trị và lợi ích cho toà soạn tạp chí, người dùng.

[illegible]

## BẢNG CỤM TỪ VIẾT TẮT

STT	Từ viết tắt	Tên đầy đủ
1	CNTT	Công nghệ thông tin
2	QL	Quản lý
3	AI	Artificial Intelligence
4	CSDL	Cơ sở dữ liệu
5	DK	Điều kiện
6	UML	Unified Modeling Language



## DANH MỤC HÌNH ẢNH

Hình 3. 1: Mô hình phân cấp chức năng.....	26
Hình 3. 2: Sơ đồ Usecase tổng quát.....	28
Hình 3. 3: Sơ đồ Use case đăng nhập .....	30
Hình 3. 4: Use case quản lý số tạp chí chưa xuất bản .....	31
Hình 3. 5: Use case quản lý số tạp chí đã phản biện .....	32
Hình 3. 6: Use case quản lý số tạp chí đã xuất bản .....	32
Hình 3. 7: Use case tài khoản .....	33
Hình 3. 8: Use case quản lý bài viết tác giả.....	34
Hình 3. 9: Use case QL sản phẩm .....	35
Hình 3. 10: Sơ đồ tuần tự của use case đăng nhập .....	36
Hình 3. 11: Sơ đồ tuần tự của use case thêm tạp chí của quản lý số tạp chí đã xuất bản.....	37
Hình 3. 12: Sơ đồ tuần tự use case sửa tạp chí .....	38
Hình 3. 13: Sơ đồ tuần tự use xoá số tạp chí đã xuất bản.....	38
Hình 3. 14: Sơ đồ tuần tự use case cập nhật quyền .....	39

# MỤC LỤC

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI.....	1
1.1 Lý do chọn đề tài .....	1
1.2 Thách thức mà các toà soạn tạp chí đang phải đối mặt khi thực hiện quá trình xuất bản tạp chí thủ công .....	2
1.3 Tầm quan trọng của việc sử dụng website hệ thống quản lý quy trình xuất bản tạp chí khoa học trong môi trường hiện nay .....	2
1.4 Giới thiệu về ngôn ngữ sử dụng .....	3
1.5 Mục tiêu của website .....	4
1.6 Phạm vi nghiên cứu .....	5
1.7 Đối tượng nghiên cứu .....	6
1.8 Phương pháp nghiên cứu .....	7
CHƯƠNG II. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG .....	8
2.1 Giới thiệu về ngành công nghệ thông tin .....	8
2.2 Ngành công nghệ thông tin đối với hệ thống quản lý quy trình xuất bản tạp chí khoa học ..	9
2.3 Công nghệ sử dụng trong hệ thống quản lý quy trình xuất bản tạp chí.....	10
2.3.1 Ưu điểm .....	11
2.3.2 Hạn chế .....	12
2.4 Ngôn ngữ lập trình và nền tảng .....	12
2.4.1 Giới thiệu về ngôn ngữ lập trình C#.....	12
2.4.2 Giới thiệu về ASP.NET Core .....	15
2.4.3 Giới thiệu về VueJS.....	17
2.4.4 Giới thiệu về ant-design và bootstrap.....	18
2.4.5 Giới thiệu về Microsoft SQL Server Management Studio .....	20
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	25
3.1 Phát biểu bài toán .....	25
3.2 Các chức năng của hệ thống .....	25

3.2.1 Chức năng dành cho tác giả.....	25
3.2.2 Chức năng dành cho biên tập viên.....	25
3.2.3 Chức năng dành cho nhà phản biện.....	26
3.3 Mô hình phân cấp chức năng.....	26
3.4 Phân tích thiết kế use case theo UML .....	27
3.4.1 Giới thiệu về các Actor.....	27
3.4.2 Biểu đồ Usecase tổng quát.....	28
3.4.3 Use case đăng nhập.....	29
3.4.4 Use case quản lý số tạp chí chưa xuất bản .....	30
3.4.5 Use case quản lý số tạp chí đã phản biện .....	31
3.4.6 Use case quản lý số tạp chí đã xuất bản .....	32
3.4.7 Use case quản lý tài khoản .....	33
3.4.8 Use case quản lý bài viết tác giả.....	33
3.4.9 Use case quản lý phản biện .....	34
3.5 Sơ đồ tuần tự của các use case theo UML.....	36
3.5.1 Sơ đồ tuần tự của use case đăng nhập .....	36
3.5.2 Sơ đồ tuần tự của use case thêm tạp chí của quản lý số tạp chí đã xuất bản.....	37
3.5.3 Sơ đồ tuần tự use case sửa tạp chí của quản lý số tạp chí đã xuất bản.....	38
3.5.4 Sơ đồ tuần tự use case xóa tạp chí trong quản lý số tạp chí đã xuất bản.....	38
3.5.5 Sơ đồ tuần tự use case cập nhật quyền .....	39
3.5.6 Sơ đồ tuần tự use case xem danh sách user .....	39
3.5.7 Sơ đồ tuần tự use case gửi bài viết .....	40
3.5.8 Sơ đồ tuần tự use case phản biện bài.....	41
3.5.9 Sơ đồ tuần tự use case quản lý tạp chí đã phản biện .....	42
3.5.10 Sơ đồ tuần tự use case quản lý tạp chí chưa xuất bản .....	43
3.6 Thiết kế cơ sở dữ liệu .....	43
CHƯƠNG IV. XÂY DỰNG VÀ TRIỂN KHAI CHƯƠNG TRÌNH .....	48

4.1 Cấu trúc lập trình .....	48
4.2 Cách triển khai lập trình .....	50
4.2.1 Cách tạo bảng trong cơ sở dữ liệu sử dụng phương pháp Code First.....	50
4.2.2 Triển khai code backend.....	52
4.2.3 Triển khai code Frontend.....	71
4.3 Triển khai phần mềm(giao diện màn hình) .....	73
4.3.1 Giao diện đăng nhập.....	73
4.3.2 Tác giả thêm mới bài viết.....	74
4.3.3 Quản lý bài viết của tác giả .....	76
4.3.4 Quản lý số tạp chí chưa xuất bản.....	76
4.3.5 Quản lý số tạp chí đã xuất bản.....	77
4.3.5 Quản lý số tạp chí đã phản biện.....	79
4.3.6 Quản lý tài khoản.....	79
4.3.7 Quản lý bài phản biện.....	81
4.3.8 Giao diện trang chủ.....	83
KẾT LUẬN .....	86
TÀI LIỆU THAM KHẢO .....	88

## CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

### 1.1 Lý do chọn đề tài

Việc xây dựng một ứng dụng hỗ trợ quy trình xuất bản cho tạp chí khoa học đáp ứng nhu cầu thực tiễn trong lĩnh vực nghiên cứu và giáo dục. Hiện nay, các tạp chí khoa học đóng vai trò quan trọng trong việc công bố các kết quả nghiên cứu mới, tuy nhiên quy trình xuất bản truyền thống còn nhiều bất cập như mất thời gian, dễ xảy ra sai sót và thiếu minh bạch. Một ứng dụng số hóa quy trình này không chỉ giúp tối ưu hóa thời gian và chi phí mà còn nâng cao độ chính xác và hiệu quả trong việc quản lý và theo dõi các bài báo khoa học.

Ứng dụng này sẽ hỗ trợ tối ưu hóa quy trình làm việc giữa các tác giả, biên tập viên và phản biện. Bằng cách tạo ra một nền tảng trực tuyến tích hợp, các bên liên quan có thể dễ dàng trao đổi thông tin, gửi và nhận phản hồi, cũng như theo dõi tiến độ của bài báo. Điều này không chỉ tăng cường tính minh bạch và công bằng trong quy trình xuất bản mà còn khuyến khích sự hợp tác và trao đổi kiến thức giữa các nhà nghiên cứu trên toàn thế giới.

Việc phát triển ứng dụng này là một thách thức và cơ hội lớn trong lĩnh vực công nghệ thông tin, đặc biệt là trong ứng dụng công nghệ vào quản lý thông tin khoa học. Nó đòi hỏi sự kết hợp giữa nhiều kỹ năng từ lập trình, quản lý dữ liệu đến thiết kế giao diện người dùng. Điều này không chỉ giúp nâng cao trình độ chuyên môn của người phát triển mà còn góp phần vào sự tiến bộ của khoa học công nghệ trong việc giải quyết các vấn đề thực tiễn.

Việc xây dựng một ứng dụng xuất bản khoa học sẽ tạo ra giá trị to lớn cho cộng đồng nghiên cứu và giáo dục. Nó không chỉ giúp các tạp chí khoa học hoạt động hiệu quả hơn mà còn góp phần nâng cao chất lượng và uy tín của các công trình nghiên cứu được công bố. Đồng thời, ứng dụng này cũng mở ra nhiều cơ hội nghiên cứu mới trong lĩnh vực công nghệ thông tin và quản lý khoa học, góp phần vào sự phát triển bền vững của nền khoa học nước nhà.

## **1.2 Thách thức mà các toà soạn tạp chí đang phải đối mặt khi thực hiện quá trình xuất bản tạp chí thủ công**

Các toà soạn tạp chí đang phải đối mặt với nhiều thách thức khi thực hiện quá trình xuất bản tạp chí thủ công, bao gồm:

1. Tốn nhiều thời gian: Quy trình xuất bản thủ công yêu cầu nhiều bước phức tạp và tốn thời gian như tiếp nhận, phản biện, chỉnh sửa và in ấn.
2. Chi phí cao: Việc xử lý tài liệu, in ấn và vận chuyển giấy tờ liên quan đến quy trình xuất bản thủ công đòi hỏi chi phí đáng kể.
3. Chi phí cao: Việc xử lý tài liệu, in ấn và vận chuyển giấy tờ liên quan đến quy trình xuất bản thủ công đòi hỏi chi phí đáng kể.
4. Chi phí cao: Việc xử lý tài liệu, in ấn và vận chuyển giấy tờ liên quan đến quy trình xuất bản thủ công đòi hỏi chi phí đáng kể..
5. Chi phí cao: Việc xử lý tài liệu, in ấn và vận chuyển giấy tờ liên quan đến quy trình xuất bản thủ công đòi hỏi chi phí đáng kể.
6. Chi phí cao: Việc xử lý tài liệu, in ấn và vận chuyển giấy tờ liên quan đến quy trình xuất bản thủ công đòi hỏi chi phí đáng kể..
7. Chi phí cao: Việc xử lý tài liệu, in ấn và vận chuyển giấy tờ liên quan đến quy trình xuất bản thủ công đòi hỏi chi phí đáng kể..

Tóm lại, việc thực hiện quá trình xuất bản thủ công có thể gặp phải nhiều khó khăn về hiệu suất, độ chính xác và khả năng tương tác với khách hàng. Để giải quyết những thách thức này, các toà soạn tạp chí có thể cân nhắc sử dụng các giải pháp CNTT và hệ thống xuất bản tạp chí tự động hóa.

## **1.3 Tầm quan trọng của việc sử dụng website hệ thống quản lý quy trình xuất bản tạp chí khoa học trong môi trường hiện nay**

Sử dụng website hệ thống quản lý quy trình xuất bản tạp chí trong môi trường hiện nay có tầm quan trọng không thể phủ nhận với nhiều lợi ích đáng kể:

1. Tối ưu hóa thời gian và chi phí: Hệ thống tự động hóa quy trình xuất bản giúp giảm thiểu thời gian và chi phí so với quy trình thủ công, từ tiếp nhận bài viết đến công bố.
2. Tối ưu hóa thời gian và chi phí: Hệ thống tự động hóa quy trình xuất bản giúp giảm thiểu thời gian và chi phí so với quy trình thủ công, từ tiếp nhận bài viết đến công bố.
3. Tối ưu hóa thời gian và chi phí: Hệ thống tự động hóa quy trình xuất bản giúp giảm thiểu thời gian và chi phí so với quy trình thủ công, từ tiếp nhận bài viết đến công bố.
4. Tối ưu hóa thời gian và chi phí: Hệ thống tự động hóa quy trình xuất bản giúp giảm thiểu thời gian và chi phí so với quy trình thủ công, từ tiếp nhận bài viết đến công bố.
5. Tối ưu hóa thời gian và chi phí: Hệ thống tự động hóa quy trình xuất bản giúp giảm thiểu thời gian và chi phí so với quy trình thủ công, từ tiếp nhận bài viết đến công bố.
6. Tối ưu hóa thời gian và chi phí: Hệ thống tự động hóa quy trình xuất bản giúp giảm thiểu thời gian và chi phí so với quy trình thủ công, từ tiếp nhận bài viết đến công bố.
7. Đáp ứng nhu cầu phát triển: Trong bối cảnh khoa học và công nghệ phát triển nhanh chóng, hệ thống quản lý hiện đại giúp các tạp chí khoa học bắt kịp xu hướng, duy trì và nâng cao uy tín.

Tóm lại, việc sử dụng website hệ thống quản lý quy trình xuất bản tạp chí chỉ là một xu hướng mà còn là một yếu tố quan trọng trong việc tối ưu hóa việc xuất bản tạp chí khoa học trong môi trường hiện đại.

#### **1.4 Giới thiệu về ngôn ngữ sử dụng**

Dự án của em sử dụng các công nghệ và ngôn ngữ lập trình hiện đại, bao gồm ASP.NET, C#, và Vue.js.

ASP.NET là một framework phát triển web mạnh mẽ của Microsoft, cho phép tôi xây dựng các ứng dụng web và dịch vụ web một cách an toàn và linh hoạt. Công nghệ này hỗ trợ nhiều công cụ và thư viện, giúp tôi quản lý trạng thái, điều hướng, xác thực, và tương tác với cơ sở dữ liệu một cách hiệu quả.

C# là ngôn ngữ lập trình chính được sử dụng trong dự án này. Là một ngôn ngữ hướng đối tượng mạnh mẽ do Microsoft phát triển, C# cung cấp cú pháp rõ ràng và dễ sử dụng. Nó giúp tôi phát triển các ứng dụng web một cách hiệu quả và dễ bảo trì.

Để xây dựng giao diện người dùng tương tác và phản hồi nhanh, tôi đã sử dụng Vue.js, một framework phổ biến. Vue.js cho phép tôi tạo ra các ứng dụng web đơn trang với tính phản ứng tự động và thành phần hóa, từ đó giúp quản lý các ứng dụng lớn một cách dễ dàng.

Sự kết hợp của ASP.NET, C#, và Vue.js đã tạo ra một nền tảng vững chắc để em phát triển ứng dụng web hiện đại và hiệu quả. Nhờ vào các công nghệ này, em có thể xây dựng một ứng dụng web đáp ứng được nhu cầu của người dùng một cách tối ưu.

## **1.5 Mục tiêu của website**

Mục tiêu chính của website quản lý quy trình xuất bản tạp chí khoa học là tối ưu hóa và hiện đại hóa toàn bộ quy trình xuất bản. Bằng cách tự động hóa các bước từ tiếp nhận bài viết, phản biện, chỉnh sửa cho đến công bố, website này sẽ giúp giảm thiểu thời gian và chi phí cho cả tác giả lẫn tòa soạn. Hệ thống tự động giúp loại bỏ các công đoạn thủ công dễ gây sai sót, từ đó nâng cao độ chính xác và hiệu quả trong việc xử lý bài viết khoa học.

Một mục tiêu quan trọng khác là tăng cường tính minh bạch và khả năng theo dõi quy trình xuất bản. Website sẽ cung cấp các công cụ để các bên liên quan, bao gồm tác giả, biên tập viên và phản biện, có thể dễ dàng theo dõi tiến độ của từng bài viết. Tính năng này không chỉ giúp cải thiện sự phối hợp giữa các bên mà còn đảm



bảo rằng quy trình xuất bản diễn ra một cách công bằng và minh bạch, từ đó nâng cao sự tin tưởng và hài lòng của các tác giả.

Cuối cùng, website nhằm nâng cao chất lượng và uy tín của các tạp chí khoa học. Bằng cách sử dụng hệ thống quản lý hiện đại, tạp chí có thể đảm bảo rằng mỗi bài viết được đánh giá và chỉnh sửa kỹ lưỡng trước khi công bố. Hơn nữa, khả năng lưu trữ và truy xuất thông tin nhanh chóng và hiệu quả sẽ hỗ trợ các tòa soạn trong việc quản lý dữ liệu bài viết một cách chuyên nghiệp, góp phần vào sự phát triển bền vững và uy tín lâu dài của các tạp chí khoa học.

### **1.6 Phạm vi nghiên cứu**

Phạm vi ứng dụng sẽ tập trung vào xây dựng một hệ thống quản lý quy trình xuất bản cho tạp chí khoa học, bao gồm các giai đoạn từ tiếp nhận bài viết, phản biện, chỉnh sửa cho đến công bố. Cụ thể, ứng dụng sẽ:

- Tiếp nhận bài viết: Người dùng sẽ có thể gửi bài viết của mình thông qua giao diện của ứng dụng. Hệ thống sẽ tự động ghi nhận thông tin và chuyển tiếp đến bộ phận phản biện hoặc biên tập viên.
- Phản biện và chỉnh sửa: Biên tập viên và phản biện sẽ có thể truy cập vào hệ thống để đánh giá bài viết, đưa ra phản hồi và chỉnh sửa. Ứng dụng sẽ cung cấp các công cụ hỗ trợ cho quá trình này, bao gồm đánh giá chất lượng, xác định người phản biện phù hợp và theo dõi tiến độ chỉnh sửa.
- Công bố: Sau khi bài viết được phản biện, chỉnh sửa và duyệt, ứng dụng sẽ tự động công bố bài viết lên tạp chí hoặc cơ sở dữ liệu khoa học, đồng thời thông báo cho tác giả và các bên liên quan.

Phạm vi người dùng của ứng dụng bao gồm:

- tạo ra nội dung bài viết và gửi bài qua hệ thống. Họ có thể theo dõi tiến độ phản hồi và chỉnh sửa từ biên tập viên và phản biện.

- Biên tập viên: Người quản lý quy trình xuất bản, đảm bảo chất lượng và uy tín của các bài viết. Họ sẽ thực hiện vai trò đánh giá, chỉnh sửa và quản lý các bước trong quy trình xuất bản.
- Phản biện: Những chuyên gia trong lĩnh vực sẽ tham gia vào quá trình phản biện và đánh giá chất lượng của bài viết. Họ sẽ đóng vai trò quan trọng trong việc đánh giá tính khoa học và chuyên môn của nội dung.

### **1.7 Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài "Xây dựng ứng dụng thực hiện quy trình xuất bản cho tạp chí khoa học" sẽ bao gồm các nhóm đối tượng sau:

-Tác giả (Người viết bài viết khoa học): Đây là nhóm người tạo ra nội dung bài viết khoa học và gửi bài qua hệ thống quản lý xuất bản. Nghiên cứu sẽ tập trung vào việc hiểu và đáp ứng các nhu cầu, mong muốn của tác giả trong quá trình viết và xuất bản bài viết, đồng thời cung cấp các công cụ và dịch vụ hỗ trợ tối ưu hóa quy trình xuất bản.

-Biên tập viên (Editorial Board): Nhóm này đóng vai trò quan trọng trong việc quản lý quy trình xuất bản, đảm bảo chất lượng và uy tín của các bài viết. Nghiên cứu sẽ tìm hiểu và đáp ứng các yêu cầu, mong muốn của biên tập viên trong việc quản lý và điều hành quy trình xuất bản.

-Phản biện (Peer Reviewers): Là nhóm các chuyên gia trong lĩnh vực tương ứng với nội dung bài viết, tham gia vào quá trình phản biện và đánh giá chất lượng của bài viết. Nghiên cứu sẽ tập trung vào việc cung cấp các công cụ và dịch vụ hỗ trợ cho phản biện, giúp tăng cường tính minh bạch và chất lượng của quy trình phản biện.

-Độc giả (Readers): Nhóm người này là mục tiêu cuối cùng của quy trình xuất bản, là những người sẽ đọc và sử dụng các bài viết được công bố. Nghiên cứu sẽ quan tâm đến việc cải thiện trải nghiệm đọc và truy cập bài viết khoa học thông qua ứng dụng, đồng thời đảm bảo tính linh hoạt và dễ sử dụng.

## 1.8 Phương pháp nghiên cứu

### *a) Phương pháp nghiên cứu lý luận:*

- Tìm hiểu về các khía cạnh lý luận của thị trường xuất bản tạp chí, bao gồm thêm tạp chí , xuất bản, tác giả gửi bài thành công.
- Khảo sát các nguyên lý thiết kế và phát triển hệ thống, bao gồm kiến trúc microservices, RESTful API, và các phương pháp front-end hiện đại.

### *b) Phương pháp chuyên gia*

- Tìm hiểu từ các chuyên gia trong lĩnh vực xuất bản tạp chí và công nghệ, bao gồm những người có kinh nghiệm trong phát triển ứng dụng quản lý quy trình và các dự án tương tự.
- Thảo luận với các chuyên gia về các yêu cầu cụ thể, thực hiện phân tích và đề xuất giải pháp tối ưu.

### *c) Phương pháp thực nghiệm*

- Xây dựng một prototype hoặc mô hình ban đầu của hệ thống để kiểm tra các ý tưởng và giải pháp lý luận.
- Thực hiện thử nghiệm và điều chỉnh hệ thống dựa trên phản hồi từ việc triển khai thực tế và sử dụng.

## CHƯƠNG II. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

### 2.1 Giới thiệu về ngành công nghệ thông tin

Công nghệ thông tin (CNTT) là lĩnh vực nghiên cứu và ứng dụng công nghệ để quản lý và xử lý thông tin. Nó bao gồm nhiều lĩnh vực khác nhau như lập trình, mạng máy tính, cơ sở dữ liệu, an ninh mạng, trí tuệ nhân tạo, và nhiều hơn nữa. CNTT đóng vai trò quan trọng trong hầu hết mọi lĩnh vực của cuộc sống hiện đại, từ truyền thông, giáo dục, y tế, kinh doanh đến giải trí.

Một số lĩnh vực trong ngành CNTT bao gồm:

- + Phát triển phần mềm: Liên quan đến việc thiết kế, lập trình và duy trì các ứng dụng phần mềm phục vụ cho nhiều mục đích khác nhau, từ ứng dụng di động, trang web cho đến phần mềm quản lý doanh nghiệp.

- + Quản trị hệ thống mạng: Bao gồm việc thiết kế, cài đặt, cấu hình và duy trì hệ thống mạng để đảm bảo kết nối thông tin thông suốt trong một tổ chức.

- + Cơ sở dữ liệu: Nghiên cứu và phát triển các phương pháp lưu trữ, quản lý và truy vấn dữ liệu một cách hiệu quả, phục vụ cho nhiều ứng dụng khác nhau.

- + An ninh mạng: Tập trung vào việc bảo vệ hệ thống và dữ liệu khỏi các mối đe dọa và tấn công từ bên ngoài, bao gồm cả việc ngăn chặn virus, tấn công mạng và các hoạt động xâm phạm khác.

- + Trí tuệ nhân tạo (AI): Là lĩnh vực phát triển các hệ thống máy tính thông minh có khả năng học hỏi, suy nghĩ và thực hiện các tác vụ như con người.

- + Công nghệ đám mây: Liên quan đến việc cung cấp tài nguyên công nghệ như lưu trữ, xử lý và ứng dụng qua internet, cho phép người dùng truy cập vào dữ liệu và ứng dụng mọi lúc mọi nơi.

CNTT đóng vai trò quan trọng trong nhiều lĩnh vực như giáo dục, y tế, kinh doanh, giải trí và chính phủ. Việc ứng dụng CNTT không chỉ giúp nâng cao hiệu quả và chất lượng trong các lĩnh vực này mà còn thúc đẩy sự đổi mới và sáng tạo.

Trong giáo dục, CNTT giúp tiếp cận tri thức một cách nhanh chóng và linh hoạt, tạo điều kiện cho học tập từ xa và học tập suốt đời. Trong y tế, CNTT cải thiện quản lý hồ sơ bệnh nhân, chẩn đoán và điều trị bệnh, mang lại dịch vụ chăm sóc sức khỏe tốt hơn. Trong kinh doanh, CNTT hỗ trợ quản lý hiệu quả, tăng cường khả năng cạnh tranh và mở rộng thị trường. Trong lĩnh vực giải trí, CNTT tạo ra nhiều nền tảng và phương tiện giải trí phong phú, đa dạng, mang lại trải nghiệm người dùng tuyệt vời. Đối với chính phủ, CNTT đóng góp vào việc xây dựng một nền hành chính điện tử hiện đại, minh bạch và hiệu quả.

Tóm lại, CNTT là một yếu tố then chốt trong sự phát triển của xã hội hiện đại. Việc ứng dụng và phát triển CNTT không chỉ đáp ứng nhu cầu hiện tại mà còn tạo tiền đề cho những bước tiến vượt bậc trong tương lai. Để khai thác tối đa tiềm năng của CNTT, cần có sự đầu tư thích đáng vào hạ tầng, đào tạo nguồn nhân lực và chính sách hỗ trợ phù hợp. Chỉ khi đó, chúng ta mới có thể tận dụng được những lợi ích to lớn mà CNTT mang lại, góp phần xây dựng một xã hội thông minh, bền vững và thịnh vượng.

## **2.2 Ngành công nghệ thông tin đối với hệ thống quản lý quy trình xuất bản tạp chí khoa học**

Ngành công nghệ thông tin đóng vai trò quan trọng trong việc phát triển và vận hành hệ thống quản lý quy trình xuất bản tạp chí khoa học. Các công nghệ và công cụ được cung cấp bởi ngành này không chỉ giúp tối ưu hóa quy trình xuất bản mà còn nâng cao hiệu suất và chất lượng của các hoạt động liên quan.

Phát triển ứng dụng và phần mềm là một trong những phần quan trọng nhất của ngành công nghệ thông tin đối với hệ thống quản lý quy trình xuất bản. Các ứng dụng và phần mềm được phát triển có thể tùy chỉnh linh hoạt để phản ánh chính xác quy trình xuất bản của từng tạp chí, từ tiếp nhận bài viết cho đến công bố.

Xử lý dữ liệu lớn (Big Data) là một lĩnh vực quan trọng khác mà ngành công nghệ thông tin đóng vai trò quan trọng. Công nghệ này giúp phân tích dữ liệu từ các

bài viết khoa học, phản biện và dữ liệu liên quan khác, từ đó tối ưu hóa quy trình xuất bản dựa trên những thông tin chính xác và chi tiết.

Bảo mật thông tin và dữ liệu là một ưu tiên hàng đầu trong quy trình xuất bản. Ngành công nghệ thông tin cung cấp các giải pháp bảo mật thông tin và dữ liệu hiệu quả, đảm bảo rằng thông tin của các tác giả và các bài viết được bảo vệ một cách an toàn và bảo mật.

Công nghệ Blockchain là một công cụ mạnh mẽ có thể được sử dụng để tăng cường tính minh bạch và đảm bảo tính chính xác trong quy trình xuất bản. Việc áp dụng Blockchain giúp tạo ra các hồ sơ không thể sửa đổi được về quy trình xuất bản, từ đó tăng cường sự tin cậy và minh bạch.

Cuối cùng, công nghệ trí tuệ nhân tạo (AI) và học máy (Machine Learning) đóng vai trò quan trọng trong việc tự động hóa một số công việc trong quy trình xuất bản. Điều này giúp giảm thiểu thời gian và công sức cho các bên liên quan, đồng thời tăng cường hiệu quả và chính xác của quy trình xuất bản.

### **2.3 Công nghệ sử dụng trong hệ thống quản lý quy trình xuất bản tạp chí**

Công nghệ C# và Vue.js đóng vai trò quan trọng trong việc phát triển hệ thống quản lý quy trình xuất bản tạp chí khoa học. Cả hai công nghệ này đều được sử dụng để xây dựng các phần của hệ thống, đồng thời tạo ra một cấu trúc linh hoạt và mạnh mẽ cho ứng dụng.

C# được áp dụng vào phần backend của hệ thống, nơi xử lý logic, tương tác với cơ sở dữ liệu và xây dựng các API để giao tiếp với phía frontend. Sức mạnh của C# trong lập trình hướng đối tượng và tính năng bảo mật giúp đảm bảo tính ổn định và an toàn cho hệ thống.

Trong khi đó, Vue.js được sử dụng để phát triển giao diện người dùng tương tác. Với cấu trúc component-based linh hoạt, Vue.js cho phép xây dựng các giao diện người dùng động và dễ bảo trì. Các tính năng như two-way data binding và quản lý trạng thái giúp tạo ra trải nghiệm người dùng tốt hơn.

Bằng cách tích hợp giữa C# và Vue.js, ta có thể phát triển các ứng dụng web mạnh mẽ và linh hoạt. Vue.js có thể gửi các yêu cầu HTTP đến các API được xây dựng bằng C#, và nhận lại dữ liệu từ các API này để hiển thị trên giao diện người dùng, tạo ra một trải nghiệm tương tác tốt nhất cho người dùng.

### 2.3.1 Ưu điểm

Sự kết hợp giữa C# và Vue.js trong việc phát triển hệ thống quản lý quy trình xuất bản tạp chí khoa học mang lại nhiều ưu điểm đáng chú ý. Đầu tiên, tính linh hoạt của C# cho phép xử lý logic phức tạp và tương tác với cơ sở dữ liệu một cách hiệu quả, trong khi Vue.js cung cấp một cách tiếp cận đơn giản và mạnh mẽ cho phát triển giao diện người dùng.

Điểm mạnh thứ hai là hiệu suất cao mà sự kết hợp này mang lại. C# tối ưu hóa cho việc xử lý logic phức tạp và tương tác với dữ liệu, trong khi Vue.js giúp tạo ra các giao diện người dùng mượt mà và tương tác tốt. Sự kết hợp này đảm bảo rằng hệ thống hoạt động mạnh mẽ và đáp ứng được nhu cầu người dùng.

Một ưu điểm khác là quản lý trạng thái dễ dàng. Vue.js cung cấp các công cụ mạnh mẽ để quản lý trạng thái ứng dụng, giúp giảm thiểu sự phức tạp khi phát triển và bảo trì. Điều này giúp tạo ra các ứng dụng có tính tương tác cao và dễ sử dụng.

Khả năng mở rộng và mở nguồn cũng là một điểm lợi thế. Cả C# và Vue.js đều có cộng đồng lớn và hỗ trợ đa dạng từ cộng đồng phát triển phần mềm. Điều này giúp việc mở rộng và mở nguồn ứng dụng trở nên dễ dàng hơn, đồng thời đảm bảo tính bảo mật và tính ổn định của hệ thống.

Cuối cùng, tính tương thích cao và hỗ trợ API giúp kết hợp với các hệ thống khác một cách dễ dàng. Cả C# và Vue.js đều có khả năng tương thích tốt với các công nghệ khác và hỗ trợ việc tạo ra các API linh hoạt. Điều này giúp tối ưu hóa việc tích hợp và mở rộng ứng dụng, từ việc tích hợp với cơ sở dữ liệu đến tương tác với các dịch vụ bên ngoài.

### 2.3.2 Hạn chế

Sự kết hợp giữa C# và Vue.js trong phát triển hệ thống quản lý quy trình xuất bản tạp chí khoa học cung cấp nhiều ưu điểm, nhưng cũng không thiếu những hạn chế cần được xem xét kỹ lưỡng.

Một trong những hạn chế đáng lưu ý đó là khả năng chịu tải của hệ thống. Mặc dù C# là một ngôn ngữ mạnh mẽ và hiệu quả, nhưng thiếu tối ưu hóa có thể dẫn đến vấn đề về khả năng chịu tải khi có lượng truy cập đồng thời lớn.

Ngoài ra, C# cũng đòi hỏi một ngưỡng đầu vào khá cao, đặc biệt đối với những người mới bắt đầu. Điều này có thể tạo ra rào cản đối với việc học hỏi và phát triển cho một số nhà phát triển.

Một hạn chế khác là khả năng tùy chỉnh giao diện. Mặc dù Vue.js cung cấp một cách tiếp cận dễ hiểu và linh hoạt cho việc phát triển giao diện người dùng, nhưng có thể gặp khó khăn trong việc tùy chỉnh giao diện theo yêu cầu cụ thể của dự án.

Đối với các ứng dụng lớn với lượng dữ liệu và logic phức tạp, Vue.js có thể gặp vấn đề về hiệu suất. Điều này đòi hỏi sự tinh chỉnh kỹ thuật để đảm bảo hiệu suất ổn định và tốt nhất.

Cuối cùng, so với một số framework JavaScript khác, Vue.js có ít hơn các công cụ và thư viện đa dạng. Điều này có thể làm tăng thời gian phát triển và độ phức tạp của dự án, đặc biệt là khi xử lý các tác vụ phức tạp hoặc đòi hỏi tích hợp với các công nghệ khác.

## 2.4 Ngôn ngữ lập trình và nền tảng

### 2.4.1 Giới thiệu về ngôn ngữ lập trình C#

C# là một ngôn ngữ lập trình hướng đối tượng, được phát triển bởi Microsoft và ra mắt lần đầu vào năm 2000. C# là một phần của nền tảng .NET, một khung công tác phần mềm hỗ trợ phát triển các ứng dụng đa dạng như ứng dụng web, ứng dụng máy tính để bàn, ứng dụng di động, và dịch vụ web. Ngôn ngữ này nổi bật với cú pháp dễ hiểu, tương tự các ngôn ngữ như C và C++, giúp các nhà phát triển dễ



dòng chuyển đổi từ các ngôn ngữ khác. C# cũng cung cấp cơ chế kiểm soát kiểu dữ liệu chặt chẽ, giúp phát hiện lỗi ngay trong quá trình biên dịch và cải thiện hiệu suất ứng dụng.

C# hỗ trợ phát triển đa nền tảng nhờ sự ra đời của .NET Core, cho phép phát triển ứng dụng cho Windows, macOS, Linux và các nền tảng di động. Ngôn ngữ này còn được liên tục cập nhật với các tính năng hiện đại như lập trình bất đồng bộ, xử lý sự kiện và thuộc tính, và biểu thức lambda. C# tích hợp chặt chẽ với nền tảng .NET, mang lại khả năng sử dụng các thư viện và công cụ có sẵn trong hệ sinh thái .NET, đồng thời hỗ trợ cộng đồng lớn và tài liệu phong phú.

Nhờ các đặc điểm trên, C# là một ngôn ngữ mạnh mẽ, linh hoạt và đa năng, phù hợp cho nhiều loại dự án khác nhau, đặc biệt là trong phát triển ứng dụng doanh nghiệp, web và dịch vụ web. Điều này làm cho C# trở thành một lựa chọn hàng đầu cho các dự án công nghệ trong môi trường công nghệ thông tin hiện đại.

#### ***a) Lý do chọn ngôn ngữ lập trình C#***

Việc chọn ngôn ngữ lập trình C# để phát triển hệ thống bán các mặt hàng thời trang nữ xuất phát từ nhiều lý do quan trọng.

Đầu tiên, C# là một ngôn ngữ linh hoạt và đa năng, hỗ trợ nhiều loại dự án khác nhau, bao gồm phát triển web, ứng dụng máy tính để bàn, ứng dụng di động, và dịch vụ web. Sự tích hợp chặt chẽ với nền tảng .NET cung cấp khả năng tận dụng các dịch vụ và thư viện trong hệ sinh thái .NET, giúp quá trình phát triển ứng dụng trở nên nhanh chóng và hiệu quả.

C# cung cấp cơ chế kiểm soát kiểu dữ liệu chặt chẽ, giúp phát hiện lỗi sớm trong quá trình biên dịch và giảm thiểu các vấn đề trong quá trình chạy ứng dụng, nâng cao chất lượng và độ tin cậy của hệ thống bán hàng thời trang nữ. Hơn nữa, C# hỗ trợ lập trình bất đồng bộ, cho phép ứng dụng xử lý nhiều tác vụ đồng thời một cách hiệu quả, đáp ứng nhu cầu truy cập và tương tác cao từ người dùng.

Với các tính năng hướng đối tượng, C# cho phép xây dựng các ứng dụng có cấu trúc tốt, dễ bảo trì và mở rộng. Cộng đồng phát triển lớn mạnh của C# cung cấp

nhiều tài liệu, hướng dẫn, thư viện và công cụ hỗ trợ, giúp các nhà phát triển dễ dàng tìm hiểu và giải quyết các vấn đề phát sinh trong quá trình phát triển hệ thống bán hàng thời trang nữ.

Cuối cùng, C# được tối ưu hóa cho hiệu suất cao, đảm bảo hệ thống web hoạt động nhanh chóng và mượt mà, đáp ứng nhu cầu xử lý dữ liệu nhanh chóng và chính xác của hệ thống bán hàng thời trang nữ. Những ưu điểm này khiến C# trở thành lựa chọn hợp lý cho việc phát triển hệ thống bán hàng thời trang nữ, đảm bảo chất lượng, hiệu suất, và khả năng bảo trì trong tương lai.

### ***b) Lợi ích của ngôn ngữ lập trình C#***

Sự sử dụng của ngôn ngữ lập trình C# trong việc quản lý quy trình xuất bản tạp chí khoa học mang lại nhiều lợi ích đáng kể, đặc biệt là trong việc xây dựng phần backend của hệ thống.

Một trong những lợi ích lớn nhất của C# là tính linh hoạt và mạnh mẽ của nó trong việc xử lý logic phức tạp. Ngôn ngữ này cung cấp một loạt các tính năng tiên tiến như lập trình hướng đối tượng, lambda expressions và LINQ (Language Integrated Query), giúp việc quản lý dữ liệu và tương tác với cơ sở dữ liệu trở nên dễ dàng và hiệu quả hơn.

C# cũng được biết đến với tính bảo mật cao. Với các tính năng như kiểm tra kiểu tại thời gian biên dịch và quản lý bộ nhớ tự động, C# giúp giảm thiểu nguy cơ các lỗi phổ biến như tràn bộ nhớ và injection code, từ đó tăng cường tính an toàn và độ tin cậy của hệ thống.

Một ưu điểm khác của C# là khả năng tích hợp tốt với nền tảng .NET Framework hoặc .NET Core. Điều này mở ra cánh cửa cho việc sử dụng các thư viện và công nghệ có sẵn trong hệ sinh thái của Microsoft, từ việc xây dựng API đến tích hợp các công cụ phát triển khác như Visual Studio.

C# cũng hỗ trợ việc phát triển các ứng dụng đa nền tảng (cross-platform) thông qua .NET Core, giúp hệ thống có khả năng chạy trên nhiều hệ điều hành và môi trường khác nhau một cách linh hoạt.

Cuối cùng, cộng đồng lớn và sự hỗ trợ mạnh mẽ từ Microsoft là một lợi ích quan trọng khác của việc sử dụng C#. Sự phát triển và cập nhật liên tục của ngôn ngữ này đảm bảo rằng các nhà phát triển có nguồn lực và hỗ trợ cần thiết để xây dựng và duy trì hệ thống quản lý quy trình xuất bản tạp chí một cách hiệu quả.

### ***c) Ứng dụng C# trong hệ thống quản lý quy trình xuất bản tạp chí***

Trong hệ thống quản lý quy trình xuất bản tạp chí, ngôn ngữ lập trình C# đóng vai trò quan trọng trong nhiều khía cạnh. Đầu tiên, C# được sử dụng để xử lý logic nghiệp vụ, giúp quản lý các quy trình phức tạp như quản lý thông tin về tác giả, bài báo và các quy trình xuất bản.

Thứ hai, C# được áp dụng để tương tác với cơ sở dữ liệu, kết nối và truy xuất dữ liệu từ các hệ thống như SQL Server hoặc MongoDB, đảm bảo rằng hệ thống có thể lưu trữ và truy xuất thông tin một cách hiệu quả.

Thứ ba, C# phát triển các dịch vụ web và API, cung cấp các điểm cuối để tương tác với các ứng dụng frontend hoặc các hệ thống bên ngoài, tạo điều kiện cho tính linh hoạt và tích hợp.

Bên cạnh đó, C# được sử dụng để xây dựng các công cụ quản trị như giao diện quản trị và bảng điều khiển, giúp người quản trị kiểm soát và giám sát các quy trình xuất bản một cách hiệu quả.

Ngoài ra, C# còn giúp xử lý các tính năng bảo mật như xác thực người dùng và quản lý quyền truy cập, đảm bảo an toàn cho thông tin quan trọng như dữ liệu tác giả và bài báo.

Cuối cùng, C# tích hợp tốt với các công nghệ khác như JavaScript, HTML/CSS, Angular, React để phát triển các ứng dụng web phức tạp và tương tác với người dùng một cách linh hoạt và hiệu quả.

## **2.4.2 Giới thiệu về ASP.NET Core**

ASP.NET Core là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cụ mà được triển khai trên đám mây (cloud) hoặc chạy on-promise.

Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, giữ lại được sự mềm dẻo trong việc xây dựng giải pháp. Ta có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

***a) Một số lợi ích khi sử dụng ASP.NET Core cho việc xây dựng backend:***

- Mã nguồn mở: ASP.NET Core là mã nguồn mở, cho phép bạn tự do sử dụng và sửa đổi.
- Đa nền tảng: ASP.NET Core có thể chạy trên Windows, macOS và Linux.
- Hiệu suất cao: ASP.NET Core được thiết kế để có hiệu suất cao, giúp bạn xây dựng các ứng dụng web có thể xử lý lượng truy cập lớn.
- Có thể mở rộng: ASP.NET Core được thiết kế để có thể mở rộng, giúp bạn dễ dàng phát triển các ứng dụng web lớn và phức tạp.
- Hỗ trợ nhiều ngôn ngữ lập trình: ASP.NET Core hỗ trợ nhiều ngôn ngữ lập trình khác nhau, bao gồm C#, F# và Visual Basic .NET.
- Cộng đồng lớn: ASP.NET Core có một cộng đồng lớn các nhà phát triển có thể hỗ trợ bạn trong quá trình phát triển.

***b) ASP.NET Core có thể được sử dụng để xây dựng backend cho nhiều loại ứng dụng web khác nhau, bao gồm:***

- Trang web: ASP.NET Core có thể được sử dụng để xây dựng backend cho các trang web tĩnh và động.
- API: ASP.NET Core có thể được sử dụng để xây dựng backend cho các API RESTful.
- Ứng dụng web một trang: ASP.NET Core có thể được sử dụng để xây dựng backend cho các ứng dụng web một trang (SPA) sử dụng JavaScript frameworks như AngularJS, ReactJS hoặc VueJS.

Ứng dụng web di động: ASP.NET Core có thể được sử dụng để xây dựng backend cho các ứng dụng web di động bằng cách sử dụng Xamarin.

### 2.4.3 Giới thiệu về VueJS

Vue.js là một framework Javascript được tạo bởi Evan You, giúp chúng ta xây dựng giao diện người dùng cũng như xây dựng Single Page Application thân thiện với người dùng, chúng xây dựng từ các thư viện, cách triển khai component, các chức năng đặc trưng của nó như SFC (Single File Component).

Khác với các framework nguyên khối, Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo các bước. Khi phát triển lớp giao diện, người dùng chỉ cần dùng thư viện lõi (core library) của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn. Cùng lúc đó, nếu kết hợp với những kĩ thuật hiện đại như SFC (single file components) và các thư viện hỗ trợ, Vue cũng đáp ứng được dễ dàng nhu cầu xây dựng những ứng dụng đơn trang (SPA – Single Page Applications) với độ phức tạp cao.

#### **Một số công dụng của vue.js:**

*Nâng cao HTML tĩnh mà không cần bước xây dựng*

Vue có thể được sử dụng như một file script độc lập – không cần bước xây dựng! Nếu bạn đã có một khung backend hiển thị hầu hết HTML, hoặc logic frontend của bạn không phức tạp đến mức cần một bước xây dựng, đây là cách dễ nhất để

tích hợp Vue vào stack của bạn. Bạn có thể coi Vue như một thay thế khai báo hơn cho jQuery trong những trường hợp như vậy.

### *Nhúng như Web Components trên trang bất kỳ*

Bạn có thể sử dụng Vue để xây dựng các Web Components chuẩn có thể được nhúng vào bất kỳ trang HTML nào, bất kể chúng được hiển thị như thế nào. Tùy chọn này cho phép bạn sử dụng Vue một cách hoàn toàn độc lập với người tiêu dùng: các web components kết quả có thể được nhúng vào các ứng dụng cũ, HTML tĩnh, hoặc thậm chí là các ứng dụng được xây dựng với các khung khách.

### *Ứng dụng Đơn Trang (SPA)*

Một số ứng dụng yêu cầu sự tương tác phong phú, độ sâu phiên làm việc cao, và logic trạng thái phức tạp trên frontend. Cách tốt nhất để xây dựng những ứng dụng như vậy là sử dụng một kiến trúc nơi VueJS không chỉ kiểm soát toàn bộ trang, mà còn xử lý các cập nhật dữ liệu và điều hướng mà không cần phải tải lại trang.

Loại ứng dụng này thường được gọi là Ứng dụng Đơn Trang (SPA). Vue cung cấp các thư viện cốt lõi và hỗ trợ công cụ toàn diện với trải nghiệm tuyệt vời cho nhà phát triển để xây dựng các SPA hiện đại, bao gồm:

- Router phía khách hàng
- Hệ thống xây dựng siêu tốc độ
- Hỗ trợ IDE
- Bộ công cụ phát triển Browser
- Tích hợp TypeScript
- Tiện ích kiểm tra
- Fullstack / Server-Side Rendering (SSR)

#### **2.4.4 Giới thiệu về ant-design và bootstrap**

\*Ant Design là một bộ công cụ UI (User Interface) và thư viện giao diện người dùng được phát triển bởi nhóm Ant Financial, một phần của Tập đoàn Alibaba. Ant

Design cung cấp các thành phần giao diện người dùng đồng nhất và tương thích trên nhiều nền tảng, giúp người phát triển xây dựng các ứng dụng web và di động một cách nhanh chóng và hiệu quả.

Đặc điểm chính của Ant Design bao gồm:

**Thành phần đồng nhất:** Ant Design cung cấp một bộ sưu tập các thành phần giao diện người dùng đồng nhất, bao gồm các nút, biểu mẫu, bảng, menu, thanh điều hướng và nhiều hơn nữa. Các thành phần này được thiết kế với sự cân nhắc kỹ lưỡng để đảm bảo tính thẩm mỹ và trải nghiệm người dùng tốt nhất.

**Tính tương thích và Responsive:** Ant Design được thiết kế để hoạt động trên nhiều nền tảng và thiết bị, từ máy tính để bàn đến điện thoại di động. Các thành phần của nó có khả năng Responsive, tự động thích ứng với kích thước màn hình khác nhau một cách linh hoạt và dễ dàng.

**Thiết kế dễ sử dụng:** Giao diện của Ant Design được thiết kế với mục tiêu tối giản và dễ sử dụng. Các thành phần được tổ chức một cách logic và dễ tìm kiếm, giúp người phát triển tăng cường hiệu suất và tiết kiệm thời gian.

**Tùy chỉnh linh hoạt:** Ant Design cho phép người dùng tùy chỉnh và điều chỉnh các thành phần theo nhu cầu cụ thể của dự án. Điều này giúp tạo ra các ứng dụng có sự độc đáo và phong cách riêng biệt.

**Cộng đồng lớn và hỗ trợ đa ngôn ngữ:** Ant Design có một cộng đồng lớn và tích hợp hỗ trợ cho nhiều ngôn ngữ khác nhau, từ đó đảm bảo rằng người dùng có thể tìm thấy các tài nguyên và hỗ trợ cần thiết một cách dễ dàng.

Tóm lại, Ant Design là một bộ công cụ UI mạnh mẽ và linh hoạt, được sử dụng rộng rãi trong việc phát triển các ứng dụng web và di động, đặc biệt là trong cộng đồng lập trình viên phát triển React và Vue.js. Với những đặc điểm trên, Vuetify là một lựa chọn phổ biến cho việc xây dựng giao diện người dùng trong các ứng dụng Vue.js.

\* Bootstrap là một framework front-end phổ biến được phát triển bởi Twitter. Nó cung cấp một bộ công cụ linh hoạt để phát triển các ứng dụng web và di động một cách nhanh chóng và dễ dàng. Dựa trên HTML, CSS và JavaScript, Bootstrap cung cấp một loạt các thành phần và mẫu thiết kế đã được chuẩn bị sẵn, giúp tạo ra các giao diện đẹp mắt và phản hồi trên mọi thiết bị.

Bootstrap bao gồm một số tính năng chính sau:

**Thành phần giao diện người dùng (UI components):** Bootstrap cung cấp một bộ sưu tập các thành phần giao diện người dùng đa dạng như nút, biểu mẫu, bảng, thanh điều hướng, menu dropdown, hình ảnh và nhiều hơn nữa. Những thành phần này được thiết kế để có giao diện thân thiện, dễ sử dụng và tương thích trên nhiều loại thiết bị khác nhau.

**Grid System:** Bootstrap sử dụng một hệ thống grid linh hoạt và mạnh mẽ để sắp xếp các thành phần trên trang web. Hệ thống grid của Bootstrap cho phép phân chia trang thành các cột và hàng, giúp tạo ra giao diện phản hồi và đa dạng trên các thiết bị khác nhau.

**Tính Responsive:** Bootstrap được thiết kế để hoạt động một cách phản hồi trên mọi thiết bị, từ máy tính để bàn đến điện thoại di động. Các thành phần và mẫu của nó tự động thích ứng với kích thước màn hình và trải nghiệm người dùng.

**JavaScript Plugins:** Bootstrap đi kèm với một loạt các plugin JavaScript tích hợp sẵn như carousel, modal, tooltip, popover và nhiều hơn nữa. Những plugin này giúp tạo ra các hiệu ứng và tính năng tương tác động cho trang web của bạn mà không cần phải viết mã JavaScript từ đầu.

**Tùy chỉnh dễ dàng:** Bootstrap có thể được tùy chỉnh dễ dàng thông qua việc sử dụng biến Sass và bộ công cụ tùy chỉnh trực tuyến của Bootstrap. Điều này cho phép bạn điều chỉnh màu sắc, font chữ, kích thước và các thuộc tính khác theo nhu cầu cụ thể của dự án.

#### **2.4.5 Giới thiệu về Microsoft SQL Server Management Studio**



SQL server là một dạng hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - Viết tắt là RDBMS). Nó được phát triển bởi gã khổng lồ trong làng công nghệ Microsoft vào năm 1989 và vẫn được sử dụng rộng rãi đến ngày nay.

Có thể khi thấy chữ server thì nhiều bạn sẽ có những hiểu nhầm ở đây. Chúng ta sẽ cùng làm rõ về khái niệm SQL server bằng cách xem qua những đặc tính dưới đây của nó nhé:

+ Thứ nhất, server là một thiết bị phần cứng nhưng SQL server thì về bản chất nó là một sản phẩm phần mềm. Nó được các kỹ sư của Microsoft xây dựng và phát triển từ cách đây gần 30 năm rồi. Vì là sản phẩm phần mềm nên nó được cài trên các thiết bị phần cứng như server.

+ Thứ hai, nó có chức năng chính là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm khác. Chúng ta sẽ lưu trữ dữ liệu vào đó và sử dụng các câu lệnh để tìm kiếm dữ liệu khi cần.

+ Thứ ba, nó sử dụng câu lệnh SQL (Transact-SQL) để trao đổi dữ liệu giữa máy khách (máy Client) và máy cài SQL Server.

#### ***a) Cấu trúc cơ bản của SQL Server:***

##### ***SQLOS***

SQLOS là viết tắt của hệ điều hành SQL server. Đây là tầng cuối cùng trong kiến trúc tổng thể của SQL server. Tại đây sẽ chịu trách nhiệm xử lý các nhiệm vụ như quản lý bộ nhớ, lên lịch nhiệm vụ, khoá dữ liệu nhằm tránh các xung đột ngoài ý muốn có thể xảy ra mỗi khi thực hiện các thao tác cập nhật

##### ***Database engine***

Đây là một công cụ có chức năng quản lý việc lưu trữ, xử lý và bảo mật dữ liệu. Trong đây sẽ bao gồm rất nhiều các công cụ khác nhau như một công cụ lưu trữ quản lý các tệp, bảng, trang, chỉ mục, bộ đệm dữ liệu và giao dịch cơ sở dữ liệu.

##### ***External protocol***

Đây là các giao thức được sử dụng để giao tiếp với Database engine. Nó bao gồm TCP/IP hay VIA (Virtual Interface Adapter),...

### ***b) Các chức năng chính***

SQL Server Management Studio là một công cụ miễn phí được cung cấp bởi Microsoft để quản lý và cấu hình các phiên bản SQL Server. Nó cung cấp một giao diện đồ họa trực quan giúp người dùng dễ dàng thực hiện các tác vụ quản trị mà không cần phải viết mã T-SQL.

SQL Server Management Studio có một số chức năng chính:

#### **a) Quản lý cơ sở dữ liệu:**

- Tạo, chỉnh sửa và xóa cơ sở dữ liệu: cho phép bạn tạo cơ sở dữ liệu mới cho trang webtoon của bạn, chỉnh sửa cấu trúc của nó và xóa nó khi cần thiết.

- Sao lưu và phục hồi cơ sở dữ liệu: cung cấp các công cụ để sao lưu cơ sở dữ liệu của bạn, đảm bảo dữ liệu được bảo vệ trong trường hợp xảy ra sự cố.

- Quản lý người dùng và quyền truy cập: cho phép bạn tạo người dùng mới và cấp quyền truy cập vào cơ sở dữ liệu cho họ, đảm bảo chỉ những người được ủy quyền mới có thể truy cập dữ liệu.

- Giám sát hiệu suất cơ sở dữ liệu: cung cấp các công cụ để theo dõi hiệu suất của cơ sở dữ liệu, giúp bạn xác định và giải quyết các vấn đề hiệu suất tiềm ẩn.

#### **b) Phát triển ứng dụng:**

- Viết và thực thi truy vấn T-SQL: cung cấp một môi trường để viết và thực thi các truy vấn T-SQL, cho phép bạn truy xuất và thao tác dữ liệu trong cơ sở dữ liệu.

- Cài đặt và quản lý stored procedures: cho phép bạn cài đặt và quản lý stored procedures, giúp bạn tự động hóa các tác vụ truy vấn và thao tác dữ liệu.

- Gỡ lỗi stored procedures: cung cấp các công cụ để gỡ lỗi stored procedures, giúp bạn xác định và sửa lỗi trong code.

- Phân tích dữ liệu: cung cấp các công cụ để phân tích dữ liệu trong cơ sở dữ liệu, giúp bạn thu thập thông tin chi tiết về hiệu suất trang webtoon của bạn.

- Báo cáo: cung cấp các công cụ để tạo báo cáo dựa trên dữ liệu trong cơ sở dữ liệu, giúp bạn dễ dàng chia sẻ thông tin chi tiết với các bên liên quan.

c) Tích hợp:

- Hỗ trợ nhiều phiên bản SQL Server: hỗ trợ nhiều phiên bản SQL Server, cho phép bạn sử dụng nó với bất kỳ phiên bản nào bạn đang sử dụng.

- Tích hợp với Visual Studio: tích hợp với Visual Studio, cho phép bạn dễ dàng phát triển các ứng dụng webtoon sử dụng SQL Server.

- Hỗ trợ PowerShell: hỗ trợ PowerShell, cho phép bạn tự động hóa các tác vụ quản trị cơ sở dữ liệu.

d) Dễ sử dụng:

- Giao diện đồ họa trực quan: cung cấp một giao diện đồ họa trực quan, giúp bạn dễ dàng sử dụng các chức năng của nó.

- Trợ giúp ngữ cảnh: cung cấp trợ giúp ngữ cảnh, giúp bạn dễ dàng tìm hiểu cách sử dụng các chức năng của nó.

- Nhiều tùy chọn tùy chỉnh: cung cấp nhiều tùy chọn tùy chỉnh, cho phép bạn điều chỉnh giao diện và chức năng của nó theo nhu cầu của bạn.

Ngoài ra, SQL Server Management Studio còn cung cấp một số tính năng nâng cao như:

- Quản lý các phiên bản SQL Server trên Azure
- Triển khai các gói Integration Services
- Quản lý các phiên bản SQL Server Analysis Services

SQL Server Management Studio là một công cụ mạnh mẽ và linh hoạt cho phép người dùng quản lý và cấu hình SQL Server một cách hiệu quả. Nó là một công cụ

cần thiết cho bất kỳ ai làm việc với SQL Server. (Lưu ý: SQL Server Management Studio chỉ chạy trên hệ điều hành Windows).

## **CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

### **3.1 Phát biểu bài toán**

Bài toán của đề tài "Xây dựng ứng dụng thực hiện quy trình xuất bản cho tạp chí khoa học" là thiết kế và phát triển một ứng dụng nhằm tự động hóa và tối ưu hóa quy trình xuất bản khoa học. Quy trình này hiện tại thường phức tạp và tốn nhiều thời gian, bao gồm nhiều bước như nhận bài, phân công người đánh giá, tổng hợp ý kiến phản hồi, chỉnh sửa, duyệt bài và xuất bản. Bài toán cần giải quyết là tạo ra một hệ thống có thể quản lý hiệu quả tất cả các bước này, cải thiện chất lượng và tốc độ xuất bản, đảm bảo tuân thủ các tiêu chuẩn xuất bản khoa học, và nâng cao trải nghiệm của người dùng như tác giả, biên tập viên và người đánh giá, đồng thời tích hợp công nghệ tiên tiến như trí tuệ nhân tạo để hỗ trợ kiểm tra chất lượng bài viết và phát hiện đạo văn.

### **3.2 Các chức năng của hệ thống**

- Quản lý số tạp chí đã xuất bản
- Quản lý số tạp chí chưa xuất bản
- Quản lý bài viết tác giả
- Quản lý bài cần phản biện
- Quản lý tài khoản
- Đăng nhập, đăng ký

#### **3.2.1 Chức năng dành cho tác giả**

- Chức năng đăng nhập để lưu các trạng thái mà tác giả thao tác.
  - + , gửi bài lên biên tập viên
  - + , xem danh sách bài được trả về.
- Quản lý bài viết tác giả.

#### **3.2.2 Chức năng dành cho biên tập viên**

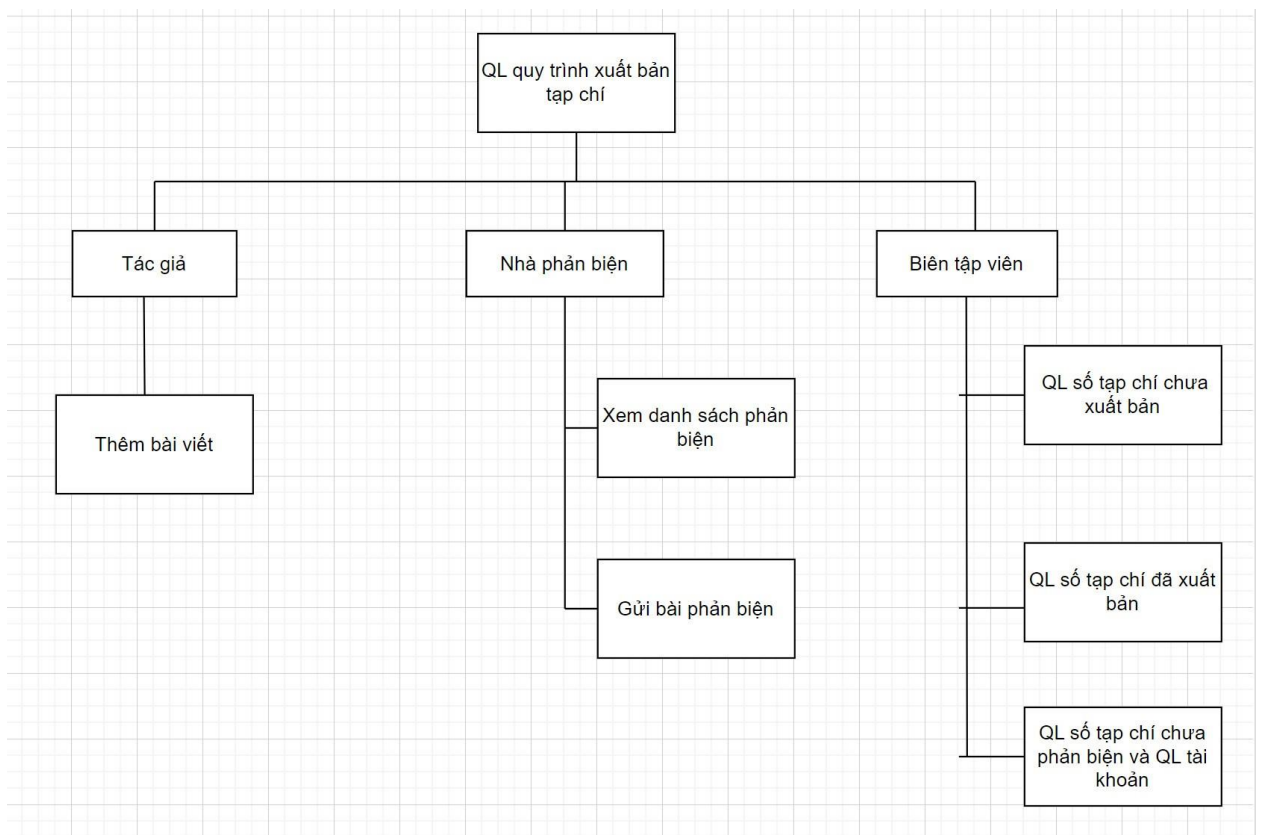
- Chức năng đăng nhập để lưu các trạng thái mà biên tập viên thao tác.
- quản lý số tạp chí đã và chưa xuất bản
- quản lý tài khoản , quyền

### 3.2.3 Chức năng dành cho nhà phản biện

- Chức năng đăng nhập để lưu các trạng thái mà nhà phản biện thao tác.
- quản lý số tạp chí cần phản biện

## 3.3 Mô hình phân cấp chức năng.


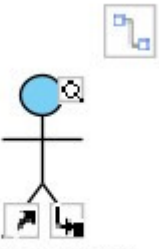
Mô hình phân cấp chức năng là một cách tiếp cận tổ chức công việc trong đó nhiệm vụ và trách nhiệm được phân chia thành các cấp độ. Tại mỗi cấp độ, có các vai trò và chức năng cụ thể, từ cấp quản lý cao nhất đến cấp nhân viên thực thi nhiệm vụ.




Hình 3. 1: Mô hình phân cấp chức năng

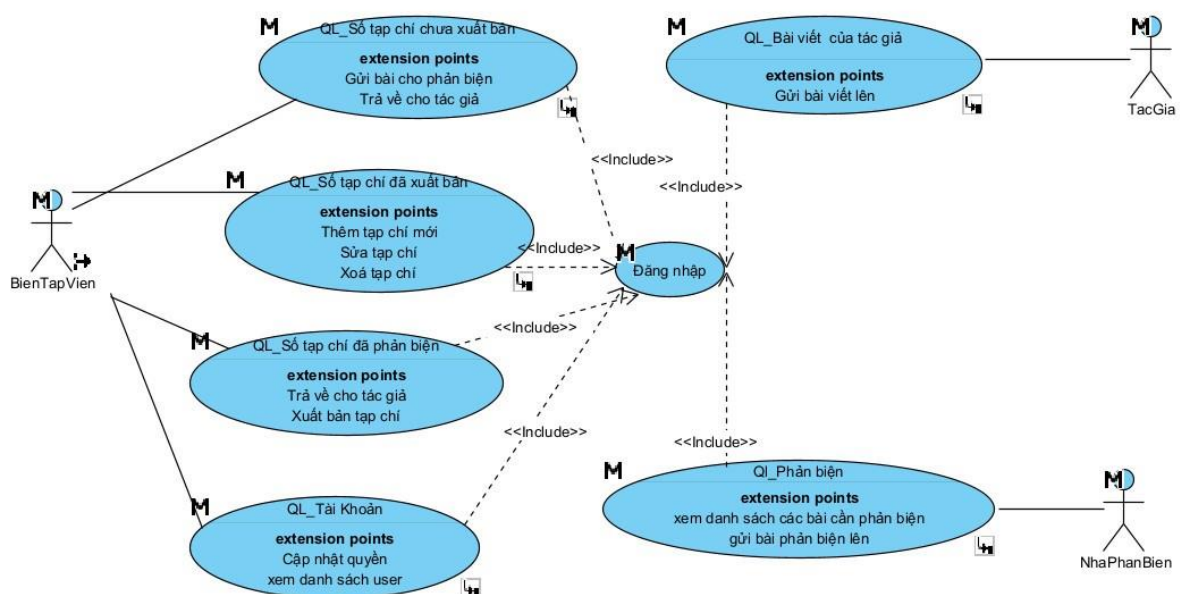
### 3.4 Phân tích thiết kế use case theo UML

#### 3.4.1 Giới thiệu về các Actor

Tên actor	Mô tả
 Tác giả	<p>Trong hệ thống quản lý quy trình xuất bản tạp chí, vai trò của tác giả là một phần quan trọng trong quy trình. Đầu tiên, tác giả có thể đăng ký tài khoản trên hệ thống để gửi bài báo hoặc quản lý các bài báo đã được gửi. Thứ hai, tác giả có thể theo dõi trạng thái của bài báo của mình, từ quá trình nộp, kiểm tra và đánh giá cho đến quyết định về việc xuất bản. Cuối cùng, tác giả có thể tương tác với biên tập viên hoặc các thành viên khác trong quy trình, bằng cách thảo luận và cung cấp phản hồi để cải thiện chất lượng bài báo và quy trình xuất bản tổng thể. Như vậy, vai trò của tác giả là một phần không thể thiếu trong việc thúc đẩy và duy trì chất lượng của tạp chí khoa học.</p>
 Biên tập viên	<p>Trong hệ thống quản lý quy trình xuất bản tạp chí, biên tập viên đóng vai trò quan trọng trong việc đảm bảo chất lượng của các bài báo được xuất bản. Đầu tiên, biên tập viên có nhiệm vụ xem xét và đánh giá các bài báo được gửi đến, quyết định liệu chúng có đáp ứng được tiêu chuẩn của tạp chí không. Thứ hai, biên tập viên thực hiện việc chỉnh sửa và sắp xếp các bài báo để đảm bảo tính rõ ràng và mạch lạc. Cuối cùng, họ có trách nhiệm tương tác với tác giả, cung cấp phản hồi và hướng dẫn để cải thiện nội dung và định hình bài báo theo hướng phát triển tích cực. Với vai trò này, biên tập viên đóng vai trò quan trọng trong việc đảm bảo chất lượng và uy tín của tạp chí khoa học.</p>

 <p>Nhà phản biện</p>	<p>Trong quy trình xuất bản tạp chí, vai trò của nhà phản biện là một phần quan trọng của quá trình đánh giá và cải thiện chất lượng của các bài báo. Đầu tiên, nhà phản biện được giao nhiệm vụ đọc và đánh giá các bài báo được gửi đến, đưa ra nhận xét và đề xuất cho biên tập viên về việc chấp nhận hoặc từ chối bài báo. Thứ hai, họ cung cấp đánh giá chuyên sâu về nội dung và phương pháp nghiên cứu của bài báo, đảm bảo tính đúng đắn và chất lượng của thông tin được trình bày. Cuối cùng, nhà phản biện có trách nhiệm tương tác với tác giả và biên tập viên để giải quyết mọi thắc mắc hoặc đề xuất cải thiện trong quá trình phản biện và biên tập. Với vai trò này, nhà phản biện đóng vai trò quan trọng trong việc đảm bảo sự khách quan và chất lượng của các bài báo được xuất bản.</p>
--	--

### 3.4.2 Biểu đồ Usecase tổng quát



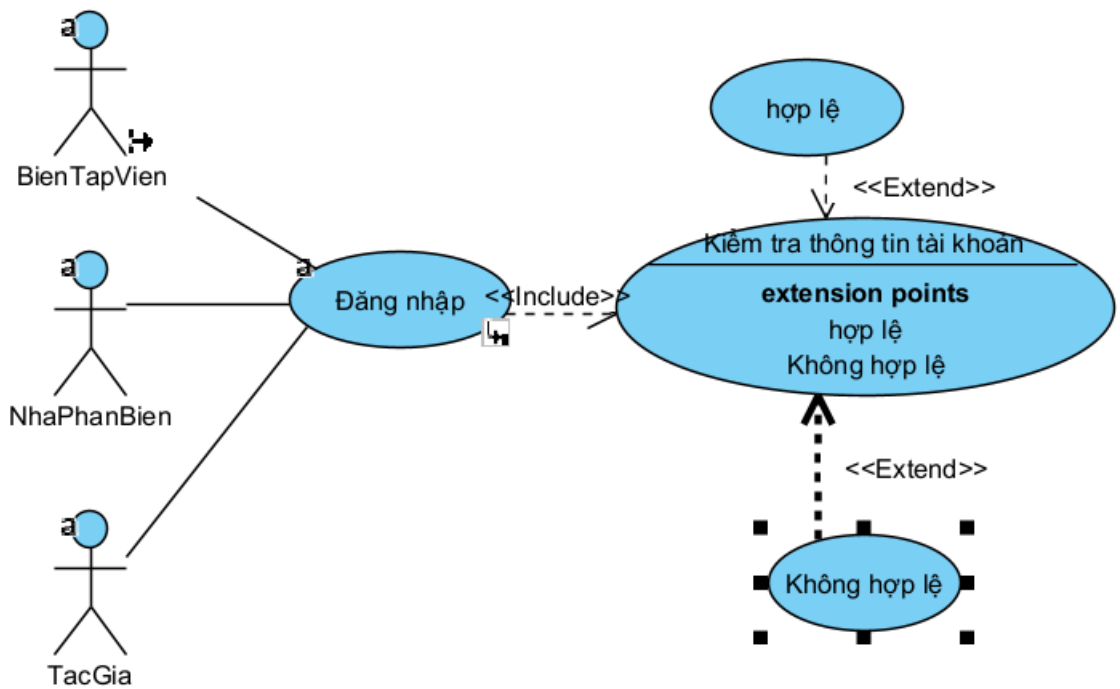
Hình 3. 2: Sơ đồ Usecase tổng quát

STT	Tên Use case	Ý nghĩa/Ghi Chú
1	Đăng nhập	Use case này giúp người dùng hoặc Admin sử dụng các chức năng của hệ thống cần đến quyền truy cập.



		- Đối với tác giả, phản biện , biên tập viên có thể truy cập vào các trang quản lý.
2	Quản lý số tạp chí đã xuất bản	Biên tập viên sẽ đăng các bài đã được phản biện lên trang chủ
3	Quản lý số tạp chí chưa xuất bản	Biên tập viên sẽ xem xét và chuyển các bài tạp chí sang cho nhà phản biện
4	Quản lý số tạp chí đã phản biện	Phản biện sẽ gửi bài vào đây và biên tập viên sẽ xem xét rồi chuyển sang số tạp chí đã xuất bản
5	Quản lý bài viết tác giả	Tác giả sẽ gửi bài lên biên tập và sẽ xem được các bài biên tập trả về
6	Quản lý bài cần phản biện	Phản biện sẽ xem danh sách các bài cần phản biện và phản biện các bài đã chọn
7	Quản lý tài khoản	Biên tập viên sẽ thêm mới và sửa quyền tài khoản

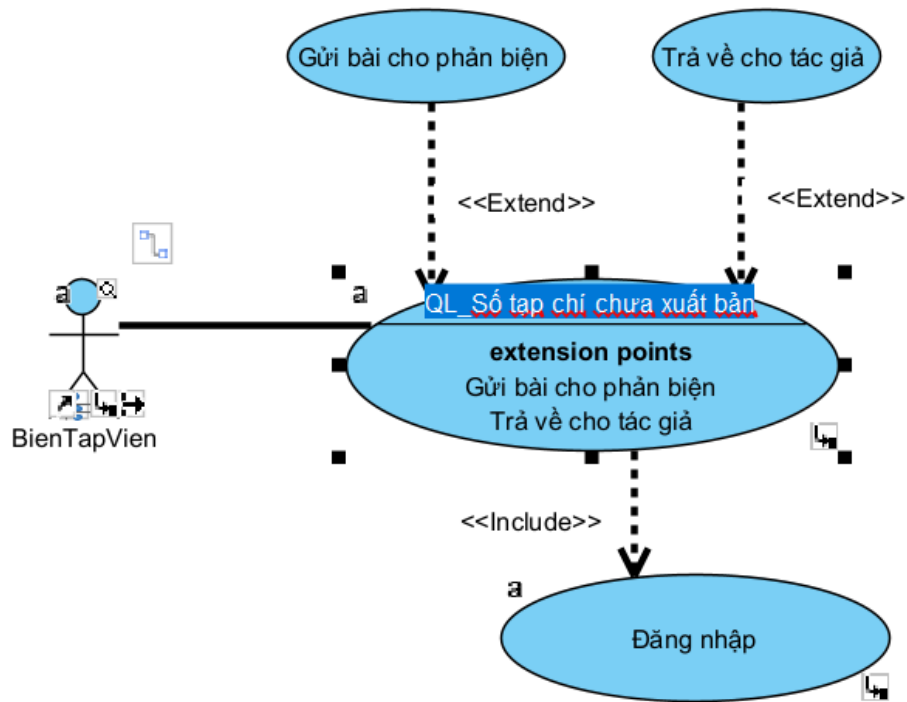
### 3.4.3 Use case đăng nhập



Hình 3. 3: Sơ đồ Use case đăng nhập

Actor	Tên Use case	Ý nghĩa/Ghi chú
Người dùng	Đăng nhập	Use case này mô tả người dùng đăng nhập. Người dùng nhập thông tin tài khoản mật khẩu, hệ thống sẽ kiểm tra thông tin của tài khoản. Nếu tài khoản đúng cho phép đăng nhập, nếu sai bắt nhập lại.
	Tác giả	Đăng nhập vào trang của tác giả
	Biên tập viên	Đăng nhập vào trang của biên tập viên
	Phản biện	Đăng nhập vào trang của nhà phản biện

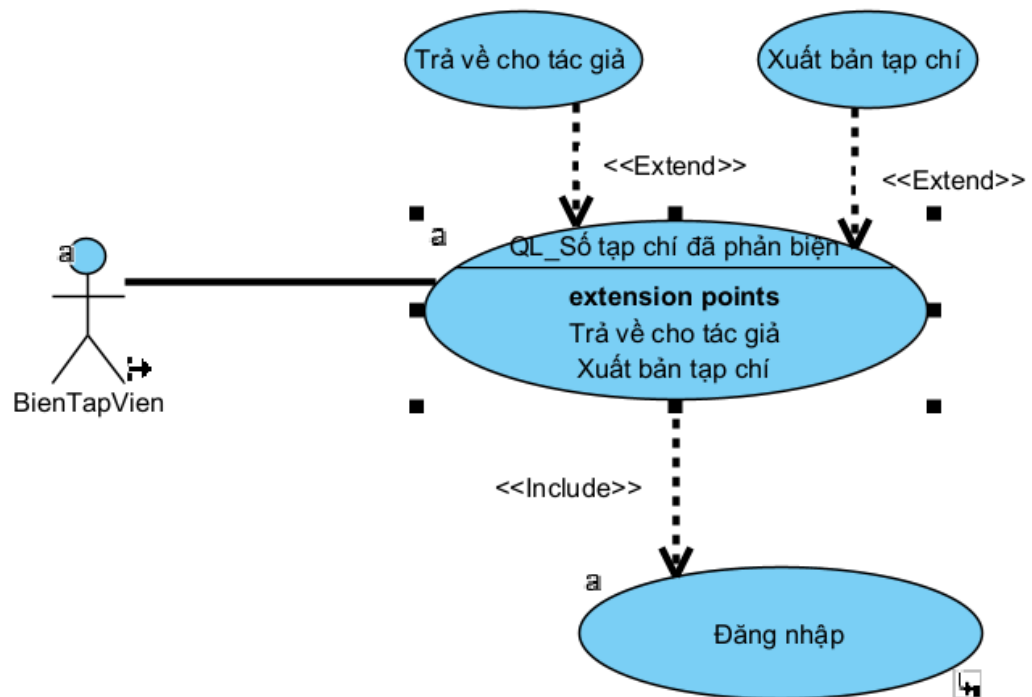
#### 3.4.4 Use case quản lý số tạp chí chưa xuất bản



Hình 3. 4: Use case quản lý số tạp chí chưa xuất bản

Actor	Tên Use case	Ý nghĩa/Ghi chú
Biên tập viên	Quản lý	Biên tập có quyền thêm sửa xoá bài tạp chí
	Đăng nhập	Đăng nhập vào quyền biên tập

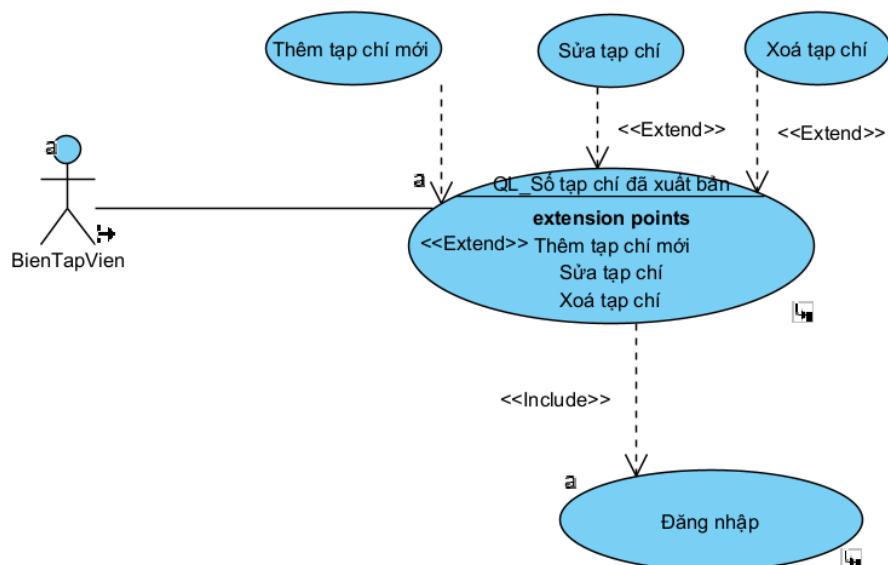
### 3.4.5 Use case quản lý số tạp chí đã phản biện



Hình 3. 5: Use case quản lý số tạp chí đã phản biện

Actor	Tên Use case	Ý nghĩa/Ghi chú
Biên tập viên	Quản lý	Use case này quản lý số tạp chí đã phản biện
	Đăng nhập	Bắt buộc phải đăng nhập biên tập viên.

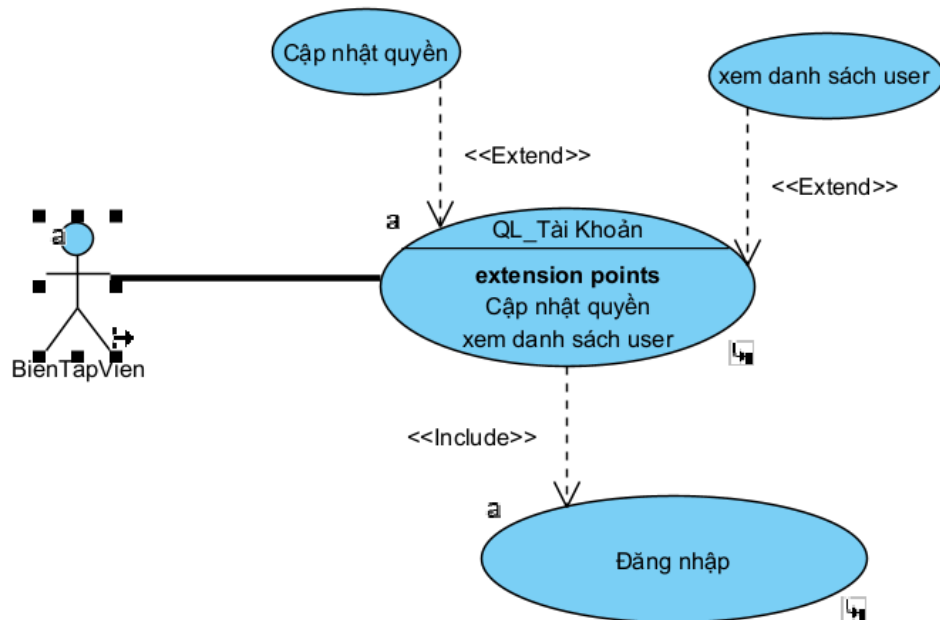
### 3.4.6 Use case quản lý số tạp chí đã xuất bản



Hình 3. 6: Use case quản lý số tạp chí đã xuất bản

Actor	Tên Use case	Ý nghĩa/Ghi chú
biên tập viên	Quản lý	Use case này quản lý số tạp chí đã xuất bản
	Đăng nhập	Bắt buộc phải đăng nhập.

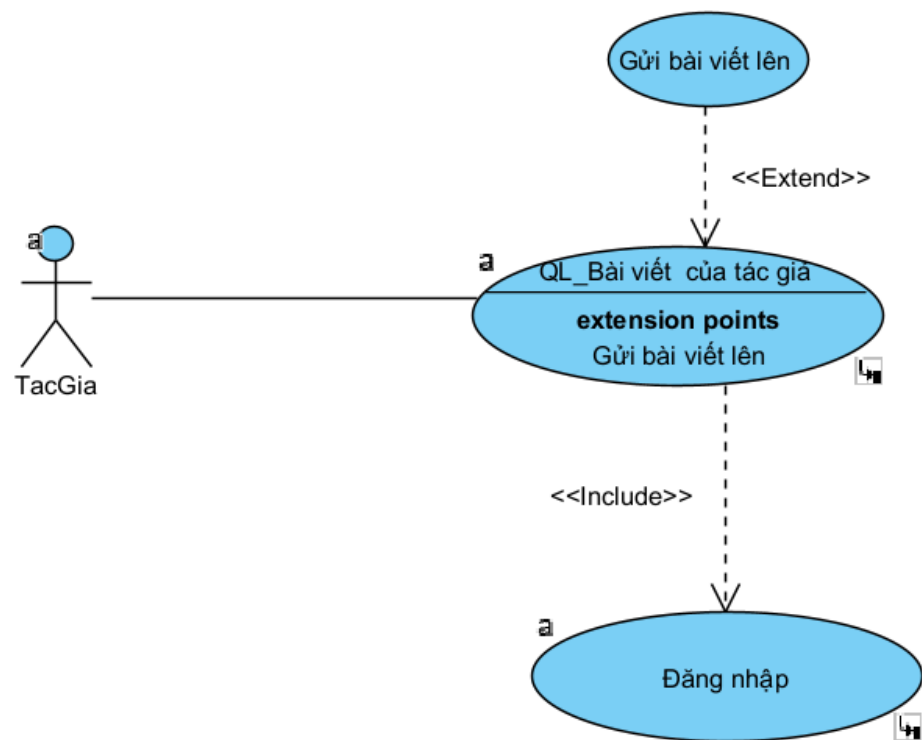
### 3.4.7 Use case quản lý tài khoản



Hình 3. 7: Use case tài khoản

Actor	Tên Use case	Ý nghĩa/Ghi chú
Biên tập viên	Biên tập viên	Quản lý tài khoản với quyền thêm tài khoản và sửa quyền
	Đăng nhập	Bắt buộc phải đăng nhập biên tập

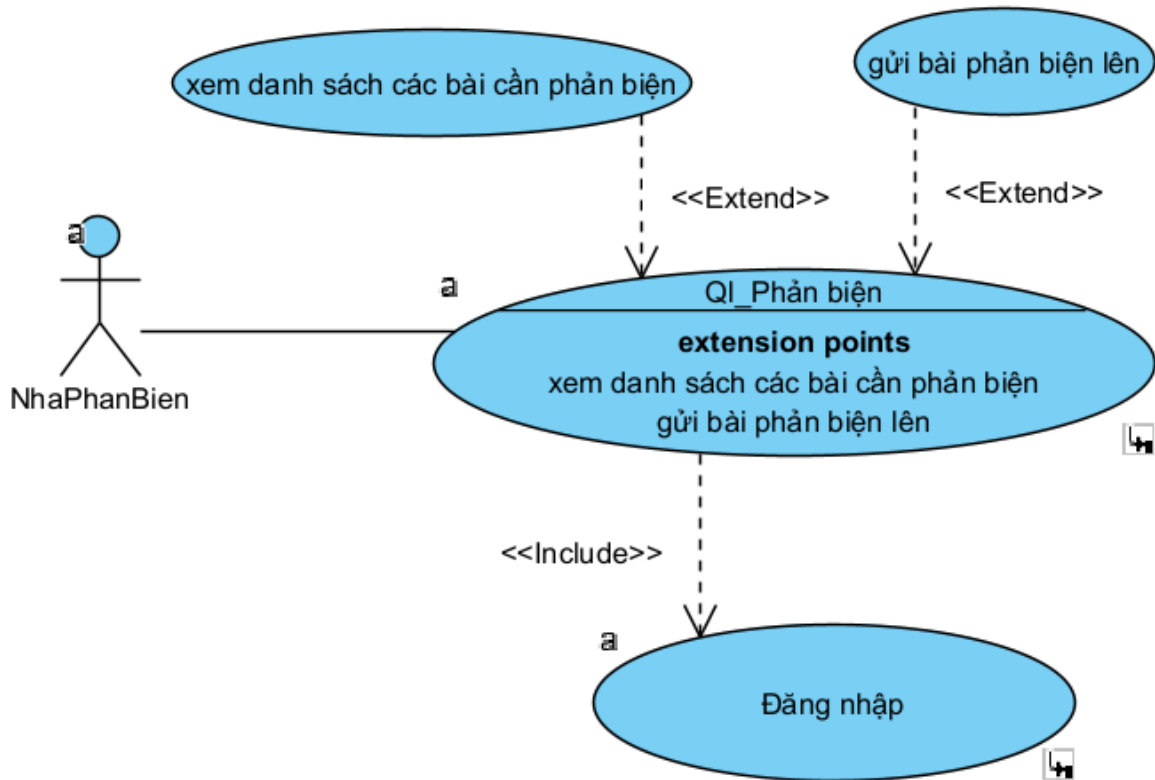
### 3.4.8 Use case quản lý bài viết tác giả



Hình 3. 8: Use case quản lý bài viết tác giả

Actor	Tên Use case	Ý nghĩa/Ghi chú
Tác giả	Quản lý bài viết tác giả	Tác giả sẽ xem danh sách bài được trả về và có thể gửi bài lên cho biên tập
	Đăng nhập	Bắt buộc phải đăng nhập

### 3.4.9 Use case quản lý phản biện

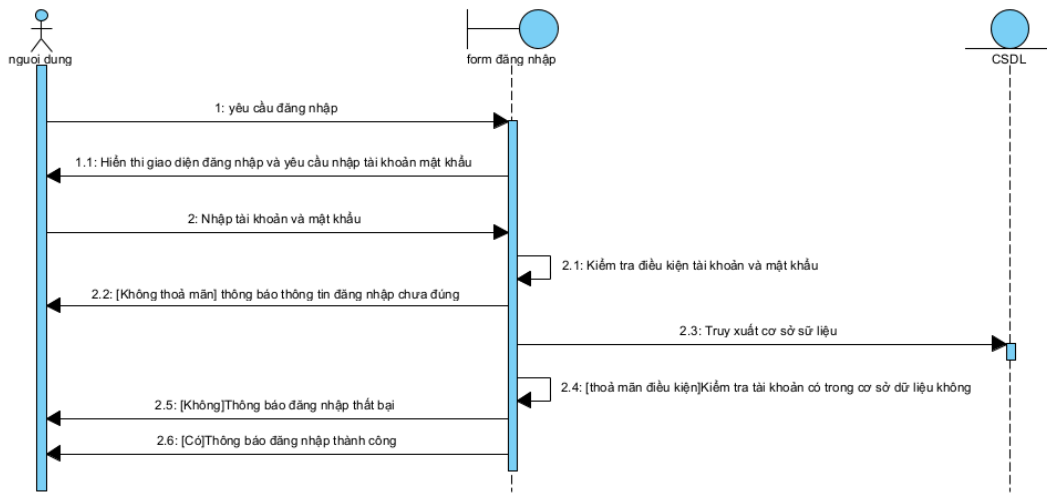


Hình 3. 9: Use case QL sản phẩm

Actor	Tên Use case	Ý nghĩa/Ghi chú
Phản biện	Quản lý phản biện	Phản biện có thể xem danh sách và chấp nhận phản biện
	Đăng nhập	Bắt buộc phải đăng nhập tài khoản

### 3.5 Sơ đồ tuần tự của các use case theo UML

#### 3.5.1 Sơ đồ tuần tự của use case đăng nhập

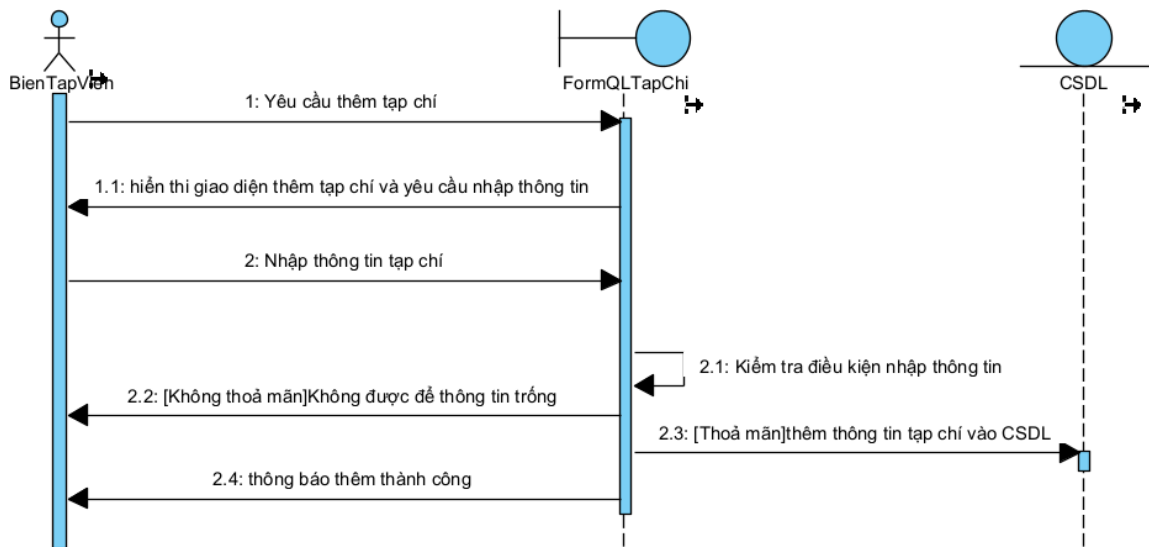


Hình 3. 10: Sơ đồ tuần tự của use case đăng nhập

➤ Tóm tắt: Người dùng sẽ đăng nhập vào hệ thống bằng tên đăng nhập và mật khẩu của mình. Nếu tên đăng nhập và mật khẩu đúng sẽ được truy cập vào hệ thống và thực hiện các chức năng mua bán. Nếu tài khoản và mật khẩu sai thì sẽ thông báo cho người dùng biết là tài khoản mật khẩu sai và bắt nhập lại.



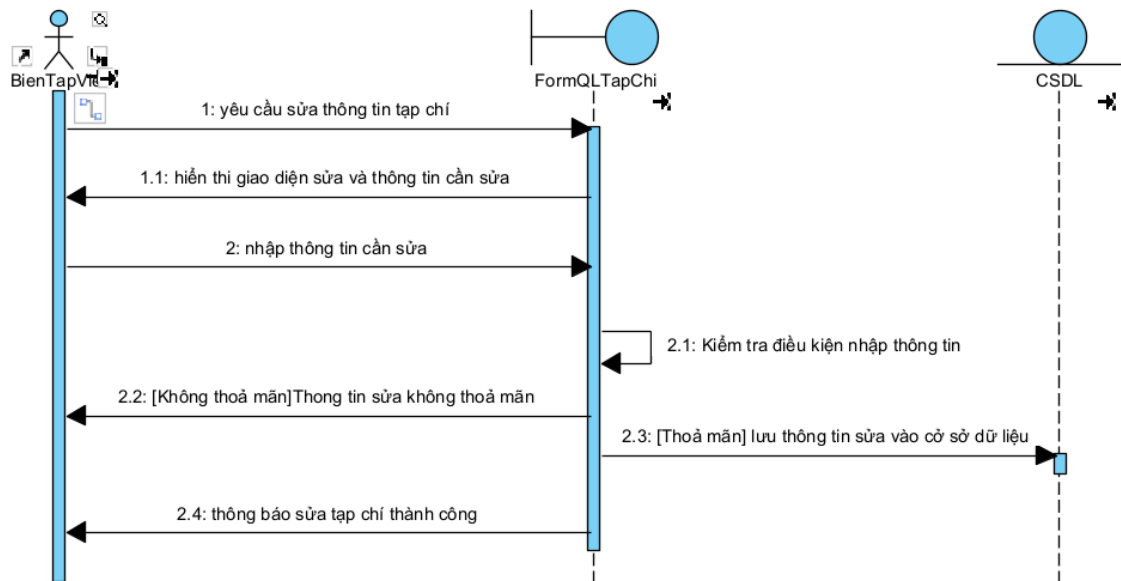
### 3.5.2 Sơ đồ tuần tự của use case thêm tạp chí của quản lý số tạp chí đã xuất bản



Hình 3. 11: Sơ đồ tuần tự của use case thêm tạp chí của quản lý số tạp chí đã xuất bản

➤ Tóm tắt: Biên tập viên sẽ ấn vào nút thêm tạp chí giao diện trả về các thông tin để biên tập viên nhập vào sau khi ấn nút ok thì sẽ kiểm tra điều kiện nếu còn ô nào bị bỏ trống sẽ trả về không được bỏ trống thông tin còn nếu đã điền đầy đủ thông tin sẽ lưu vào cơ sở dữ liệu

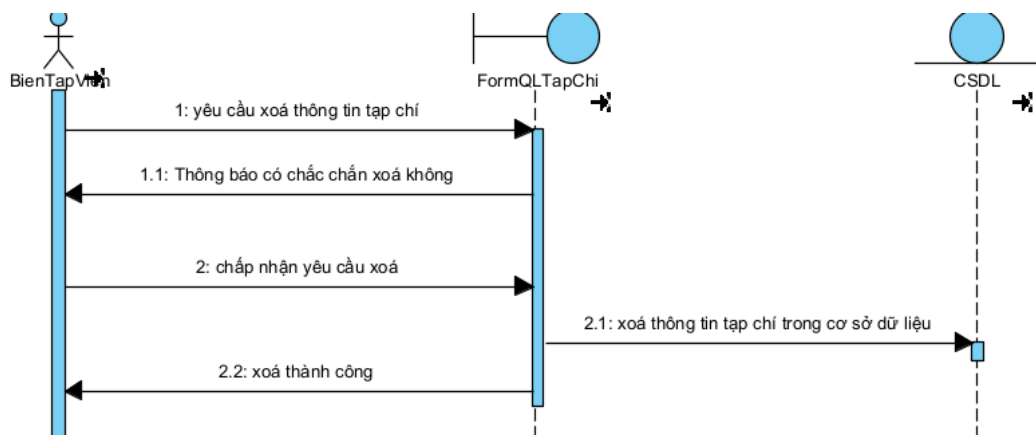
### 3.5.3 Sơ đồ tuần tự use case sửa tạp chí của quản lý số tạp chí đã xuất bản



Hình 3. 12: Sơ đồ tuần tự use case sửa tạp chí

➤ Tóm tắt: Biên tập viên sẽ ấn vào nút sửa tạp chí giao diện trả về các thông tin để biên tập viên nhập vào sau khi ấn nút ok thì sẽ kiểm tra điều kiện nếu còn ô nào bị bỏ trống sẽ trả về không được bỏ trống thông tin còn nếu đã điền đầy đủ thông tin sẽ lưu vào cơ sở dữ liệu

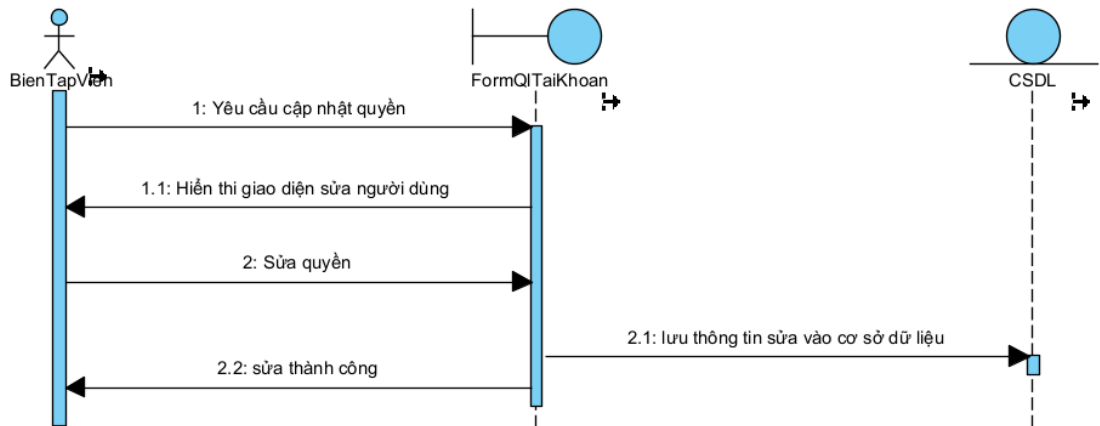
### 3.5.4 Sơ đồ tuần tự use case xoá tạp chí trong quản lý số tạp chí đã xuất bản



Hình 3. 13: Sơ đồ tuần tự use xoá số tạp chí đã xuất bản

➤ Tóm tắt: Biên tập viên sẽ ấn vào nút xoá tạp chí giao diện trả trang hỏi bạn có chắc chắn muốn xoá nếu đồng ý sẽ xoá dữ liệu còn nếu không đồng ý thì sẽ trở lại trang

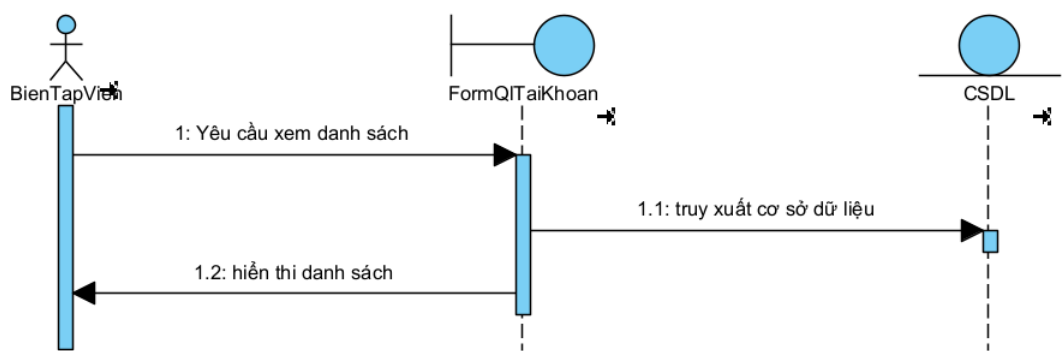
### 3.5.5 Sơ đồ tuần tự use case cập nhật quyền



Hình 3. 14: Sơ đồ tuần tự use case cập nhật quyền

➤ Tóm tắt: Biên tập viên sẽ ấn vào nút sửa thông tin người dùng hệ thống sẽ hiển thị lên danh sách quyền mà người dùng có thể sửa sau khi ấn đồng ý thì hệ thống sẽ lưu lên cơ sở dữ liệu

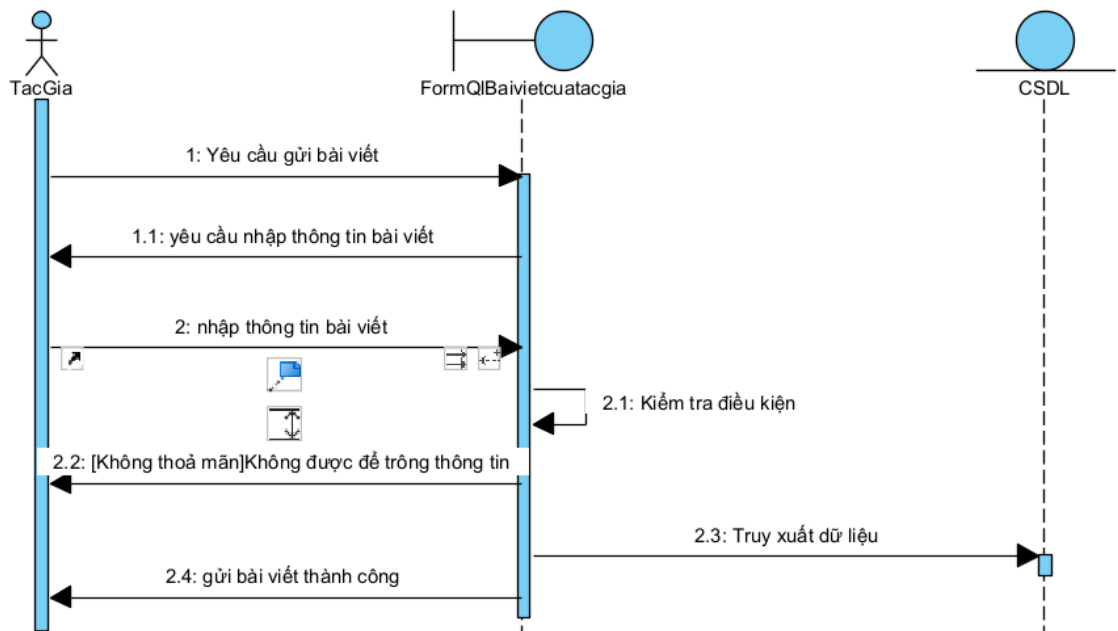
### 3.5.6 Sơ đồ tuần tự use case xem danh sách user



Hình 3.20: Sơ đồ tuần tự use case xem danh sách quyền

➤ Tóm tắt: Biên tập viên sẽ ấn vào danh sách user hệ thống sẽ truy xuất cơ sở dữ liệu sau đó hiển thị ra thông tin danh sách người dùng

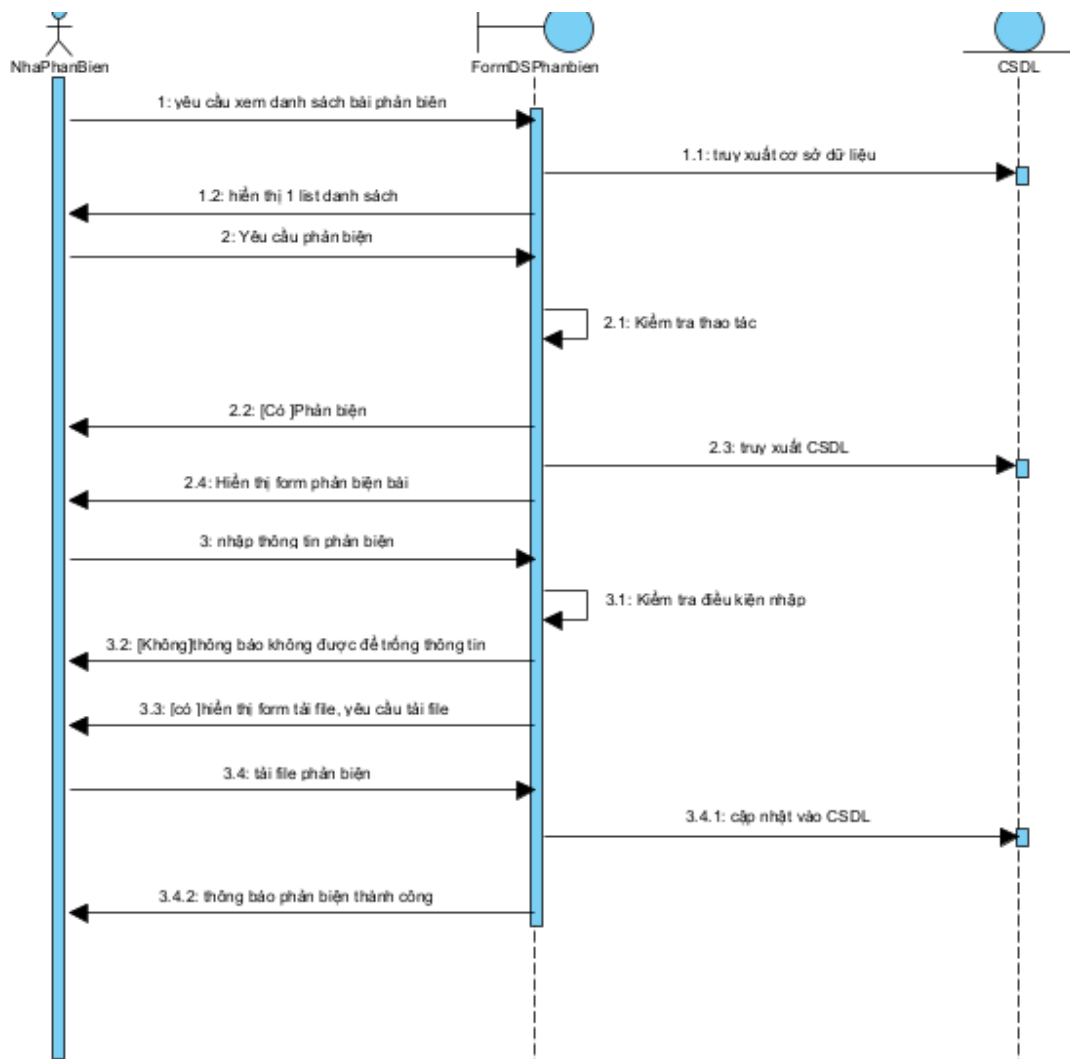
### 3.5.7 Sơ đồ tuần tự use case gửi bài viết



Hình 3.21: Sơ đồ tuần tự use case gửi bài viết

➤ Tóm tắt: Tác giả ấn vào nút thêm bài mới hệ thống sẽ trả về form nhập thông tin bài viết sau khi nhập đầy đủ thông tin tác giả sẽ ấn nút tiếp tục và hệ thống sẽ chuyển sang trang khác để tác giả nhập tiếp thông tin và cuối cùng sẽ ấn tiếp tục để hiển thị lên trang tải file pdf sau khi hoàn tất hệ thống sẽ truy xuất vào cơ sở dữ liệu và trả về thông báo gửi bài viết thành công cho tác giả

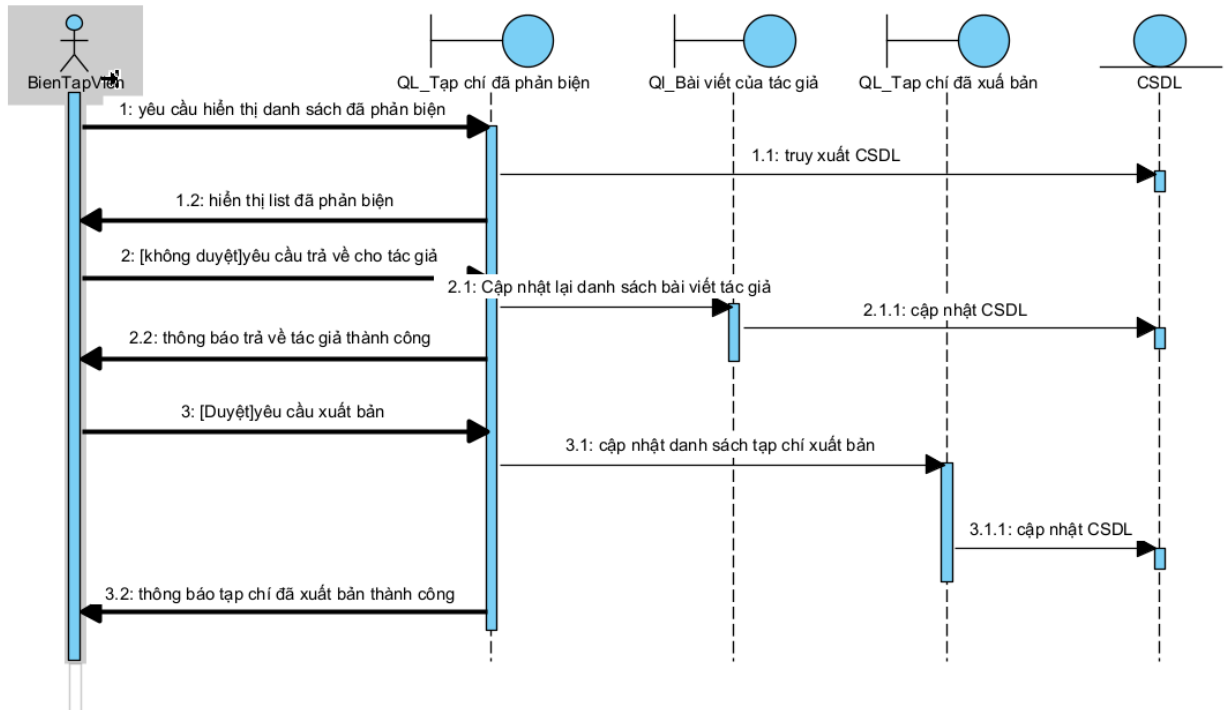
### 3.5.8 Sơ đồ tuần tự use case phản biện bài



Hình 3.22: Sơ đồ tuần tự use case phản biện bài

➤ Tóm tắt: Sau khi đăng nhập vào tài khoản nhà phản biện hệ thống sẽ trả về danh sách bài nhà phản biện cần phải phản biện sau đó nhà phản biện sẽ chọn bài cần phản biện sau đó sẽ ấn vào nút sửa , hệ thống sẽ trả về 1 form hiển thị ngày hạn biên tập viên giao và sau đó biên tập viên ấn tiếp tục và gửi file phản biện của nhà phản biện lên

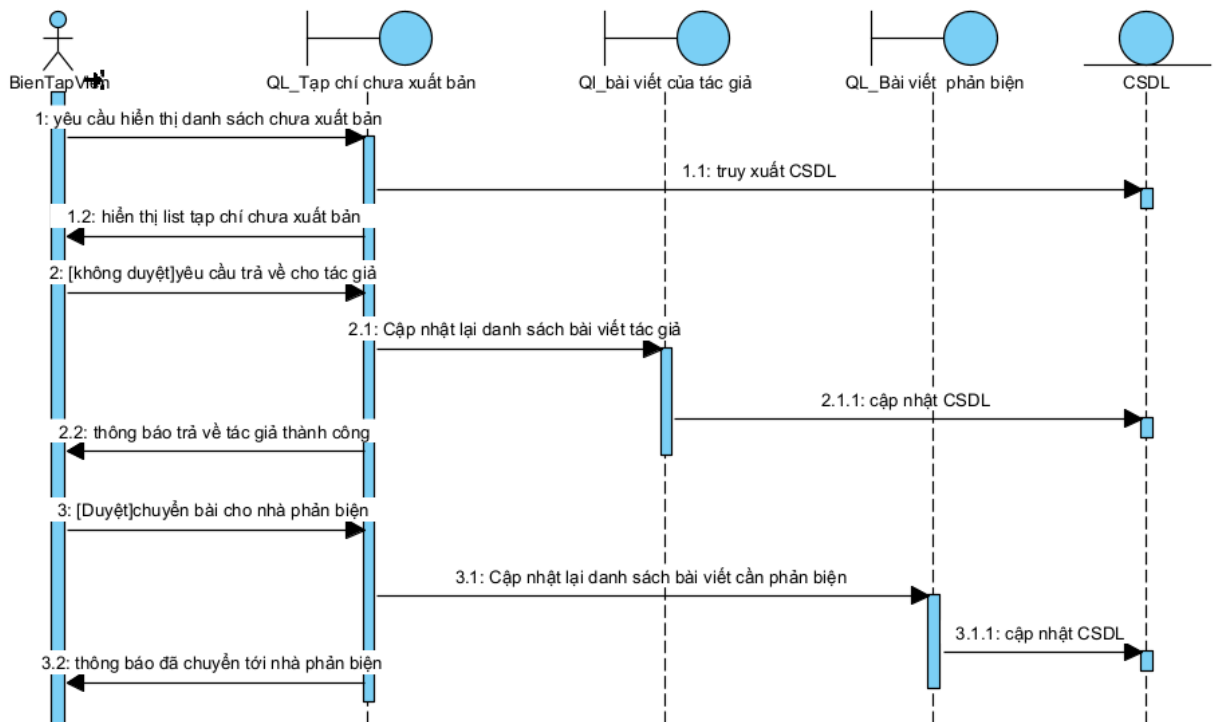
### 3.5.9 Sơ đồ tuần tự use case quản lý tạp chí đã phản biện



Hình 3.23: Sơ đồ tuần tự use case quản lý số tạp chí đã phản biện

➤ Tóm tắt: sau khi ấn vào trang danh sách tạp chí đã phản biện hệ thống sẽ truy xuất cơ sở dữ liệu và trả về 1 danh sách các bài đã phản biện xog biên tập viên sẽ xem xét và trả về cho tác giả hoặc chuyển sang trang tạp chí đã xuất bản

### 3.5.10 Sơ đồ tuần tự use case quản lý tạp chí chưa xuất bản



Hình 3.23: Sơ đồ tuần tự use case quản lý số tạp chí chưa xuất bản

➤ Tóm tắt: sau khi ấn vào trang danh sách tạp chí chưa xuất bản hệ thống sẽ truy xuất cơ sở dữ liệu và trả về 1 danh sách các bài chưa xuất bản xong biên tập viên sẽ ấn vào nút gửi sau đó chọn nhà phản biện và nhập thông tin ngày giờ hạn phản biện sau đó sẽ gửi sang cho nhà phản biện đã chọn

### 3.6 Thiết kế cơ sở dữ liệu

Bảng 1: User (Tài khoản)

Thuộc tính	Kiểu dữ liệu
Id	uniqueidentifier
FullName	nvarchar(MAX)
UserName	nvarchar(256)
NormalizedUserName	nvarchar(256)
Email	nvarchar(256)
NormalizedEmail	nvarchar(256)
EmailConfirmed	bit
PasswordHash	nvarchar(MAX)
SecurityStamp	nvarchar(MAX)

ConcurrencyStamp	nvarchar(MAX)
PhoneNumber	nvarchar(MAX)
PhoneNumberConfirmed	bit
TwoFactorEnabled	bit
LockoutEnd	datetimeoffset(7)
LockoutEnabled	bit
AccessFailedCount	bit

Bảng 3. 1: Bảng User

Bảng 2: Bảng bài viết

Thuộc tính	Kiểu dữ liệu
Id	uniqueidentifier
NgonNgu	nvarchar(MAX)
ChuyenMuc	nvarchar(MAX)
LoiNhanChoBanBientap	nvarchar(MAX)
Tieude	nvarchar(MAX)
TomTat	nvarchar(MAX)
SDT	int
Thongtinbosungtukhoa	nvarchar(MAX)
TaiLieuThamKhao	nvarchar(MAX)
TrangThai	nvarchar(MAX)
ThuocTapChi	nvarchar(MAX)
IdTacGia	uniqueidentifier

Bảng 3. 2: Bảng bài viết

Bảng 3: Approle

Thuộc tính	Kiểu dữ liệu
Id	uniqueidentifier
Name	nvarchar(MAX)
NormalizedName	nvarchar(MAX)
ConcurrencyStamp	nvarchar(MAX)
RoleCode	nvarchar(MAX)

Bảng 3. 3: Bảng approle



*Bảng 4: ApproleClaim*

Thuộc tính	Kiểu dữ liệu
Id	int
RoleId	uniqueidentifier
ClaimType	nvarchar(MAX)
ClaimValue	nvarchar(MAX)

*Bảng 3.4: Bảng approleclaim**Bảng 5: userclaim*

Thuộc tính	Kiểu dữ liệu
Id	int
UserId	uniqueidentifier
ClaimType	nvarchar(MAX)
ClaimValue	nvarchar(MAX)

*Bảng 3.5: Bảng userclaim**Bảng 6: Appuserlogin*

Thuộc tính	Kiểu dữ liệu
LoginProvider	nvarchar(MAX)
ProviderKey	nvarchar(MAX)
ProviderDisplayName	nvarchar(MAX)
UserId	uniqueidentifier

*Bảng 3.6: Bảng Appuserlogin**Bảng 7: Appuserrole*

Thuộc tính	Kiểu dữ liệu
Id	uniqueidentifier
UserId	uniqueidentifier
RoleId	uniqueidentifier

*Bảng 3.7: Bảng Appuserrole*

*Bảng 8: Appusertoken*

Thuộc tính	Kiểu dữ liệu
UserId	uniqueidentifier
LoginProvider	nvarchar(MAX)
Name	nvarchar(MAX)
Value	nvarchar(MAX)

*Bảng 3.8: Bảng Appusertoken**Bảng 9: Appnetuserroles*

Thuộc tính	Kiểu dữ liệu
UserId	uniqueidentifier
RoleId	uniqueidentifier

*Bảng 3.9: Bảng Appnetuserrole**Bảng 10: Bảng biên tập viên*

Thuộc tính	Kiểu dữ liệu
Id	uniqueidentifier
AnhBia	nvarchar(MAX)
Filepdf	nvarchar(MAX)
TenTacGia	nvarchar(MAX)
TenTapChi	nvarchar(MAX)
NgayPhatHanh	datetime2(7)

*Bảng 3.10: Bảng bientapvien**Bảng 11: Bảng FileManager*

Thuộc tính	Kiểu dữ liệu
Id	uniqueidentifier
Name	nvarchar(MAX)
Path	nvarchar(MAX)
Mime	nvarchar(MAX)
ParentId	uniqueidentifier
Size	float

*Bảng 3.12: Bảng Filemanager*

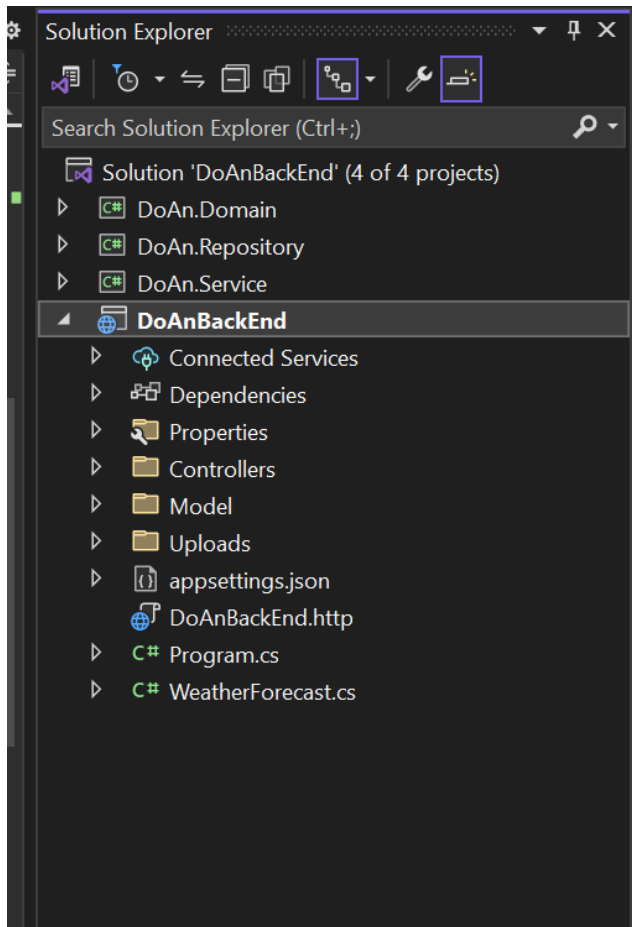
*Bảng 12: Bảng nhaphanbien*

Thuộc tính	Kiểu dữ liệu
Id	uniqueidentifier
NgayBTVGiao	datetime2(7)
NgayDenHanPhanHoi	datetime2(7)
NgayDenHanPhanBien	datetime2(7)
DeXuat	nvarchar(MAX)
IdBaiViet	nvarchar(MAX)
IdBienTapvien	nvarchar(MAX)
IdNhaPhanBien	nvarchar(MAX)

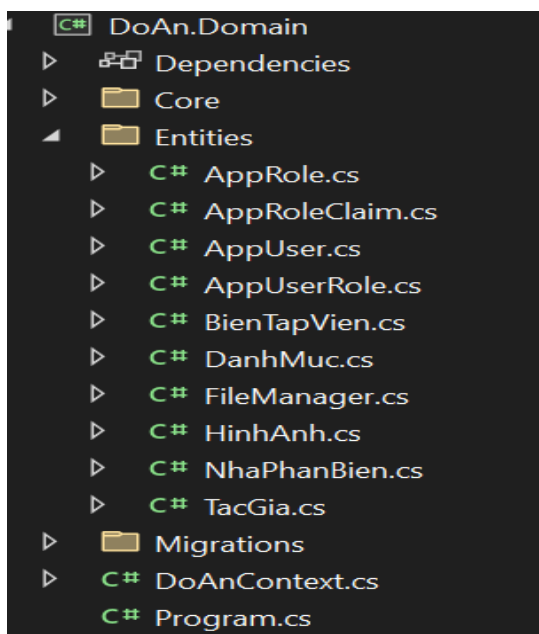
*Bảng 3.12: Bảng nhaphanbien*

## CHƯƠNG IV. XÂY DỰNG VÀ TRIỂN KHAI CHƯƠNG TRÌNH

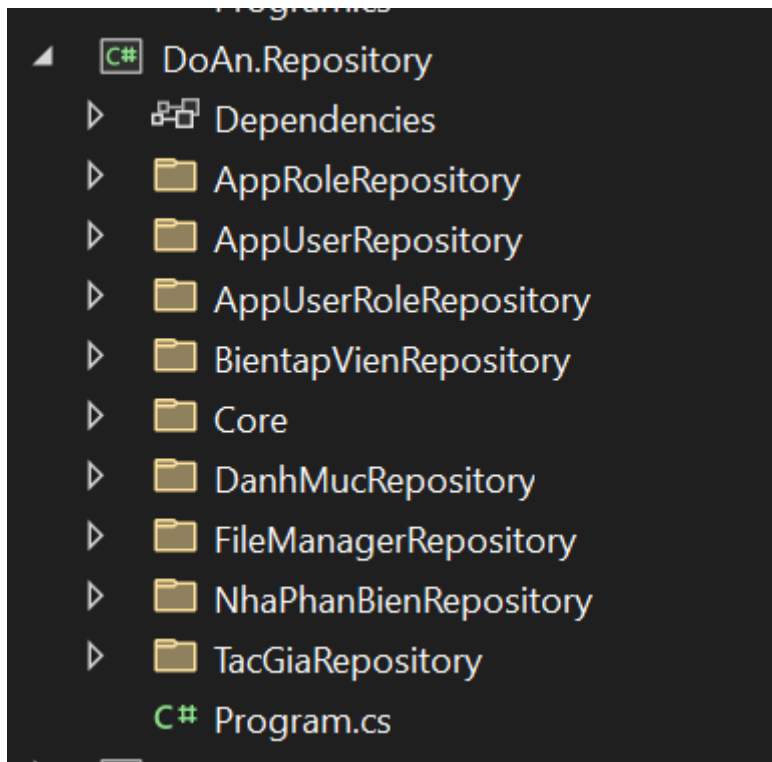
### 4.1 Cấu trúc lập trình



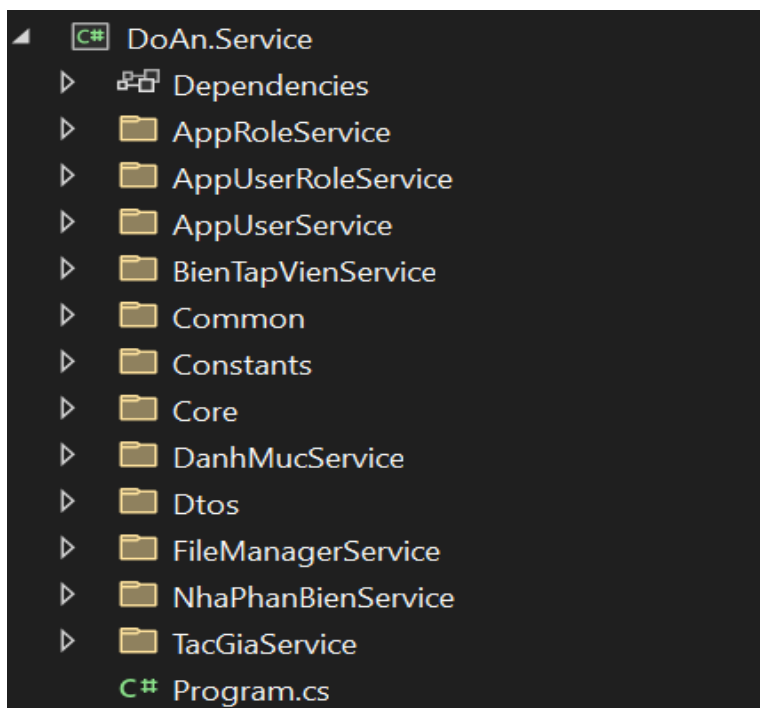
Hình 3.21: Cấu trúc code backend



Hình 3.22: Cấu trúc lớp domain



Hình 3.23: Cấu trúc lớp repository



Hình 3.24: Cấu trúc lớp Service

## 4.2 Cách triển khai lập trình

### 4.2.1 Cách tạo bảng trong cơ sở dữ liệu sử dụng phương pháp Code First

#### - Tạo 1 lớp Entity

Tạo các lớp mô hình đại diện cho các bảng trong cơ sở dữ liệu. Ví dụ một bảng NhaPhanbien

```

6      using System.Text;
7      using System.Threading.Tasks;
8
9      namespace DoAn.Domain.Entities
10     {
11         [Table("NhaphanBien")]
12         public class NhaPhanBien : Entity
13         {
14             public DateTime NgayBTVGiao { get; set; }
15             public DateTime NgayDenHanPhanHoi { get; set; }
16             public DateTime NgayDenHanPhanBien { get; set; }
17             public string? DeXuat { get; set; }
18             public string? IdBaiViet { get; set; }
19             public string? IdBienTapvien { get; set; }
20             public string? IdNhaPhanBien { get; set; }
21         }
22     }
23
24

```

Hình 3.25: Entity

#### - Tạo lớp ngữ cảnh (DbContext Class)

Tạo một lớp ngữ cảnh kế thừa từ DbContext, trong đó định nghĩa các DbSet đại diện cho các bảng

```

public DbSet<NhaPhanBien> NhaPhanBien { get; set; }

```

Hình 3.26: dbcontext

#### - Cấu hình chuỗi kết nối (Connection String)

Thêm chuỗi kết nối vào tệp appsettings.json

```

{
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "ConnStr": "Data Source=ADMIN-PC\\THANHDOC;Initial Catalog=DBTapChikhoaHoc;Persist Security Info=True;User ID=sa;Password=12;TrustServerCertificate=True"
  },
  "JWT": {

```

Hình 3.26: connection string

- Tạo và áp dụng Migration

Sử dụng các lệnh từ Package Manager Console để tạo và áp dụng migrations.

- Mở Package Manager Console từ Visual Studio
- Chạy lệnh sau để tạo migration đầu tiên:

```

Done.
PM> Add-Migration 'createnhphanbien'

```

Lệnh này sẽ tạo một lớp migration trong thư mục Migrations, trong đó có chứa các lệnh để tạo bảng SanPham

- Sau đó, áp dụng migration bằng cách chạy lệnh:

```

PM> update-database

```

Lệnh này sẽ áp dụng các thay đổi trong migration lên cơ sở dữ liệu của bạn, tạo ra bảng SanPham.

- Kiểm tra cơ sở dữ liệu

	Column Name	Data Type	Allow Nulls
🔑	Id	uniqueidentifier	<input type="checkbox"/>
	NgayBTVGiao	datetime2(7)	<input type="checkbox"/>
	NgayDenHanPhanHoi	datetime2(7)	<input type="checkbox"/>
	NgayDenHanPhanBien	datetime2(7)	<input type="checkbox"/>
▶	DeXuat	nvarchar(MAX)	<input checked="" type="checkbox"/>
	IdBaiViet	nvarchar(MAX)	<input checked="" type="checkbox"/>
	IdBienTapvien	nvarchar(MAX)	<input checked="" type="checkbox"/>
	IdNhaPhanBien	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Đã thành công trong việc tạo bảng

#### 4.2.2 Triển khai code backend

Triển khai code cho chức năng quản lý nhà phản biện với bảng NhaPhanBien vừa được tạo

- Tạo lớp Repository

Tạo giao diện repository để định nghĩa các phương thức tương tác với cơ sở dữ liệu:

```

1  using DoAn.Domain.Entities;
2  using DoAn.Repository.Core;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace DoAn.Repository.NhaPhanBienRepository
10 {
11     5 references
12     public interface INhaPhanBienRepository : IRepository<NhaPhanBien>
13     {
14     }
15 }

```

Triển khai giao diện repository để thực hiện các phương thức tương tác với cơ sở dữ liệu:

```

1  using DoAn.Domain;
2  using DoAn.Domain.Entities;
3  using DoAn.Repository.Core;
4  using System;
5  using System.Collections.Generic;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9
10 namespace DoAn.Repository.NhaPhanBienRepository
11 {
12     1 reference
13     internal class NhaPhanBienRepository : Repository<NhaPhanBien>, INhaPhanBienRepository
14     {
15         0 references
16         public NhaPhanBienRepository(DoAnContext context) : base(context) { }
17     }
18 }

```



## - Tạo lớp service

Tạo giao diện service để định nghĩa các phương thức xử lý nghiệp vụ:

```
namespace DoAn.Service.NhaPhanBienService
{
    3 references
    public interface INhaPhanBienService : IService<NhaPhanBien>
    {
        4 references
        ResponseWithDto<PagedList<NhaPhanBienDto>> GetDataByPage(NhaPhanBienSearchDto searchDto);
        1 reference
        ResponseWithDto<NhaPhanBien> AddNhaPhanBien(NhaPhanBienDto nhaPhanBien);
        1 reference
        ResponseWithMessageDto DeleteNhaPhanBien(Guid id);
        1 reference
        ResponseWithMessageDto UpdateNhaPhanBien(Guid id, NhaPhanBienDto nhaPhanBien);
        2 references
        ResponseWithDto<NhaPhanBienDto> FindById(Guid id);
        2 references
        ResponseWithDto<List<NhaPhanBienSelect>> NhaPhanBienSelect();
    }
}
```

Triển khai giao diện service để thực hiện các phương thức xử lý nghiệp vụ

```
namespace DoAn.Service.NhaPhanBienService
```

```
{
```

```
    public class NhaPhanBienService : Service<NhaPhanBien> , INhaPhanBienService
```

```
    {
```

```
        private readonly INhaPhanBienRepository _NhaPhanBienRepository;
```

```
        private readonly IAppUserRepository _appUserRepository;
```

```
        private readonly IAppUserRoleRepository _appUserRoleRepository;
```

```
        private readonly IAppRoleRepository _appRoleRepository;
```

```
        private readonly ITacGiaRepository _tacGiaRepository;
```

```
        private readonly IFileManagerRepository _fileManagerRepository;
```

```
        public
                                NhaPhanBienService(ITacGiaRepository
tacGiaRepository, INhaPhanBienRepository  nhaPhanBienRepository, IAppUserRepository
appUserRepository, IAppUserRoleRepository appUserRoleRepository, IAppRoleRepository
appRoleRepository, IFileManagerRepository fileManagerRepository) :
base(nhaPhanBienRepository)
```

```
    {
```

```
        _NhaPhanBienRepository = nhaPhanBienRepository;
```

```

        _appUserRepository = appUserRepository;
        _appUserRoleRepository = appUserRoleRepository;
        _appRoleRepository = appRoleRepository;
        _tacGiaRepository = tacGiaRepository;
        _fileManagerRepository = fileManagerRepository;
    }

    public ResponseWithDataDto<NhaPhanBien> AddNhaPhanBien(NhaPhanBienDto
nhaPhanBien)
    {
        try
        {
            var data = _NhaPhanBienRepository.Add(nhaPhanBien);
            return data != null ? new ResponseWithDataDto<NhaPhanBien>()
            {
                Data = data,
                Message = "Them thanh cong",
                Status = StatusConstant.SUCCESS,
            } : new ResponseWithDataDto<NhaPhanBien>()
            {
                Data = null,
                Message = "Thêm lỗi",
                Status = "Error"
            };
        }
        catch (Exception ex)
        {
            return new ResponseWithDataDto<NhaPhanBien>()
            {
                Data = null,
                Message = ex.Message,
                Status = StatusConstant.ERROR
            };
        }
    }

```

```

        };
    }
}

public ResponseWithMessageDto DeleteNhaPhanBien(Guid id)
{
    try
    {
        var nhaPhanBien = _NhaPhanBienRepository.GetById(id);
        if (nhaPhanBien == null)
        {
            return new ResponseWithMessageDto()
            {
                Message = "Không tồn tại",
                Status = StatusConstant.ERROR
            };
        }
        var data = _NhaPhanBienRepository.Delete(nhaPhanBien);
        return new ResponseWithMessageDto()
        {
            Message = "Xóa thành công",
            Status = StatusConstant.SUCCESS
        };
    }
    catch (Exception ex)
    {
        return new ResponseWithMessageDto()
        {
            Message = ex.Message,
            Status = StatusConstant.ERROR
        };
    }
}

```

```

    }

    public ResponseWithDataDto<NhaPhanBienDto> FindById(Guid id)
    {
        try
        {
            var data = from tblNhaPhanBien in
                _NhaPhanBienRepository.GetQueryable()
                join tbltacgia in _tacGiaRepository.GetQueryable()
                on tblNhaPhanBien.IdBaiViet equals
tbltacgia.Id.ToString()
                join tblFileManager in
                _fileManagerRepository.GetQueryable()
                on tbltacgia.Id equals tblFileManager.ParentId
                where tblNhaPhanBien.Id == id
                select new NhaPhanBienDto
                {
                    Id = tblNhaPhanBien.Id,
                    NgayBTVGiao = tblNhaPhanBien.NgayBTVGiao,
                    NgayDenHanPhanBien =
tblNhaPhanBien.NgayDenHanPhanBien,
                    NgayDenHanPhanHoi = tblNhaPhanBien.NgayDenHanPhanHoi,
                    DeXuat = tblNhaPhanBien.DeXuat,
                    Tieude = tbltacgia.Tieude,
                    TomTat = tbltacgia.TomTat,
                    pathBaiViet = tblFileManager.Path,

                };

            return new ResponseWithDataDto<NhaPhanBienDto>()
            {
                Data = data.FirstOrDefault(),
                Status = StatusConstant.SUCCESS,
                Message = "Lấy thành công"
            }
        }
    }

```

```

        };
    }
    catch (Exception ex)
    {
        return new ResponseWithDataDto<NhaPhanBienDto>()
        {
            Data = null,
            Status = StatusConstant.ERROR,
            Message = ex.Message
        };
    }
}

public ResponseWithDataDto<PagedList<NhaPhanBienDto>>
GetDataByPage(NhaPhanBienSearchDto searchDto)
{
    try
    {
        var query = from tblNhaPhanBien in
            _NhaPhanBienRepository.GetQueryable()
                join tbltacgia in _tacGiaRepository.GetQueryable()
                on tblNhaPhanBien.IdBaiViet equals
tbltacgia.Id.ToString()
                join tblFileManager in
            _fileManagerRepository.GetQueryable()
                on tbltacgia.Id equals tblFileManager.ParentId
            select new NhaPhanBienDto
            {
                Id = tblNhaPhanBien.Id,
                NgayBTVGiao = tblNhaPhanBien.NgayBTVGiao,
                NgayDenHanPhanBien =
tblNhaPhanBien.NgayDenHanPhanBien,

```

```

        NgayDenHanPhanHoi
tblNhaPhanBien.NgayDenHanPhanHoi,

        DeXuat = tblNhaPhanBien.DeXuat,
        IdBaiViet = tblNhaPhanBien.IdBaiViet,
        IdNhaPhanBien = tblNhaPhanBien.IdNhaPhanBien,
        IdBienTapvien = tblNhaPhanBien.IdBienTapvien,
        Tieude = tbltacgia.Tieude,
        TomTat = tbltacgia.TomTat,
        pathBaiViet = tblFileManager.Path,
        TrangThai = tbltacgia.TrangThai,

};

if (searchDto != null)
{
    if (searchDto.DexuatFilter != null)
    {
        query = query.Where(record =>
record.DeXuat.Trim().ToLower().Contains(searchDto.DexuatFilter.Trim().ToLower()));
    }
}

var result = PagedList<NhaPhanBienDto>.Create(query, searchDto);
return new ResponseWithDataDto<PagedList<NhaPhanBienDto>>>()
{
    Data = result,
    Status = StatusConstant.SUCCESS,
    Message = "Lấy thành công"
};
}

catch (Exception ex)
{

```

```

return new ResponseWithDataDto<PagedList<NhaPhanBienDto>>>()
{
    Data = null,
    Status = StatusConstant.ERROR,
    Message = ex.Message
};
}
}

public ResponseWithDataDto<List<NhaPhanBienSelect>> NhaPhanBienSelect()
{
    try
    {
        var query = from AppUsertbl in _appUserRepository.GetQueryable()
                    join AppUserRoletbl in _appUserRoleRepository.GetQueryable()
                    on AppUsertbl.Id equals AppUserRoletbl.UserId into
userRole

                    from userRoleGroup in userRole.DefaultIfEmpty()
                    join AppRoletbl in _appRoleRepository.GetQueryable()
                    on userRoleGroup.RoleId equals AppRoletbl.Id into role
                    from roleGroup in role.DefaultIfEmpty()

        select new NhaPhanBienSelect
        {
            Id = AppUsertbl.Id,
            FullName = AppUsertbl.FullName,
            Username = AppUsertbl.UserName,
            Email = AppUsertbl.Email,
            PhoneNumber = AppUsertbl.PhoneNumber,
            RoleCode = roleGroup.RoleCode,

```

```

        RoleName = roleGroup.Name

        };

        return new ResponseWithDataDto<List<NhaPhanBienSelect>>()
        {
            Data = query.Where(x=>x.RoleCode== "NHAPHANBIEN").ToList(),
            Status = StatusConstant.SUCCESS,
            Message = "Lấy thành công"
        };
    }
    catch (Exception ex)
    {
        return new ResponseWithDataDto<List<NhaPhanBienSelect>>()
        {
            Data = null,
            Status = StatusConstant.ERROR,
            Message = ex.Message
        };
    }
}

public ResponseWithMessageDto UpdateNhaPhanBien(Guid id, NhaPhanBienDto
nhaPhanBien)
{
    try
    {
        var NhaPhanBienNew = _NhaPhanBienRepository.GetById(id);
        if (NhaPhanBienNew == null)
        {
            return new ResponseWithMessageDto()
            {
                Message = "Không tìm thấy Tac Gia",
            }
        }
    }
    catch (Exception ex)
    {
        return new ResponseWithMessageDto()
        {
            Message = ex.Message
        };
    }
}

```



```

        Status = StatusConstant.ERROR,
    };
}
else
{
    NhaPhanBienNew.NgayBTVGiao = nhaPhanBien.NgayBTVGiao;
    NhaPhanBienNew.NgayDenHanPhanHoi =
nhaPhanBien.NgayDenHanPhanHoi;
    NhaPhanBienNew.NgayDenHanPhanBien =
nhaPhanBien.NgayDenHanPhanBien;
    NhaPhanBienNew.DeXuat = nhaPhanBien.DeXuat;
    NhaPhanBienNew.IdNhaPhanBien = nhaPhanBien.IdNhaPhanBien;

    return new ResponseWithMessageDto()
    {
        Message = "Cập nhật TacGia thành công",
        Status = StatusConstant.SUCCESS,
    };
}

}
catch (Exception ex)
{
    return new ResponseWithMessageDto()
    {
        Message = ex.Message,
        Status = "Error"
    };
}
}
}
}

```

### - Tạo lớp controller

Tạo controller để xử lý các yêu cầu HTTP và gọi các phương thức từ service

```
namespace DoAnBackEnd.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class NhaPhanBienController : ControllerBase
    {
        private readonly INhaPhanBienService _NhaPhanBienService;
        private readonly ITacGiaService _tacGiaService;

        public NhaPhanBienController(INhaPhanBienService nhaPhanBienService,
        ITacGiaService tacGiaService)
        {
            _NhaPhanBienService = nhaPhanBienService;
            _tacGiaService = tacGiaService;
        }

        [HttpGet]
        public async Task<IActionResult> GetDataByPage([FromQuery]
        NhaPhanBienSearchDto search)
        {
            try
            {
                var objId =
                HttpContext.User.FindFirst(ClaimTypes.NameIdentifier)?.Value;

                var npb = _NhaPhanBienService.GetDataByPage(search);
                if (npb.Data.Items != null)
                {
                    string jsonData = JsonConvert.SerializeObject(npb.Data.Items);
                    /*HttpContext.Session.SetString("serachData"
                    +
                    typeof(SanPhamDto).Name, jsonData);*/
                }
            }
        }
    }
}
```

```

    }

    return StatusCode(StatusCodes.Status200OK, npb);
}

catch (Exception ex)
{
    return    StatusCode(StatusCodes.Status500InternalServerError,    new
ResponseWithMessageDto { Status = StatusConstant.ERROR, Message = ex.Message });
}

}

[HttpGet("{id}")]
public async Task<IActionResult> FindById(Guid id)
{
    try
    {
        var rs = _NhaPhanBienService.FindById(id);

        return    rs.Status    ==    StatusConstant.SUCCESS    ?
StatusCode(StatusCodes.Status200OK,    rs)    :
StatusCode(StatusCodes.Status500InternalServerError, rs);
    }

    catch (Exception ex)
    {
        return    StatusCode(StatusCodes.Status500InternalServerError,    new
ResponseWithMessageDto
        {
            Status = StatusConstant.ERROR,
            Message = ex.Message
        });
    }
}

[HttpGet("Nhaphanbien")]

```

```

        public async Task<IActionResult> GetDataByPageByNhaphanbien([FromQuery]
        NhaPhanBienSearchDto search)
        {
            try
            {
                var objId =
                HttpContext.User.FindFirst(ClaimTypes.NameIdentifier)?.Value;

                var npb =
                _NhaPhanBienService.GetDataByPage(search).Data.Items.Where(x => x.IdNhaPhanBien ==
                objId).ToList();

                if (npb != null)
                {
                    string jsonData = JsonConvert.SerializeObject(npb);
                    /*HttpContext.Session.SetString("serachData"
                    typeof(SanPhamDto).Name, jsonData);*/
                }

                return StatusCode(StatusCode.Status200OK, npb);
            }
            catch (Exception ex)
            {
                return StatusCode(StatusCode.Status500InternalServerError, new
                ResponseWithMessageDto { Status = StatusConstant.ERROR, Message = ex.Message });
            }
        }

        [HttpGet("Nhaphanbien/{id}")]
        public async Task<IActionResult> GetDataByPageByNhaphanbienById(Guid
        id, [FromQuery] NhaPhanBienSearchDto search)
        {
            try
            {
                var objId =
                HttpContext.User.FindFirst(ClaimTypes.NameIdentifier)?.Value;

```

```

        var npb =
        _NhaPhanBienService.GetDataByPage(search).Data.Items.Where(x => x.IdNhaPhanBien ==
        objId && x.Id==id).ToList();

        if (npb != null)
        {
            string jsonData = JsonConvert.SerializeObject(npb);
            /*HttpContext.Session.SetString("serachData"
            +
            typeof(SanPhamDto).Name, jsonData);*/

        }

        return StatusCode(StatusCodes.Status200OK, npb);
    }

    catch (Exception ex)
    {
        return StatusCode(StatusCodes.Status500InternalServerError, new
        ResponseWithMessageDto { Status = StatusConstant.ERROR, Message = ex.Message });
    }
}

[HttpPost("Create")]
public async Task<IActionResult> Create([FromBody] NhaPhanBienVM nhaPhanBien)
{
    try
    {
        var tacgia =
        _tacGiaService.FindById(Guid.Parse(nhaPhanBien.IdBaiViet));

        var objId =
        HttpContext.User.FindFirst(ClaimTypes.NameIdentifier)?.Value;

        var d = new NhaPhanBien()
        {

            NgayBTVGiao = nhaPhanBien.NgayBTVGiao,
            NgayDenHanPhanBien = nhaPhanBien.NgayDenHanPhanBien,
            NgayDenHanPhanHoi = nhaPhanBien.NgayDenHanPhanHoi,

```

```

        DeXuat = nhaPhanBien.DeXuat,
        IdBienTapvien=objId,
        IdBaiViet=nhaPhanBien.IdBaiViet,
        IdNhaPhanBien=nhaPhanBien.IdNhaPhanBien

    };

    await _NhaPhanBienService.Create(d);
    var tacgiaDto = new TacGiaDto()
    {
        IdTacGia = tacgia.Data.IdTacGia,
        ChuyenMuc = tacgia.Data.ChuyenMuc,
        Id = tacgia.Data.Id,
        LoiNhanChoBanBientap = tacgia.Data.LoiNhanChoBanBientap,
        NgonNgu = tacgia.Data.NgonNgu,
        SDT = tacgia.Data.SDT,
        pathBaiViet = tacgia.Data.pathBaiViet,
        TaiLieuThamKhao= tacgia.Data.TaiLieuThamKhao,
        Thongtinbosungtukhoa = tacgia.Data.Thongtinbosungtukhoa,
        ThuocTapChi = tacgia.Data.ThuocTapChi,
        Tieude = tacgia.Data.Tieude,
        TomTat = tacgia.Data.TomTat,
        TrangThai = "Đã gửi cho phản biện",
    };

    await _tacGiaService.Update(tacgiaDto);

    return StatusCode(StatusCodes.Status201Created, new
    ResponseWithDataDto<NhaPhanBien>()
    {
        Status = StatusConstant.SUCCESS,
        Message = "Thêm thành công",
        Data = d
    });
}

```

```

        catch (Exception ex)
        {
            return StatusCode(StatusCode.Status500InternalServerError, new
ResponseWithMessageDto { Status = StatusConstant.ERROR, Message = ex.Message });
        }
    }

    [HttpPut("edit/{id}")]
    public async Task<IActionResult> Edit(Guid id, [FromBody] NhaPhanBienVM
action)
    {
        if (ModelState.IsValid)
        {
            try
            {
                var tacgia =
_tacGiaService.FindById(Guid.Parse(action.IdBaiViet));

                var objId =
HttpContext.User.FindFirst(ClaimTypes.NameIdentifier)?.Value;

                var rs = _NhaPhanBienService.FindBy(x =>
x.Id.Equals(id)).FirstOrDefault();
                if (rs != null)
                {
                    rs.NgayBTVGiao = action.NgayBTVGiao;
                    rs.NgayDenHanPhanHoi = action.NgayDenHanPhanHoi;
                    rs.NgayDenHanPhanBien = action.NgayDenHanPhanBien;
                    rs.DeXuat = action.DeXuat;
                    rs.IdNhaPhanBien = rs.IdNhaPhanBien;
                    rs.IdBienTapvien = rs.IdBienTapvien;
                    rs.IdBaiViet = rs.IdBaiViet;
                    await _NhaPhanBienService.Update(rs);
                }
            }
        }
    }

```

```

//Edit baiviet
var tacgiaDto = new TacGiaDto();
tacgiaDto.IdTacGia = tacgia.Data.IdTacGia;
tacgiaDto.ChuyenMuc = tacgia.Data.ChuyenMuc;
tacgiaDto.Id = tacgia.Data.Id;
tacgiaDto.Loinhanchobanbientap =
tacgia.Data.Loinhanchobanbientap;
tacgiaDto.NgonNgu = tacgia.Data.NgonNgu;
tacgiaDto.SDT = tacgia.Data.SDT;
tacgiaDto.pathBaiViet = tacgia.Data.pathBaiViet;
tacgiaDto.TaiLieuThamKhao = tacgia.Data.TaiLieuThamKhao;
tacgiaDto.Thongtinbosungtukhoa =
tacgia.Data.Thongtinbosungtukhoa;
tacgiaDto.ThuocTapChi = tacgia.Data.ThuocTapChi;
tacgiaDto.Tieude = tacgia.Data.Tieude;
tacgiaDto.TomTat = tacgia.Data.TomTat;
tacgiaDto.TrangThai = "Đã phản biện";

await _tacGiaService.Update(tacgiaDto);
return StatusCode(StatusCodes.Status200OK, new
ResponseWithDataDto<NhaPhanBien>
{
    Status = StatusConstant.SUCCESS,
    Message = "Cập nhật thành công",
    Data = rs
});
}
else
{
    return StatusCode(StatusCodes.Status204NoContent, new
ResponseWithMessageDto

```



```

        {
            Status = StatusConstant.ERROR,
            Message = "Không tìm thấy NXB"
        });
    }
}
catch (Exception ex)
{
    return StatusCode(StatusCodes.Status500InternalServerError, new
ResponseWithMessageDto
    {
        Status = StatusConstant.ERROR,
        Message = ex.Message
    });
}
else
{
    return StatusCode(StatusCodes.Status500InternalServerError, new
ResponseWithMessageDto
    {
        Status = StatusConstant.ERROR,
        Message = "Yêu cầu nhập đầy đủ thông tin"
    });
}
}
[HttpDelete("delete/{id}")]
public async Task<IActionResult> Delete(Guid id)
{
    try
    {

```

```

        var      nhaPhanBien      =      _NhaPhanBienService.FindBy(x =>
x.Id.Equals(id)).FirstOrDefault();

        if (nhaPhanBien == null)
        {
            return      StatusCode(StatusCodes.Status404NotFound,      new
ResponseWithMessageDto
            {
                Message = "Không tìm thấy nha phan bien",
                Status = StatusConstant.ERROR
            });
        }
        else
        {
            await _NhaPhanBienService.Delete(nhaPhanBien);
            return      StatusCode(StatusCodes.Status200OK,      new
ResponseWithMessageDto
            {
                Message = "Xóa thành công nha phan bien",
                Status = StatusConstant.SUCCESS
            });
        }
    }
    catch (Exception ex)
    {
        return      StatusCode(StatusCodes.Status500InternalServerError,      new
ResponseWithMessageDto
        {
            Status = StatusConstant.ERROR,
            Message = ex.Message
        });
    }
}

[HttpGet("NPBSelect")]

```

```

public IActionResult NPBSelect()
{
    try
    {
        var objId =
HttpContext.User.FindFirst(ClaimTypes.NameIdentifier)?.Value;

        var npb = _NhaPhanBienService.NhaPhanBienSelect();
        if (npb.Data != null)
        {
            string jsonData = JsonConvert.SerializeObject(npb.Data);
            /*HttpContext.Session.SetString("serachData"
typeof(SanPhamDto).Name, jsonData);*/
        }
        return StatusCode(StatusCode.Status200OK, npb);
    }
    catch (Exception ex)
    {
        return StatusCode(StatusCode.Status500InternalServerError, new
ResponseWithMessageDto { Status = StatusConstant.ERROR, Message = ex.Message });
    }
}
}

```

Với các bước trên, em đã cấu hình và triển khai một ứng dụng ASP.NET Core theo mô hình Domain Repository Service Controller. Mô hình này giúp tách biệt rõ ràng các tầng trong ứng dụng, tuân thủ nguyên tắc SOLID và giúp mã nguồn dễ dàng bảo trì và mở rộng.

### 4.2.3 Triển khai code Frontend

- Lấy danh sách các bài cần phản biện từ API

```

    },
    async loadData(pageIndex, pageSize, params) {
        var searchParam = {
            pageIndex: pageIndex,
            pageSize: pageSize,
        };
        if (params) {
            searchParam = params;
        }

        var urlQuery = new URLSearchParams(searchParam).toString();
        var result = await APIService.get("/NhaPhanBien/Nhaphanbien?" + urlQuery);
        console.log('result', result.data);
        if (result.data != null) {
            var res = result.data;
            this.lstNXB = result.data;
            this.optionPage.pageIndex = res.pageIndex;
            this.optionPage.pageSize = res.pageSize;
            this.optionPage.totalCount = res.totalCount;
        }
    },
},

```

- Thêm bài phản biện từ API

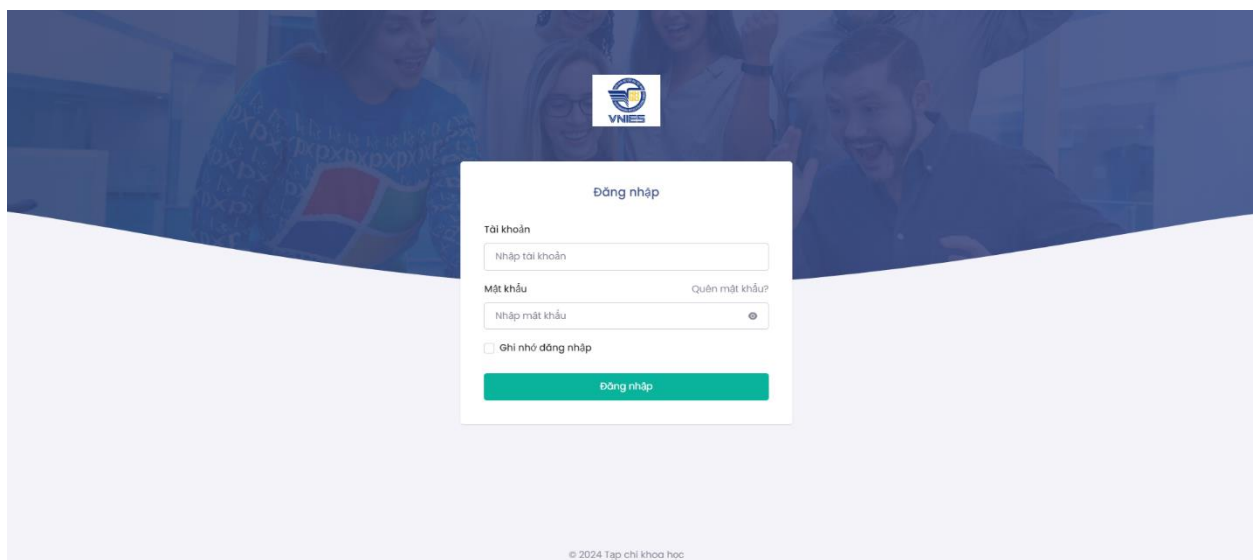
```

    },
    openModalAdd(){
      Swal.fire({
        text: "Bạn có chắc chắn muốn gửi bài lên",
        icon: "warning",
        showCancelButton: true,
        cancelButtonText: "Hủy",
        cancelButtonColor: "#f46a6a",
        confirmButtonColor: "#34c38f",
        confirmButtonText: "chắc chắn",
      }).then(async (result) => {
        if (result.value) {
          const pbDeXuat = {
            deXuat : this.phanbien.DeXuat
          }
          ApiService.put(`/NhaPhanBien/edit/${this.id}`, pbDeXuat)
            .then(response =>{
              console.log(response.data)
              const TrangThai = {
                trangThai : 'đã phản biện'
              }
              console.log('thành công', response)
              this.$refs.childRef.handleUpload(this.id);
            })
            .catch(error =>{
              console.log(error)
            })
            this.closeModal();
        }
      })
    }
  }
}

```

## 4.3 Triển khai phần mềm(giao diện màn hình)

### 4.3.1 Giao diện đăng nhập



- Chức năng: tác giả, nhà phản biện, biên tập viên đăng nhập vào

### 4.3.2 Tác giả thêm mới bài viết

**VELZON**

Quản lý Bài viết của tác giả

Thêm mới bài viết

**BẮT ĐẦU**

tác giả > Bắt đầu

Anna Adame  
Founder

Nguồn ngữ gốc:

Chuyên mục:

**Yêu cầu nộp bài**

Bạn phải đọc và xác nhận rằng bạn đã hoàn thành các yêu cầu bên dưới trước

- ☐ Sự phù hợp với mục đích - phạm vi của Tạp chí.
- ☐ Sự tuân thủ các chính sách gửi bài, phản biện, biên tập của Tạp chí
- ☐ Sự tuân thủ các chính sách gửi bài, phản biện, biên tập của Tạp chí
- ☐ Tác giả sử dụng mẫu bản thảo để chuẩn bị bài viết.

Lời nhắn cho ban biên tập

**B I U S** **H<sub>1</sub> H<sub>2</sub>** **X<sub>1</sub> X<sub>2</sub>** **Normal** **Sans Serif**

Nhập nội dung văn bản...

☐ Sự tuân thủ các chính sách gửi bài, phản biện, biên tập của Tạp chí

☐ Tác giả sử dụng mẫu bản thảo để chuẩn bị bài viết.

Lời nhắn cho ban biên tập

**B I U S** **H<sub>1</sub> H<sub>2</sub>** **X<sub>1</sub> X<sub>2</sub>** **Normal** **Sans Serif**

Nhập nội dung văn bản...

☐ Đồng ý thu thập và lưu trữ dữ liệu cá nhân theo cam kết bảo mật

**Tiếp tục** **Cancel**

2024 © Velzon.

Design & Develop by Themestbrand

**VELZON**

Quản lý Bài viết của tác giả

Thêm mới bài viết

**NHẬP DỮ LIỆU MÔ TẢ**

tác giả > Nhập dữ liệu mô tả

Tiêu đề:

Tóm tắt:

B I U S H<sub>1</sub> H<sub>2</sub> X<sub>2</sub> X<sup>2</sup> Normal Normal Sans Serif

Nhập nội dung văn bản...

Số điện thoại:

**VELZON**

Quản lý Bài viết của tác giả

Thêm mới bài viết

Tải bài viết:

Tải File PDF

Cancel OK

Tài liệu tham khảo:

B I U S H<sub>1</sub> H<sub>2</sub> X<sub>2</sub> X<sup>2</sup> Sans Serif

Nhập nội dung văn bản...

Tiếp tục

2024 © Velzon. Design & Develop by Themebrand

➤ Chức năng: Sau khi đăng nhập với tài khoản tác giả thì sẽ chuyển sang trang thêm mới bài viết tác giả sẽ nhập thông tin và ấn tiếp tục đến khi hiển thị ra form để tải file pdf bài viết của tác giả lên

### 4.3.3 Quản lý bài viết của tác giả

**QUẢN LÝ SỐ TẠP CHÍ CHƯA XUẤT BẢN**

Danh sách số tạp chí

Q. Tiêu đề 🔍 Tìm kiếm

STT	Tiêu đề	Chuyên mục	Ngôn ngữ	Trạng thái	Thao tác
1	bài viết này đẹp lắm	string	string	Chưa gửi cho phản biện	<a href="#">✎</a> <a href="#">🗑</a>
2				Chưa gửi cho phản biện	<a href="#">✎</a> <a href="#">🗑</a>
3	linh ngu	Y tế	Tiếng anh	Chưa gửi cho phản biện	<a href="#">✎</a> <a href="#">🗑</a>
4	asdasdsad	Y tế	Tiếng anh	Chưa gửi cho phản biện	<a href="#">✎</a> <a href="#">🗑</a>
5	hi	Y tế	Tiếng anh	Chưa gửi cho phản biện	<a href="#">✎</a> <a href="#">🗑</a>

2024 © Velzon. Design & Develop by Themestbrand

➤ Chức năng: sau khi đăng nhập vào tài khoản tác giả và click vào danh sách tạp chí đã được gửi lên sẽ hiện ra danh sách tạp chí đã gửi lên ở đây tác giả có thể xem được bài của mình có bị trả về hay không

### 4.3.4 Quản lý số tạp chí chưa xuất bản

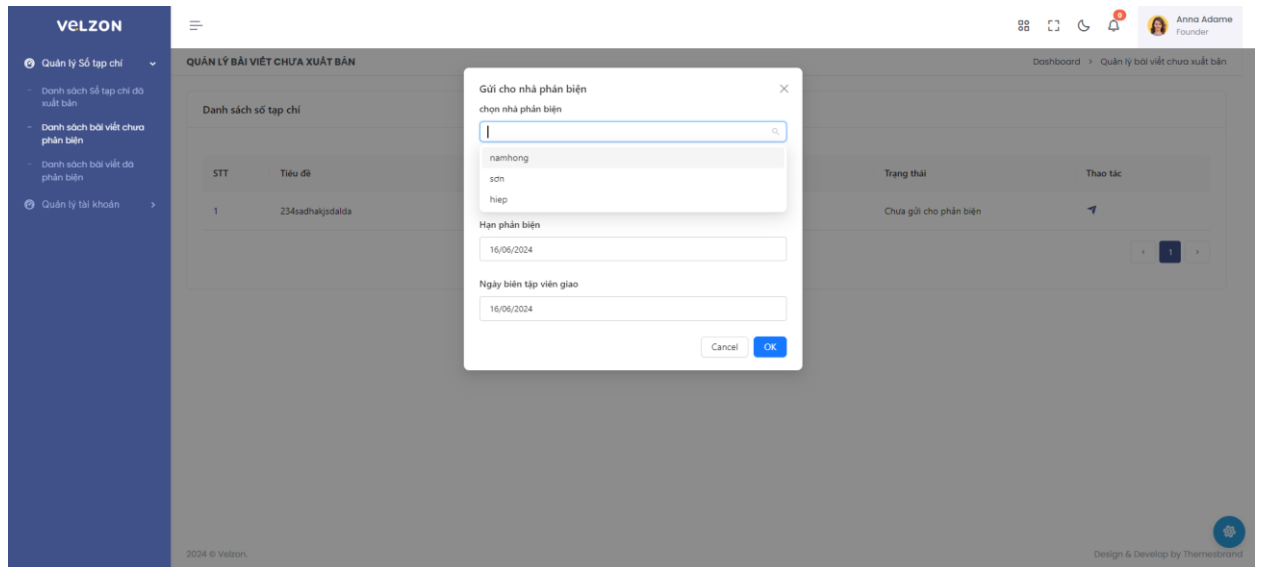
**QUẢN LÝ BÀI VIẾT CHƯA XUẤT BẢN**

Danh sách số tạp chí

STT	Tiêu đề	Chuyên mục	Ngôn ngữ	Trạng thái	Thao tác
1	Điểm lại những mốc son lịch sử trong chặng đường 65 năm thành lập Bộ Khoa học và Công nghệ.	CNTT	Tiếng việt	Chưa gửi cho phản biện	<a href="#">🚩</a>
2	Nguyễn Bộ trưởng Hoàng Văn Phong: Tin tưởng vào sức mạnh trí tuệ và khát vọng cống hiến của đội ngũ các nhà khoa học và công nghệ Việt Nam	CNTT	Tiếng việt	Chưa gửi cho phản biện	<a href="#">🚩</a>
3	Khoa học, công nghệ và đổi mới sáng tạo trước những cơ hội và thử thách	CNTT	Tiếng việt	Chưa gửi cho phản biện	<a href="#">🚩</a>
4	Xây dựng đội ngũ trí thức vững mạnh là bài đáp "nguyên khí quốc gia"	CHAOMUNG	Tiếng việt	Chưa gửi cho phản biện	<a href="#">🚩</a>
5	234asdhajksalda	CHAOMUNG	Tiếng việt	Chưa gửi cho phản biện	<a href="#">🚩</a>

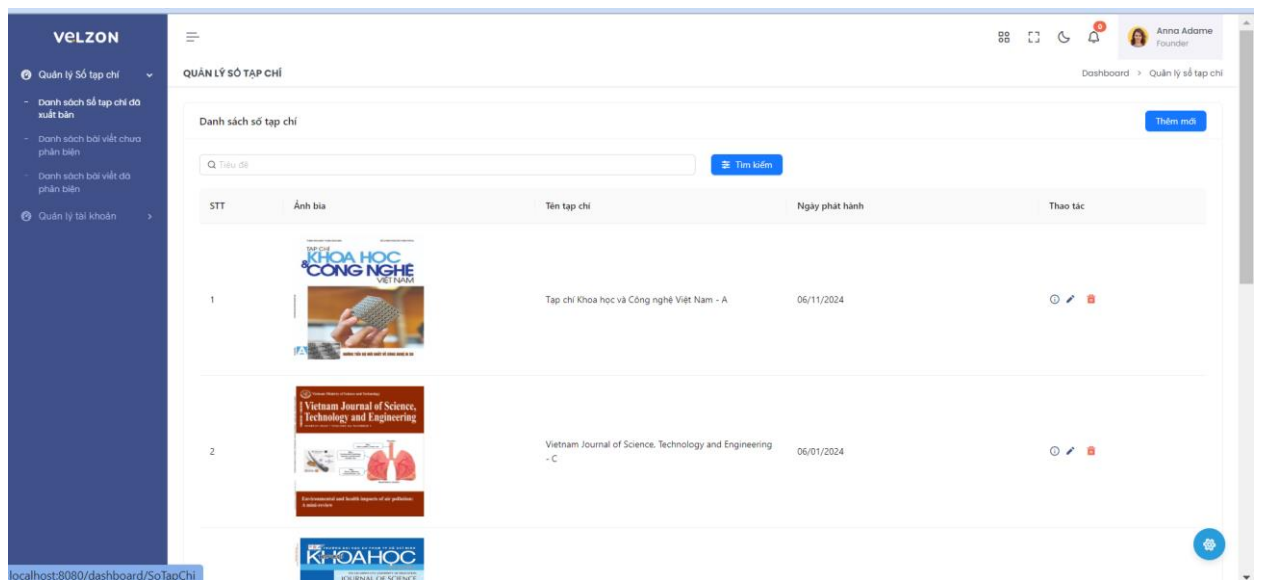
2024 © Velzon. Design & Develop by Themestbrand

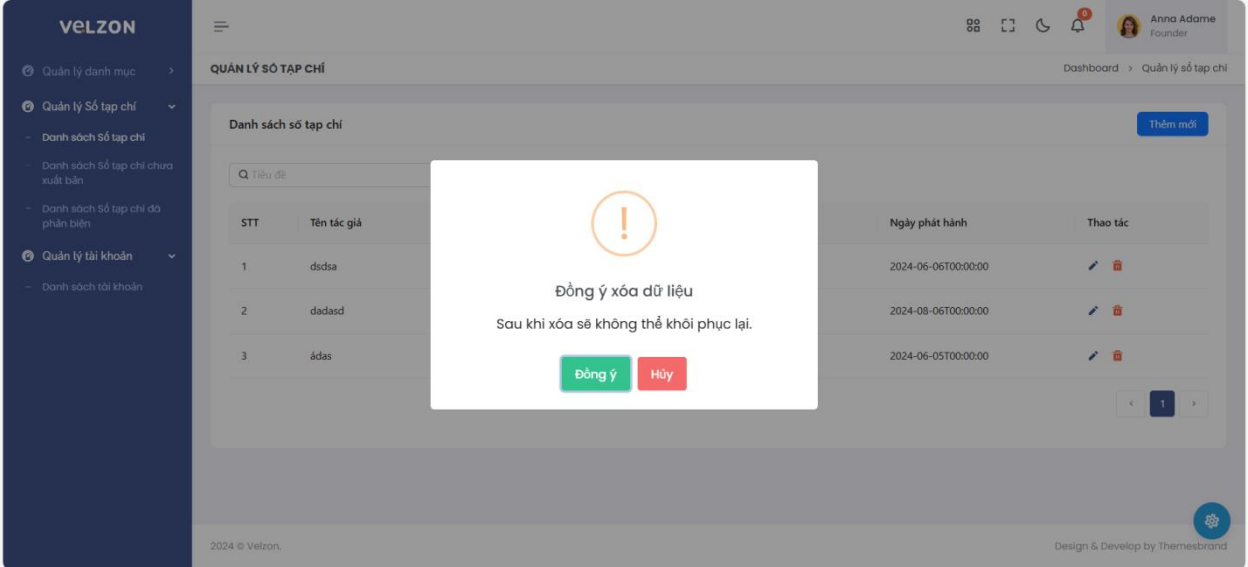
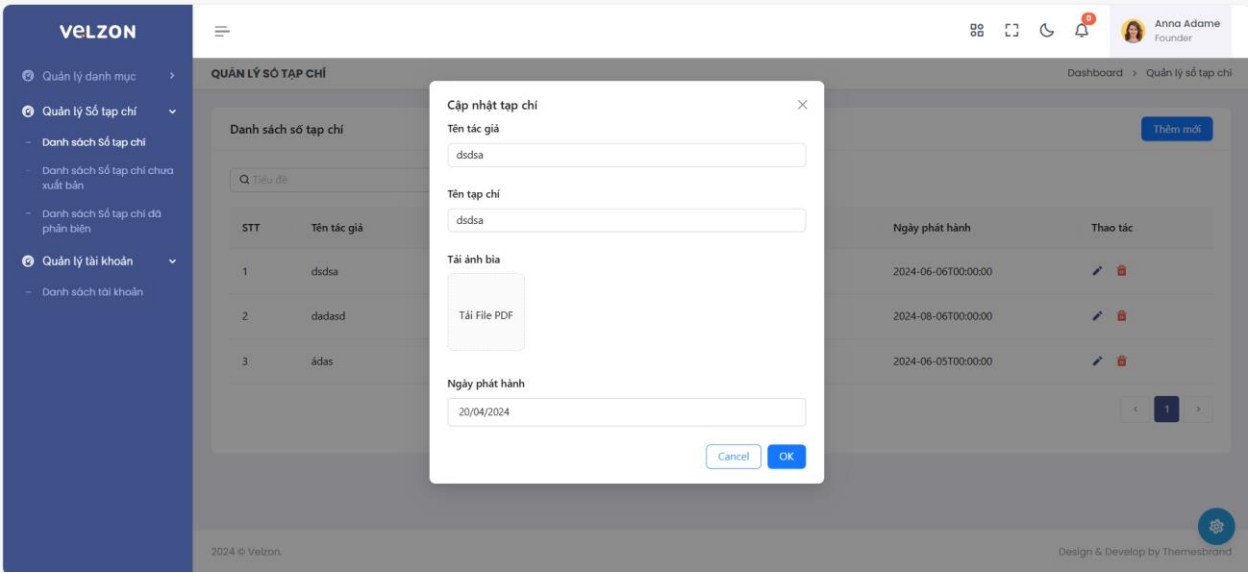
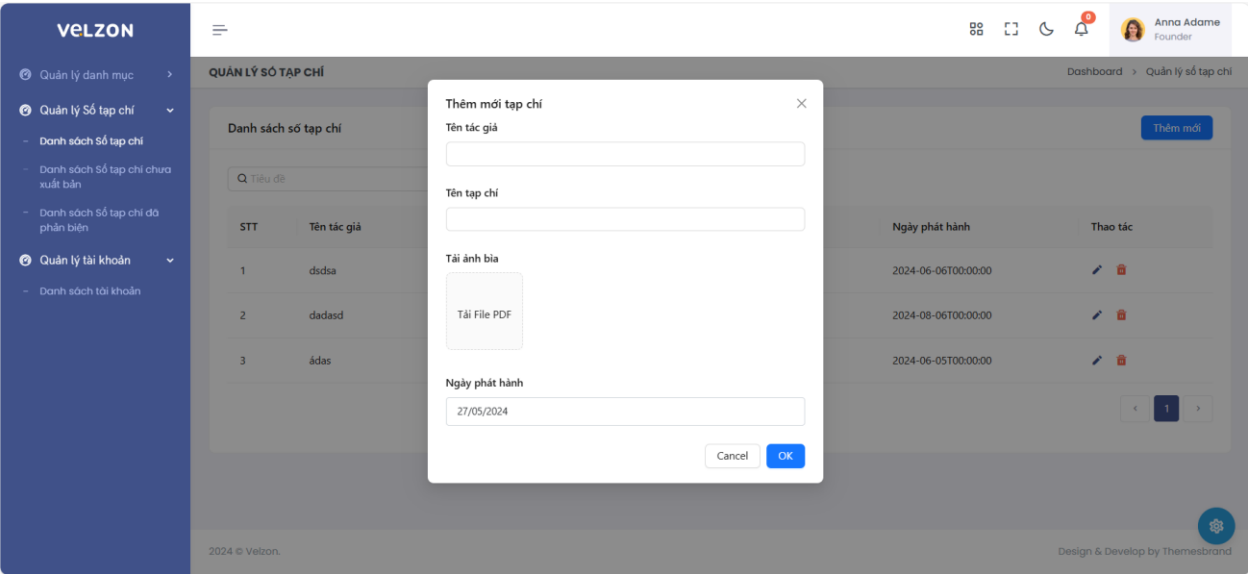




➤ Chức năng: biên tập viên đăng nhập vào tài khoản biên tập viên sau đó ấn vào danh sách số tạp chí chưa xuất bản hệ thống sẽ trả về 1 danh sách sau đó biên tập viên có thể ấn nút biểu tượng bút chì để hiển thị lên danh sách phản biện và biên tập viên có thể chọn và gửi đến nhà phản biện đã chọn

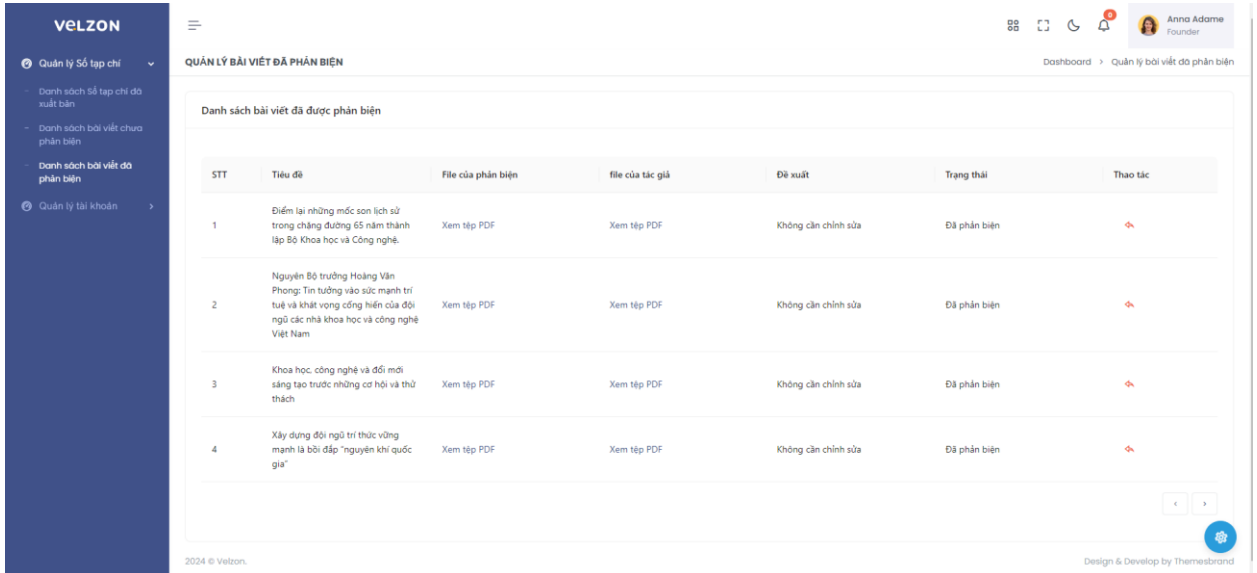
#### 4.3.5 Quản lý số tạp chí đã xuất bản













➤ Chức năng: biên tập viên sẽ đăng nhập vào tài khoản biên tập viên sau đó click vào danh sách số tạp chí sẽ hiển thị ra danh sách tạp chí đã xuất bản ở đây tác giả có thể thêm sửa xóa, xem chi tiết bài tạp chí đã đăng

#### 4.3.5 Quản lý số tạp chí đã phản biện



**QUẢN LÝ BÀI VIẾT ĐÃ PHẢN BIỆN**

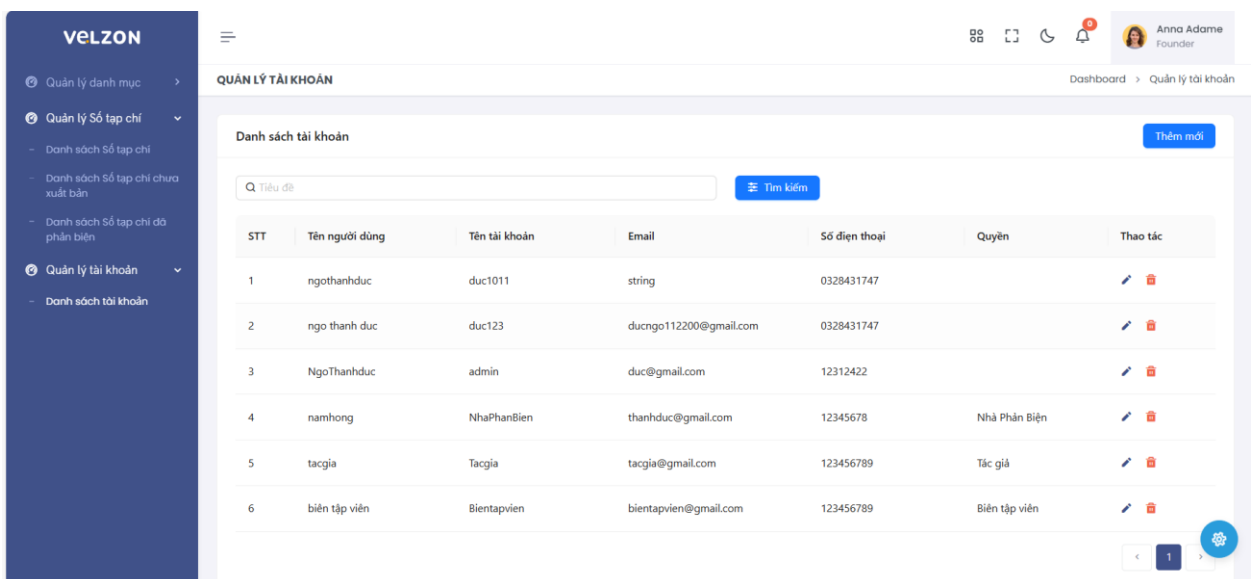
Danh sách bài viết đã được phản biện

STT	Tiêu đề	File của phản biện	file của tác giả	Đề xuất	Trạng thái	Thao tác
1	Điểm tại những mốc son lịch sử trong chặng đường 65 năm thành lập Bộ Khoa học và Công nghệ.	Xem tệp PDF	Xem tệp PDF	Không cần chỉnh sửa	Đã phản biện	 
2	Nguyễn Bộ trưởng Hoàng Văn Phóng: Tin tưởng vào sức mạnh trí tuệ và khát vọng cống hiến của đội ngũ các nhà khoa học và công nghệ Việt Nam	Xem tệp PDF	Xem tệp PDF	Không cần chỉnh sửa	Đã phản biện	 
3	Khoa học công nghệ và đổi mới sáng tạo trước những cơ hội và thử thách	Xem tệp PDF	Xem tệp PDF	Không cần chỉnh sửa	Đã phản biện	 
4	Xây dựng đội ngũ tri thức vững mạnh là bởi đắp "nguyên khí quốc gia"	Xem tệp PDF	Xem tệp PDF	Không cần chỉnh sửa	Đã phản biện	 

2024 © Velzon. Design & Develop by Themescrand

➤ Chức năng: sau khi đăng nhập vào tài khoản biên tập viên và click vào danh sách bài tạp chí đã phản biện sẽ hiển thị ra danh sách số tạp chí đã phản biện ở đây biên tập viên sẽ xem xét trả về cho tác giả hoặc đăng lên trang tạp chí

#### 4.3.6 Quản lý tài khoản















**QUẢN LÝ TÀI KHOẢN**

Danh sách tài khoản

Thêm mới

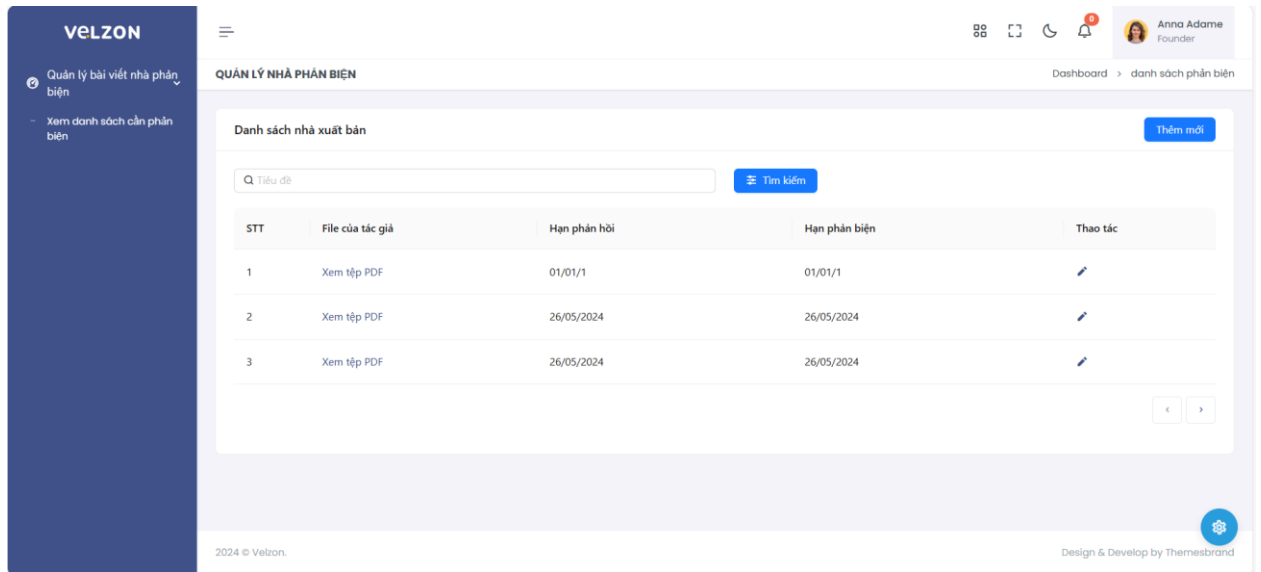
Q Tiêu đề Tìm kiếm

STT	Tên người dùng	Tên tài khoản	Email	Số điện thoại	Quyền	Thao tác
1	ngothanhduc	duc1011	string	0328431747		 
2	ngo thanh duc	duc123	ducngo112200@gmail.com	0328431747		 
3	NgoThanhduc	admin	duc@gmail.com	12312422		 
4	namhong	NhaPhanBien	thanhduc@gmail.com	12345678	Nhà Phản Biện	 
5	tacgia	Tacgia	tacgia@gmail.com	123456789	Tác giả	 
6	biên tập viên	Bientapvien	bientapvien@gmail.com	123456789	Biên tập viên	 

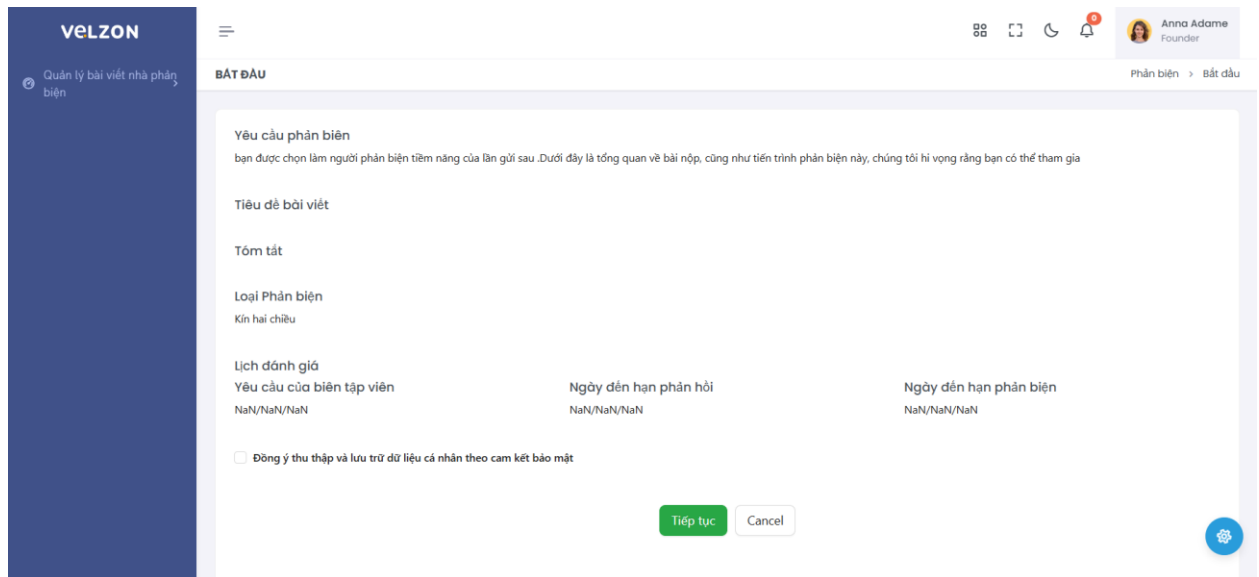
1

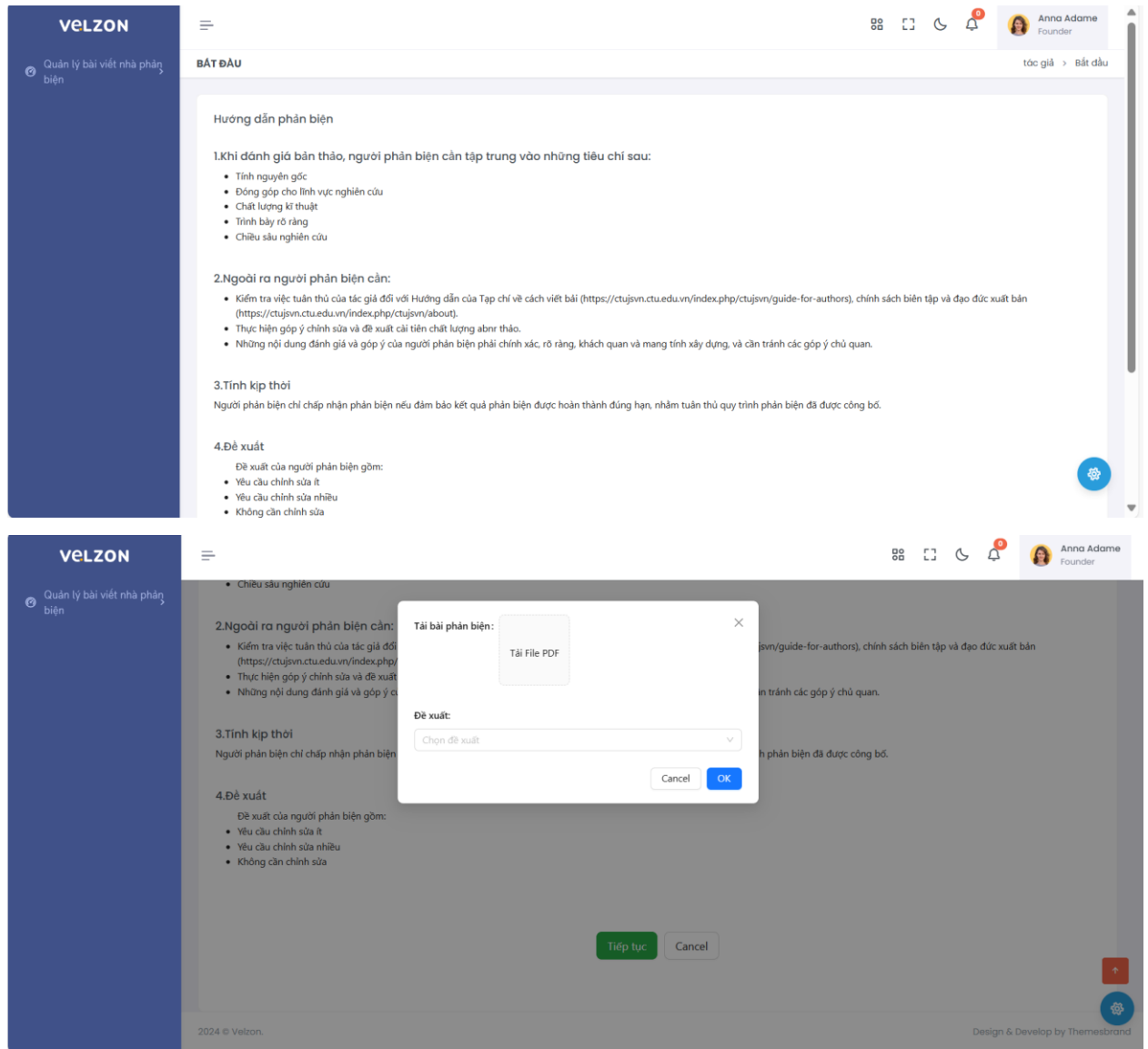
➤ Chức năng: biên tập viên đăng nhập bằng tài khoản biên tập viên ở đây sẽ xem được danh sách các tài khoản. Biên tập viên sẽ có quyền thêm sửa xoá tài khoản

### 4.3.7 Quản lý bài phản biện




➤ Chức năng: nhà phản biện sẽ đăng nhập vào tài khoản nhà phản biện sau đó sẽ nhận được danh sách các bài mà biên tập viên gửi đến ở đây nhà phản biện có thể xem file của tác giả hoặc tải xuống và khi ấn vào chỉnh sửa sẽ hiện ra thông tin như sau:





➤ Chức năng: sau đó phản biện sẽ tải bài phản biện lên và chọn đề xuất rồi gửi cho biên tập viên

### 4.3.8 Giao diện trang chủ




**Tap chí khoa học Việt Nam trực tuyến - VNIES**


[Home](#)
[Đăng nhập](#)

Tap chí khoa học Việt Nam Trực tuyến là một dịch vụ cho phép độc giả tiếp cận tri thức khoa học được xuất bản tại Việt Nam và nâng cao hiểu biết của thế giới về nền học thuật của Việt Nam. Để biết thêm thông tin về VJOL và cách thức tham gia dịch vụ, xin vui lòng tham khảo trang Giới thiệu. Vietnam JOL do INASP tài trợ.


[Tap chí trên VNIES](#)
[Danh mục tap chí khoa học Việt Nam](#)



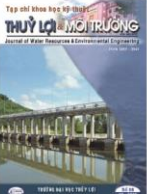
**Tap chí Khoa học và Công nghệ Việt Nam - A**  
 2024-11-08T00:00:00  
[Xem tap chí](#)




**Vietnam Journal of Science, Technology and Engineering - C**  
 2024-01-08T00:00:00  
[Xem tap chí](#)




**Tap chí Khoa học - Trường Đại học Sư phạm TP Hồ Chí Minh**  
 2024-02-06T00:00:00  
[Xem tap chí](#)




**Tap chí Khoa học Kỹ thuật Thủy lợi và Môi trường**  
 2024-03-06T00:00:00  
[Xem tap chí](#)



**Journal Science and Technology Water Resources**  
 2024-05-06T00:00:00  
[Xem tap chí](#)



**Tap chí Khí tượng Thủy văn**  
 2024-12-06T00:00:00  
[Xem tap chí](#)

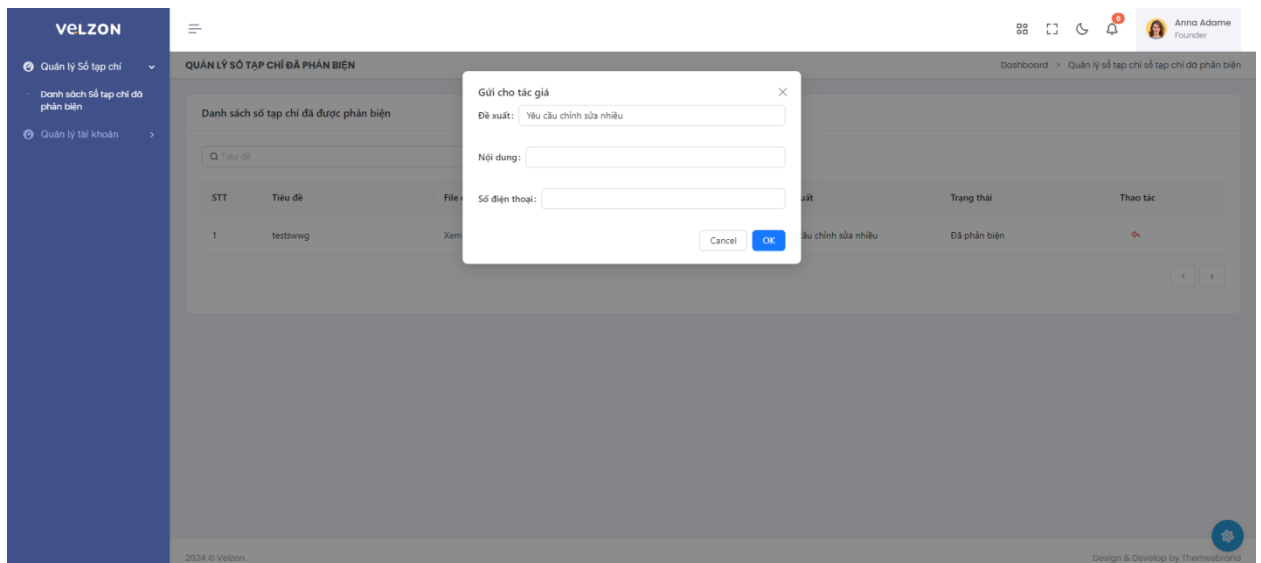


BỘ KHOA HỌC VÀ CÔNG NGHỆ - MINISTRY OF SCIENCE AND TECHNOLOGY OF VIETNAM  
 CỤC THÔNG TIN KHOA HỌC VÀ CÔNG NGHỆ QUỐC GIA - NATIONAL AGENCY FOR SCIENCE AND TECHNOLOGY INFORMATION  
 Số 107/35 24-36 Lý Thường Kiệt, Q. Hoàn Kiếm, TP. Hà Nội  
 Điện thoại: (2024) 39 340 000, Fax: 3939 770 553, E-mail: [doangpt1230@gmail.com](mailto:doangpt1230@gmail.com)

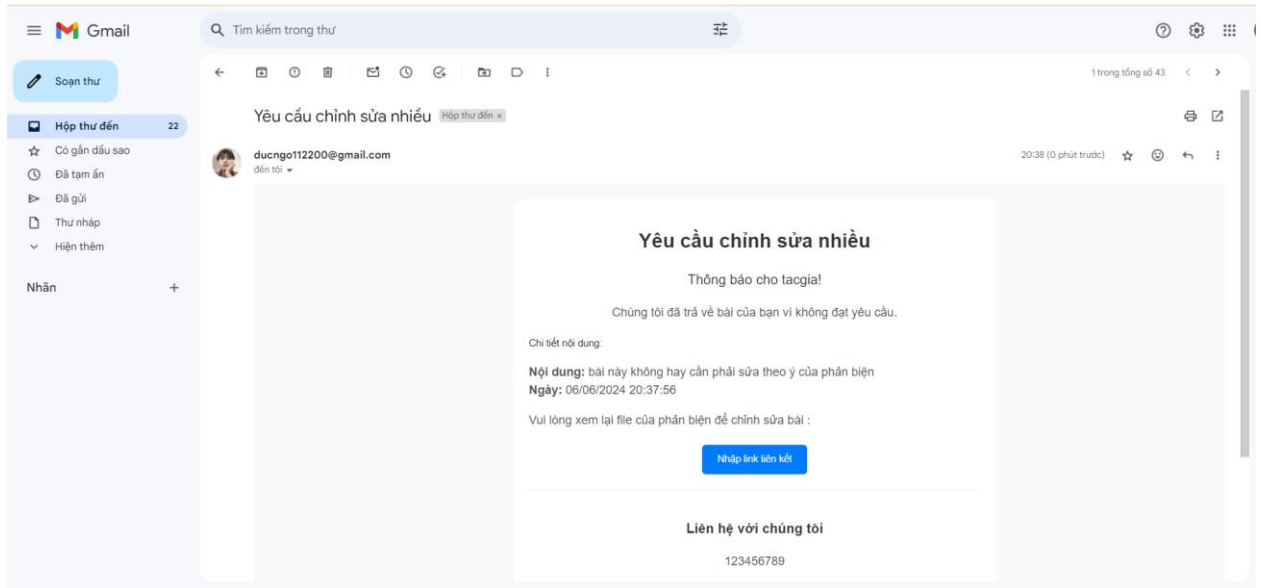


➤ Chức năng: ở đây độc giả có thể xem được thông tin về bài tạp chí sau khi đã được biên tập viên đăng tải

#### 4.3.9 Giao diện gửi email thông báo cho tác giả







➤ Chức năng: biên tập viên có thể trả bài về cho tác giả và thông báo cho tác giả qua email kèm với file phản biện đã gửi để tác giả có thể tham khảo

## KẾT LUẬN

Sau quá trình tìm hiểu, nghiên cứu, thực hiện, em đã học hỏi và hoàn thành xong đề tài **“Xây dựng ứng dụng thực hiện quy trình xuất bản cho tạp chí khoa học”**. Xây dựng website có thể mang lại những lợi ích nhất định như sau:

Với một hệ thống ứng dụng tự động, các bước từ việc gửi bài viết, đánh giá, phản biện, đến xuất bản đều được thực hiện nhanh chóng và chính xác hơn. Điều này giúp giảm thiểu các sai sót và lỗi quy trình, tiết kiệm thời gian và nguồn lực. Các tác giả và nhà nghiên cứu sẽ có trải nghiệm thuận tiện hơn khi tương tác với tạp chí, từ đó tăng cường sự hợp tác và chất lượng nghiên cứu được xuất bản.

Ngoài ra, ứng dụng này còn mang lại lợi ích lớn trong việc lưu trữ và truy xuất thông tin. Một hệ thống số hóa không chỉ giúp dễ dàng lưu trữ các bài viết, dữ liệu phản biện và các tài liệu liên quan mà còn cho phép truy cập nhanh chóng và chính xác khi cần thiết. Điều này đặc biệt hữu ích cho việc tìm kiếm tài liệu tham khảo, phân tích xu hướng nghiên cứu, và tổng hợp thông tin cho các báo cáo khoa học. Hơn nữa, việc số hóa còn hỗ trợ bảo mật thông tin tốt hơn, hạn chế nguy cơ mất mát dữ liệu so với các hệ thống lưu trữ truyền thống.

Tuy nhiên, website vẫn còn nhiều nhược điểm như giao diện người dùng chưa thân thiện: Một số người dùng có thể gặp khó khăn trong việc sử dụng website nếu giao diện không trực quan và dễ sử dụng. Điều này có thể làm giảm trải nghiệm người dùng và khiến họ ít muốn sử dụng website.

Về hướng phát triển đề tài, em sẽ thực hiện như sau:

Cải thiện giao diện người dùng: Tối ưu hóa giao diện để thân thiện và dễ sử dụng hơn, đảm bảo mọi người dùng. Tác giả, biên tập viên, nhà phản biện và độc giả đều có thể dễ dàng tương tác với hệ thống. Sử dụng phản hồi từ người dùng để cải thiện và nâng cấp giao diện liên tục.

Tối ưu hóa hiệu suất và khả năng mở rộng: Sử dụng các công nghệ và phương pháp tối ưu hóa để cải thiện hiệu suất của website. Đảm bảo hệ thống có thể mở rộng dễ dàng khi lượng người dùng và dữ liệu tăng lên.

Qua đây, em xin được cảm ơn **Khoa Công Nghệ Thông Tin** đã cho em cơ hội để thực hiện một đề tài thú vị như thế này. Và em cũng xin được cảm ơn **Thầy Trần Quý Nam** đã hướng dẫn rất nhiệt tình và tận tụy trong quá trình em làm đồ án. Em có thể trong khi làm đồ án này còn có nhiều thiếu sót cũng như kinh nghiệm chưa có nhiều mong thầy đóng góp ý kiến để khắc phục. Cuối cùng, em xin chân thành cảm ơn!

## TÀI LIỆU THAM KHẢO

- [1]. <https://material.angular.io/>
- [2]. <https://angular.io/>
- [3]. <https://cloud.google.com/dialogflow/docs>
- [4]. [https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp)
- [5]. <https://nodejs.org/docs/latest/api/>
- [6]. <https://sqladvice.com/tong-quan-ve-sql-server>