

## TRƯỜNG ĐẠI HỌC SƯ PHẠM HÀ NỘI KHOA CÔNG NGHỆ THÔNG TIN

**----------**

**KHÓA LUẬN TỐT NGHIỆP**

## ĐỀ TÀI:

PHÁT TRIỂN PHẦN MỀM HỖ TRỢ XẾP LỊCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN THẠC SĨ

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **PGS. TS. Trần Đăng Hưng** |
| **Sinh viên thực hiện:** | **Hà Văn Dũng** |
| **Lớp:** | **K69C** |
| **Mã sinh viên:** | **695105026** |

## Hà Nội, tháng 04 năm 2023

## LỜI CẢM ƠN

Bắt đầu khóa luận của mình, tôi xin chân thành cảm ơn thầy PGS. TS Trần Đăng Hưng và các thầy cô trong khoa Công nghệ thông tin tại trường Đại học Sư Phạm Hà Nội, những người đã dành thời gian và nỗ lực tận tình hướng dẫn, giúp đỡ tôi trong quá trình nghiên cứu và viết khóa luận. Qua đó, tôi đã học được nhiều kinh nghiệm quý báu về cách tiếp cận và giải quyết các vấn đề chuyên môn, cũng như cách tổ chức, triển khai và trình bày một bài nghiên cứu khoa học một cách chính xác và khoa học.Mặc dù đã cố gắng hết sức để thực hiện đề tài một cách tốt nhất, song vì còn hạn chế về mặt kiến thức và kinh nghiệm nên không tránh khỏi những thiếu sót. Tôi rất mong nhận được những ý kiến đóng góp quý giá từ thầy cô.

Tôi cũng muốn gửi lời cảm ơn đến các thầy, cô giáo trong trường, những người đã truyền đạt cho tôi những kiến thức, kỹ năng và tinh thần trách nhiệm trong suốt quá trình học tập. Những kiến thức này sẽ là nền tảng cho tôi phát triển sự nghiệp và hoàn thiện bản thân trong tương lai.

Dù tôi đã cố gắng hết sức để hoàn thành đề tài của mình, tôi nhận thức rõ rằng vẫn còn rất nhiều điều để học hỏi và cải thiện. Vì vậy, tôi rất mong nhận được những ý kiến đóng góp quý báu từ các thầy, cô giáo, để tôi có thể tiếp tục hoàn thiện và phát triển bản thân.

Cuối cùng, tôi xin gửi lời cảm ơn chân thành tới thầy Trần Đăng Hưng và các thầy, cô giáo trong khoa Công nghệ thông tin, cùng tất cả những người đã đóng góp và hỗ trợ tôi trong suốt quá trình học tập và nghiên cứu tại trường Đại học Sư Phạm Hà Nội. Tôi mong rằng quý thầy, cô giáo sẽ luôn giữ vững tinh thần trách nhiệm và nhiệt huyết trong công tác giảng dạy và nghiên cứu, để dẫn dắt và hướng dẫn các thế hệ trẻ tiếp nối phát triển đất nước.

Tôi xin chân thành cảm ơn!

Hà Nội, ngày......tháng. năm 2023

Sinh viên

Hà Văn Dũng

## LỜI CAM ĐOAN

Tôi xin cam đoan rằng kết quả đạt được trong khóa luận tốt nghiệp “Phát triển phần mềm hỗ trợ xếp lịch hội đồng bảo vệ luận văn Thạc sĩ” là hoàn toàn do sự nghiên cứu và cố gắng của bản thân tôi, với sự hỗ trợ và chỉ dẫn nhiệt tình của PGS. TS Trần Đăng Hưng - người thầy đã luôn sát cánh bên cạnh và giúp đỡ tôi trong quá trình thực hiện đề tài.

Những nội dung được trình bày trong khóa luận là sự kết hợp giữa kiến thức cá nhân của tôi và các nguồn tài liệu khác, được tổng hợp và phân tích cẩn thận trước khi được bổ sung vào khóa luận. Tất cả các tài liệu tham khảo cũng được trích dẫn rõ ràng và đầy đủ trong phần tài liệu tham khảo của khóa luận.

Tôi cam đoan những lời trên là sự thật và nếu có bất kỳ sai sót hoặc thiếu sót nào trong quá trình thực hiện khóa luận, tôi xin hoàn toàn chịu trách nhiệm và sẵn sàng bổ sung, sửa đổi để đạt được kết quả tốt nhất. Tôi cũng hy vọng rằng khóa luận của tôi sẽ đóng góp một phần nhỏ vào công cuộc phát triển khoa học và giáo dục của khoa công nghệ thông tin nói riêng và trường Đại học Sư phạm Hà Nội nói chung.

|  |
| --- |
| Hà Nội, ngày......tháng. năm 2023 |
| Tác giả  Hà Văn Dũng |

Contents

[TRƯỜNG ĐẠI HỌC SƯ PHẠM HÀ NỘI KHOA CÔNG NGHỆ THÔNG TIN 1](#_Toc122959730)

[ĐỀ TÀI: 1](#_Toc122959731)

[Hà Nội, tháng 12 năm 2022 1](#_Toc122959732)

[LỜI CẢM ƠN 2](#_Toc122959733)

[LỜI CAM ĐOAN 3](#_Toc122959734)

[LỜI NÓI ĐẦU 6](#_Toc122959735)

[CHƯƠNG I: NGHIÊN CỨU GIẢI PHÁP CÔNG NGHỆ 7](#_Toc122959736)

[1.1 Lựa chọn kiến trúc hệ thống 7](#_Toc122959737)

[Hình 1.1. Sơ đồ client-server nâng cao 7](#_Toc122959738)

[1.2 Lựa chọn nền tảng công nghệ phát triển hệ thống 7](#_Toc122959739)

[Hình 1.2. Phần back-end và front-end trong mô hình client – server 8](#_Toc122959740)

[1.3 Các giải pháp kỹ thuật chủ yếu được nghiên cứu trong đề tài 8](#_Toc122959741)

[1.3.1 Xây dựng hệ thống bằng Visual Studio và ngôn ngữ lập trình C#, Blazor 9](#_Toc122959742)

[a. Môi trường phát triển 9](#_Toc122959743)

[Hình 1.4 Môi trường phát triển Visual Studio 9](#_Toc122959744)

[b. Ngôn ngữ lập trình C# 9](#_Toc122959745)

[Hình 1.6 Ngôn ngữ lập trình C# (.NETCore) 10](#_Toc122959746)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU 10](#_Toc122959747)

[2.1 Lập kế hoạch và khám phá yêu cầu 10](#_Toc122959748)

[2.1 Giới thiệu 10](#_Toc122959749)

[2.1.1 Mục đích 10](#_Toc122959750)

[2.1.2 Phạm vi 10](#_Toc122959751)

[2.2. Mô tả chung 11](#_Toc122959752)

[2.2.1 bối cảnh phát triển hệ thống phần mềm 11](#_Toc122959753)

[Là một trong những khoa đi đầu về giảng dạy, nghiên cứu áp dụng các thành tựu tin học thì khoa công nghệ thông tin đang dần hướng đến thay đổi quy trình cũ, áp dụng kĩ thuật phần mềm không những vào trong giảng dạy mà còn trong các quy trình quản lí nghiệp vụ. Điều này đang là một hướng đi mới, cần được phát triển mạnh vì khi áp dụng thực tế đã nhận được những kết quả tích cực, công việc giảng dạy cũng như quản lí đạt được hiệu suất cao, chuyên nghiệp, dễ dàng áp dụng và đồng bộ các quy trình. 11](#_Toc122959754)

[2.2.1 Các chức năng thực hiện 11](#_Toc122959755)

[2.2.2 Các lớp người dùng 12](#_Toc122959756)

[2.2.3 Môi trường vận hành 12](#_Toc122959757)

[2.3. Xác định tập yêu cầu 12](#_Toc122959758)

[2.3.1 Các yêu cầu chức năng 12](#_Toc122959759)

[2.3.2 Các yêu cầu phi chức năng 15](#_Toc122959760)

[2.2 Phân tích yêu cầu 16](#_Toc122959761)

[2.2.1 Mô hình hóa ca sử dụng 16](#_Toc122959762)

[2.2.2 Mô hình hóa các tiến trình nghiệm vụ chính của hệ thống sử dụng biểu đồ hoạt động 20](#_Toc122959763)

[2.3.3 Thiết kế tầng lưu trữ dữ liệu 21](#_Toc122959764)

## LỜI NÓI ĐẦU

Trong bối cảnh hiện nay, việc xếp lịch hội đồng bảo vệ luận văn Thạc sĩ tại trường Đại học Sư phạm Hà Nội vẫn đang được thực hiện bằng phương pháp thủ công, tốn nhiều thời gian và công sức trong quá trình sắp xếp. Tuy nhiên, việc áp dụng công nghệ vào quản lý lịch làm việc đã trở thành xu thế hiện nay và được rất nhiều tổ chức và doanh nghiệp áp dụng để nâng cao hiệu quả công việc.

Với mục tiêu giải quyết vấn đề trên, khóa luận này sẽ tập trung trình bày cơ sở lý thuyết và xây dựng một hệ thống Webapp đáp ứng nhu cầu số hóa việc xếp lịch hội đồng bảo vệ luận văn Thạc sĩ tại trường Đại học Sư phạm Hà Nội. Nhờ đó, việc xếp lịch sẽ trở nên dễ dàng và nhanh chóng hơn, tiết kiệm thời gian và năng lực cho người xếp lịch. Khóa luận sẽ tập trung vào việc trình bày một cách cụ thể, chi tiết về quy trình xây dựng hệ thống, bao gồm cả cơ sở lý thuyết và thực tiễn.

Tất cả những nội dung được trình bày trong khóa luận này đều được tổng hợp từ nhiều nguồn tài liệu khác nhau và được trích dẫn rõ ràng trong phần tài liệu tham khảo của khóa luận. Tôi xin cam đoan rằng kết quả đạt được trong khóa luận tốt nghiệp này là do sự nghiên cứu và làm việc của cá nhân tôi dưới sự hướng dẫn tận tình của PGS. TS Trần Đăng Hưng. Nếu có bất kỳ sai sót hay khác biệt nào, tôi xin hoàn toàn chịu trách nhiệm về những nội dung trình bày trong khóa luận của mình.

## CHƯƠNG I: GIỚI THIỆU

Trong chương I, khóa luận sẽ giới thiệu về đề tài và lý do tại sao đề tài này được lựa chọn để nghiên cứu. Sau đó, khóa luận sẽ trình bày mục đích và phạm vi của nghiên cứu, cùng với những nội dung chính mà khóa luận sẽ đề cập trong các phần tiếp theo.

## Lý do chọn đề tài

Nhận thấy trong công việc quản lý xếp lịch bảo vệ luận văn thạc sĩ, khoa vẫn đang thực hiện một quy trình thủ công và truyền thống, quản lý chưa được đồng bộ và thực sự liên kết. Vì vậy hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ sẽ giúp cho việc quản lý và xếp lịch hội đồng bảo vệ luận văn thạc sĩ dễ dàng, quy trình đồng nhất ngắn gọn, đạt hiệu quả cao. Từ đó có thể thuận tiện hơn trong công việc xếp lịch bảo vệ luận văn thạc sĩ để có thể tiết kiệm thời gian xếp lịch.

Trong thời đại của công nghệ thông tin, việc ứng dụng phần mềm vào quản lý và giảng dạy đã trở thành xu hướng tất yếu. Khoa công nghệ thông tin là một trong những khoa tiên phong trong việc áp dụng kĩ thuật phần mềm vào quy trình giảng dạy và quản lý nghiệp vụ. Việc này giúp cho quy trình quản lý và giảng dạy đạt được hiệu suất cao, chuyên nghiệp, dễ dàng áp dụng và đồng bộ các quy trình với nhau.

Tuy nhiên, trong quá trình quản lý xếp lịch bảo vệ luận văn thạc sĩ, khoa công nghệ thông tin vẫn đang áp dụng một quy trình thủ công và truyền thống. Việc quản lý và xếp lịch chưa được đồng bộ và thực sự liên kết, gây ra khó khăn và tốn thời gian trong việc xếp lịch bảo vệ luận văn thạc sĩ. Do đó, việc phát triển một hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ là cần thiết để giúp cho quy trình quản lý và xếp lịch trở nên dễ dàng hơn và đồng nhất hơn.

Hệ thống này sẽ giúp cho công tác xếp lịch bảo vệ luận văn thạc sĩ trở nên đơn giản hơn, tối ưu thời gian và giảm thiểu tình trạng trùng lịch hoặc lộn lịch. Đồng thời, việc quản lý và tra cứu thông tin liên quan đến hội đồng bảo vệ cũng được đơn giản hóa hơn, giúp cho việc quản lý và xử lý thông tin trở nên chính xác và nhanh chóng hơn.

Tóm lại, việc phát triển hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ là một bước tiến mới và đúng đắn của khoa công nghệ thông tin. Đây là cách để đáp ứng nhu cầu quản lý và xếp lịch hiệu quả hơn, đồng thời giúp cho quy trình giảng dạy và quản lý nghiệp vụ được cải thiện và đồng bộ hơn.

## Mục đích và phạm vi nghiên cứu

Mục đích của nghiên cứu này là xây dựng một hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ cho trường đại học Sư phạm Hà Nội. Nghiên cứu sẽ giúp cải thiện quy trình tổ chức hội đồng bảo vệ luận văn thạc sĩ, từ đó tăng hiệu quả và giảm thời gian cho quá trình này.

Phạm vi nghiên cứu bao gồm các giai đoạn trong quy trình xếp lịch hội đồng bảo vệ luận văn thạc sĩ, bao gồm lựa chọn thành viên trong hội đồng và các ứng viên bảo vệ luận văn.

Để đạt được mục đích và phạm vi nghiên cứu này, sẽ sử dụng phương pháp nghiên cứu định tính để thu thập thông tin từ các chuyên gia và giảng viên trong lĩnh vực này. Các phương pháp thu thập thông tin bao gồm phỏng vấn, khảo sát và phân tích tài liệu liên quan. Các kết quả thu được sẽ được sử dụng để phát triển một hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ với các tính năng tiên tiến, đáp ứng nhu cầu của trường đại học Sư phạm Hà Nội.

## Nội dung chính khóa luận đề cập

Chương II sẽ đi sâu vào việc nghiên cứu và phân tích các giải pháp công nghệ phục vụ cho quá trình xây dựng hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ. Khóa luận sẽ phân tích kỹ các công nghệ được chọn để tìm ra lợi ích mà chúng đem lại cho hệ thống.

Chương III sẽ trình bày quá trình phân tích và thiết kế hệ thống. Khóa luận sẽ đề cập đến các yếu tố cần thiết để xây dựng hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ, bao gồm các khả năng và tính năng mà hệ thống cần có.Chương IV. Xây dựng hệ thống và kết quả đạt được.

Chương IV là phần trọng tâm của khóa luận, với nội dung chính là quá trình xây dựng hệ thống và kết quả đạt được. Khóa luận sẽ mô tả chi tiết quá trình phát triển hệ thống, các công nghệ và công cụ được sử dụng, cùng với kết quả đạt được trong quá trình triển khai hệ thống.

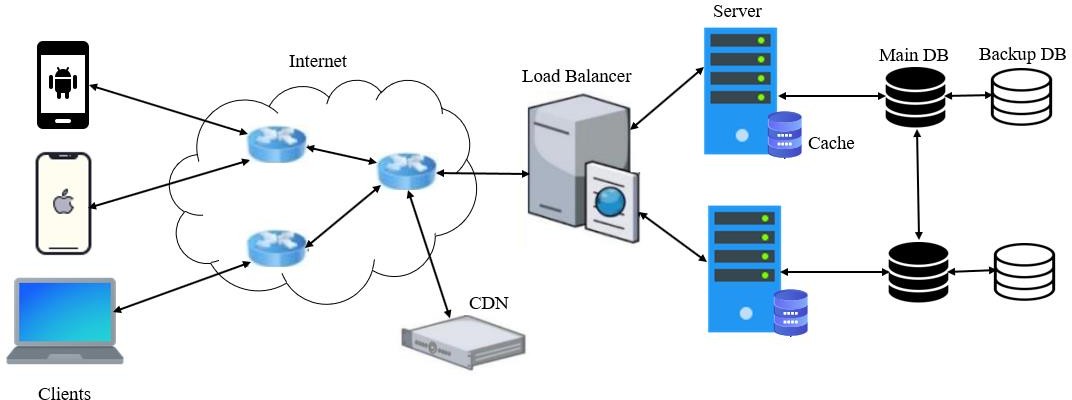
Cuối cùng, trong chương V, khóa luận sẽ tổng hợp và đưa ra kết luận tổng quan về quá trình nghiên cứu và xây dựng hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ. Ngoài ra, khóa luận cũng sẽ đưa ra hướng phát triển cho hệ thống trong tương lai, nhằm cải thiện hiệu quả và tính ứng dụng của hệ thống.

## CHƯƠNG II: NGHIÊN CỨU GIẢI PHÁP CÔNG NGHỆ

Trong phần này của khóa luận, chúng ta sẽ tiến hành một quá trình đánh giá và lựa chọn kiến trúc hệ thống phù hợp nhất để xây dựng ứng dụng hỗ trợ xếp lịch hội đồng bảo vệ luận văn Thạc sĩ. Sự tập trung nghiên cứu vào các giải pháp công nghệ phù hợp nhất để phát triển phần mềm được đề cập trong phần này. Chúng ta sẽ tìm hiểu các nền tảng công nghệ khác nhau để đưa ra quyết định về nền tảng công nghệ phù hợp nhất cho dự án. Đồng thời, chúng ta sẽ giải thích các giải pháp kỹ thuật được áp dụng trong khóa luận, đảm bảo tính khác biệt so với các phần trước. Nói cách khác, phần này sẽ đưa ra một bức tranh toàn diện và chi tiết hơn về cách thức triển khai công nghệ trong khóa luận.

## Lựa chọn kiến trúc hệ thống

Kiến trúc máy khách - máy chủ (client-server) là một mô hình phân tán trong đó máy chủ lưu trữ, quản lý và cung cấp dịch vụ cho các máy khách kết nối thông qua mạng hoặc internet. Mô hình này còn được gọi là mô hình mạng tính toán hoặc mạng khách-chủ, do tất cả các yêu cầu và dịch vụ được phân phối qua mạng.



**Hình 1.1. Sơ đồ client-server nâng cao**

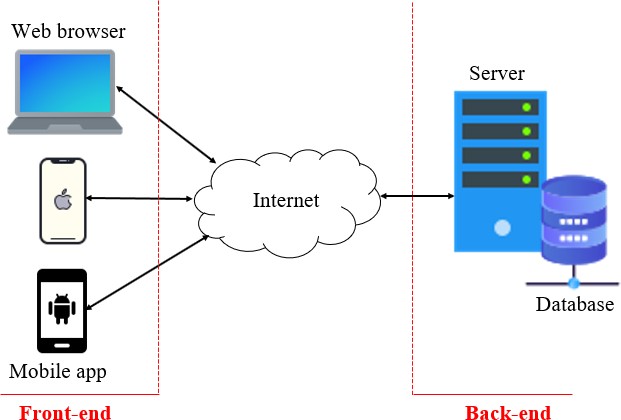
Trong kiến trúc này, các máy khách (client) là các phần mềm hoặc ứng dụng nhận thông tin đầu vào từ người dùng và gửi yêu cầu đến máy chủ (server) để xử lý. Máy chủ nhận và xử lý các yêu cầu từ máy khách và trả về kết quả tương ứng.

Để tối ưu hóa việc sử dụng tài nguyên, kiến trúc client-server còn có thêm bộ cân bằng tải (balancer) để phân phối lưu lượng mạng đến một nhóm máy chủ hỗ trợ. Bộ cân bằng tải giúp cân bằng tải trên các máy chủ và giảm thiểu tình trạng quá tải trên một máy chủ duy nhất.

Mô hình máy khách - máy chủ đã trở thành một mô hình phổ biến trong phát triển phần mềm và các ứng dụng trên internet. Nó cho phép xử lý dữ liệu lớn và hỗ trợ nhiều người dùng truy cập cùng một lúc.

Từ sơ đồ kiến trúc client – server **Hình 1.1** nhận thấy việc xây dựng phần mềm hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ là phù hợp. Với kiến trúc này nhà phát triển có thể dễ dàng cập nhật ứng dụng trên trên môi trường Webapp mà không cần yêu cầu người dùng phải thực hiện cập nhật trên máy local của người dùng.

## Lựa chọn nền tảng công nghệ phát triển hệ thống



**Hình 1.2. Phần back-end và front-end trong mô hình client – server**

Hiện nay, có rất nhiều công nghệ để xây dựng webapp, phổ biến nhất là các ngôn ngữ lập trình web như PHP, JavaScript, Python, Ruby và Java. Tuy nhiên, với sự phát triển của công nghệ .NET Core, một số nền tảng công nghệ mới đã xuất hiện và trở nên phổ biến như Blazor.



**Hình 1.3. Blazor framework**

Blazor là một framework của Microsoft cho phép phát triển ứng dụng web động và tĩnh sử dụng C# và .NET trên trình duyệt. Nó cho phép lập trình viên tạo ra các ứng dụng web động mà không cần sử dụng JavaScript, giúp tiết kiệm thời gian phát triển và giảm thiểu lỗi.

Blazor hỗ trợ hai mô hình lập trình là Blazor Server và Blazor WebAssembly. Blazor Server cho phép tạo các ứng dụng web động sử dụng ASP.NET Core trên máy chủ. Blazor WebAssembly cho phép tạo các ứng dụng web động chạy trên trình duyệt.

Blazor sử dụng Razor, một ngôn ngữ lập trình dành cho phát triển web của Microsoft, cùng với C# và .NET. Nó cung cấp nhiều tính năng như kiểm tra lỗi tại thời điểm biên dịch, tương thích ngược với các phiên bản .NET Core và hỗ trợ tích hợp dễ dàng với các công cụ phát triển khác của Microsoft như Visual Studio.

Với Blazor, lập trình viên có thể phát triển các ứng dụng web động và tĩnh đa nền tảng với khả năng tương thích ngược, tính năng phong phú và độ tin cậy cao, giúp cho việc phát triển ứng dụng web trở nên dễ dàng hơn và nhanh chóng hơn.

Điểm nổi bật của Blazor đó là khả năng tương tác với UI bằng C# code, không cần yêu cầu sử dụng Javascript. Blazor sử dụng Razor Component cho phép trộn code C# và HTML, tận dụng sức mạnh của thư viện và framework .NET trong ứng dụng Blazor, cùng với tốc độ ứng dụng web nhanh bởi code C# được biên dịch thành các đoạn mã nhị phân nhờ WebAssembly, rất gần với ngôn ngữ máy mà trình duyệt hiểu được. Blazor cũng hỗ trợ hầu hết các trình duyệt hiện đại và không cần cài thêm bất kỳ plugin nào, cùng với việc tận dụng tài nguyên ở client nên không cần server quá mạnh và không cần cài .NET Core trên server.

Chính vì vậy, ở khóa luận này sử dụng C# và Blazor là một lựa chọn hợp lý để xây dựng hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ. Sử dụng những công nghệ này sẽ giúp tăng tính bảo mật và độ tin cậy của hệ thống, đồng thời cung cấp tính năng mạnh mẽ cho việc phát triển ứng dụng và tăng tốc độ hoạt động của hệ thống.

## 2.3 Các giải pháp kỹ thuật chủ yếu được nghiên cứu trong đề tài

Hiện nay có rất nhiều kỹ thuật để xây dựng một hệ thống. Trong khóa luận này sẽ sử dụng những kỹ thuật sau để xây dựng hệ thống: ngôn ngữ lập trình C# (.NET Core), Blazor, cơ sở dữ liệu SQL, Nhibernate framework, Ant Design Blazor.

1. Trong đề tài này, chúng tôi tập trung vào các giải pháp kỹ thuật chủ yếu để xây dựng một hệ thống quản lý dữ liệu bằng cách sử dụng công nghệ C# (.NET Core) và Blazor. Các giải pháp kỹ thuật chính bao gồm:
2. Thiết kế kiến trúc hệ thống: Chúng tôi sử dụng mô hình kiến trúc Clean để đảm bảo tính tổ chức và dễ bảo trì của hệ thống. Đây là một mô hình phát triển phần mềm có cấu trúc sạch sẽ, đảm bảo sự tách biệt giữa các lớp logic, cơ sở dữ liệu và giao diện người dùng.
3. Lựa chọn công nghệ backend: Chúng tôi sử dụng ngôn ngữ C# và framework .NET Core để phát triển phần backend của hệ thống. .NET Core là một nền tảng mã nguồn mở, đa nền tảng và đáp ứng, cung cấp nhiều tính năng mạnh mẽ để xây dựng các ứng dụng web hiệu quả.
4. Lựa chọn công nghệ frontend: Chúng tôi sử dụng Blazor framework để phát triển phần giao diện người dùng. Blazor là một framework lập trình web đa nền tảng cho phép lập trình viên tương tác và làm việc với UI bằng C# code, giúp tiết kiệm thời gian và công sức so với việc sử dụng JavaScript.
5. Quản lý dữ liệu: Chúng tôi sử dụng Entity Framework Core để quản lý cơ sở dữ liệu. Đây là một thư viện ORM (Object-Relational Mapping) của .NET Core, cung cấp nhiều tính năng để giúp lập trình viên tương tác với cơ sở dữ liệu một cách dễ dàng và hiệu quả.
6. Triển khai: Chúng tôi sử dụng các dịch vụ đám mây như Azure hoặc AWS để triển khai hệ thống và đảm bảo tính sẵn sàng, tin cậy và có thể mở rộng của hệ thống.

## 2.3.1 Xây dựng hệ thống bằng Visual Studio và ngôn ngữ lập trình C#, Blazor

**Framework**

1. **Môi trường phát triển**

Việc xây dựng hệ thống sử dụng Visual Studio và ngôn ngữ lập trình C# cùng với Blazor Framework đem lại nhiều lợi ích cho nhà phát triển. Trong đó, Visual Studio cung cấp môi trường phát triển hoàn chỉnh với nhiều tính năng hỗ trợ giúp tối ưu quá trình phát triển, bao gồm tích hợp công cụ quản lý phiên bản, đa nền tảng, xây dựng và triển khai ứng dụng một cách dễ dàng.



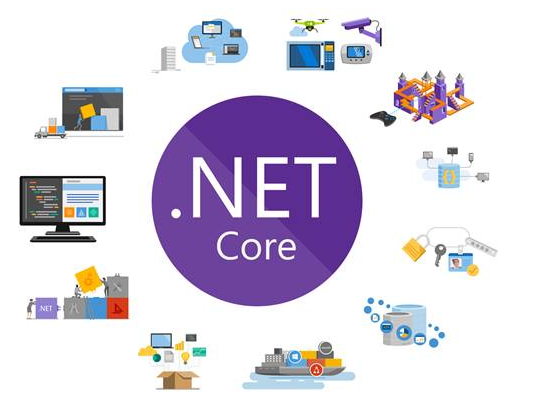
**Hình 1.4 Môi trường phát triển Visual Studio**

## Ngôn ngữ lập trình C#

C# là một ngôn ngữ lập trình hướng đối tượng được xây dựng bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

Đến năm 2013, Microsoft định hướng đi đa nền tảng và phát triển .NET core. .NET core hiện được sử dụng trong các ứng dụng Universal Windows platform và ASP.NET Core. Từ đây, C# có thể được sử dụng để phát triển các loại ứng dụng đa nền tảng trên các hệ điều hành khác nhau (Windows, Linux, MacOS,…)

Ngôn ngữ lập trình C# cũng là một lựa chọn lý tưởng khi phát triển ứng dụng do nó có tính bảo mật cao, dễ dàng bảo trì và phát triển, đồng thời được hỗ trợ bởi nhiều framework và thư viện phổ biến.

****

**Hình 1.5 Ngôn ngữ lập trình C# (.NETCore)**

## Cơ sở dữ liệu quan hệ

Cơ sở dữ liệu quan hệ (Relational Database) là loại cơ sở dữ liệu sử dụng các bảng và các quan hệ giữa chúng để lưu trữ và quản lý dữ liệu. Các bảng chứa các hàng và cột dữ liệu, mỗi hàng tương ứng với một bản ghi và mỗi cột chứa dữ liệu của một thuộc tính cụ thể. Các quan hệ được xác định bằng cách sử dụng khóa chính và khóa ngoại để liên kết các bảng với nhau. Cơ sở dữ liệu quan hệ là một trong những hệ thống quản lý cơ sở dữ liệu phổ biến nhất và được sử dụng rộng rãi trong các ứng dụng doanh nghiệp.

Ưu điểm của cơ sở dữ liệu quan hệ là khả năng lưu trữ và quản lý dữ liệu lớn, có tính chất cấu trúc và dễ dàng thao tác. Các bảng có thể được liên kết với nhau thông qua các quan hệ để tạo ra thông tin kết hợp từ nhiều bảng, giúp cho việc tìm kiếm, truy xuất và xử lý dữ liệu trở nên dễ dàng hơn. Hơn nữa, cơ sở dữ liệu quan hệ đảm bảo tính toàn vẹn dữ liệu bằng cách kiểm tra ràng buộc toàn vẹn dữ liệu, giúp tránh được các lỗi dữ liệu và đảm bảo tính chính xác của thông tin.

Tuy nhiên, cơ sở dữ liệu quan hệ cũng có nhược điểm. Việc thiết kế cơ sở dữ liệu quan hệ phức tạp và tốn nhiều thời gian, đặc biệt là khi có nhiều bảng và quan hệ giữa chúng. Hơn nữa, cơ sở dữ liệu quan hệ có khả năng bị lỗi khi dữ liệu phát triển lớn và cần phải tối ưu hóa để đảm bảo hiệu suất. Ngoài ra, cơ sở dữ liệu quan hệ không thể đáp ứng các yêu cầu động và linh hoạt của các ứng dụng hiện đại.

Một vài hệ quản trị cơ sở dữ liệu phổ biến.



**Hình 1.6 Hệ quản trị cơ sở dữ liệu MySQL**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở được sử dụng phổ biến trên toàn thế giới. Nó là một phần mềm cơ sở dữ liệu tiêu chuẩn và đáng tin cậy, được phát triển và bảo trì bởi Oracle Corporation. MySQL được phát hành dưới giấy phép GNU General Public License, cho phép người dùng sử dụng, sao chép, chỉnh sửa và phân phối miễn phí.

MySQL có khả năng xử lý các truy vấn và giao dịch với tốc độ nhanh và hiệu suất cao. Điều này là do MySQL sử dụng một số kỹ thuật tối ưu hóa để cải thiện khả năng chịu tải của hệ thống. MySQL cũng hỗ trợ nhiều chức năng khác nhau để quản lý cơ sở dữ liệu, bao gồm sao lưu và phục hồi, xác thực người dùng, quản lý phiên, mã hóa dữ liệu và kiểm soát truy cập.

Một trong những ưu điểm của MySQL là tính linh hoạt trong việc cấu hình và quản lý. MySQL có khả năng hoạt động trên nhiều hệ điều hành khác nhau, bao gồm Windows, Linux và Mac OS X, cho phép người dùng lựa chọn hệ điều hành phù hợp với nhu cầu của họ. Hơn nữa, MySQL cung cấp các công cụ để tối ưu hóa cơ sở dữ liệu, tăng cường bảo mật và quản lý các tính năng của hệ thống.

Tuy nhiên, như mọi công nghệ khác, MySQL cũng có một số nhược điểm. Một trong những vấn đề thường gặp nhất là khả năng quản lý cơ sở dữ liệu lớn. Với cơ sở dữ liệu lớn, MySQL có thể gặp khó khăn trong việc xử lý các truy vấn và tương tác với dữ liệu. Ngoài ra, MySQL cũng không hỗ trợ đầy đủ các tính năng mở rộng, chẳng hạn như khả năng tìm kiếm toàn văn bản và xử lý dữ liệu phức tạp.

Có thể thấy rằng MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, linh hoạt và đáng tin cậy, được sử dụng phổ biến trên toàn thế giới.



**Hình 1.7 Hệ quản trị cơ sở dữ liệu SQL Server**

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft Corporation. SQL Server cho phép lưu trữ và quản lý dữ liệu trong các cơ sở dữ liệu quan hệ, đáp ứng nhu cầu của các doanh nghiệp với khả năng mở rộng và tính sẵn sàng cao.

SQL Server được phát triển trên nền tảng Windows, cho phép tích hợp với các ứng dụng phần mềm của Microsoft, cũng như hỗ trợ các giao thức chuẩn để liên kết với các hệ thống khác như Unix và Linux. Nó cung cấp các tính năng mạnh mẽ như truy vấn nhanh chóng, đa luồng, hỗ trợ dữ liệu đa phương tiện, khả năng đồng bộ hóa dữ liệu và nhiều tính năng khác.

SQL Server cũng được tích hợp với các công cụ hỗ trợ phát triển ứng dụng của Microsoft, như Visual Studio, để tạo ra các ứng dụng dựa trên cơ sở dữ liệu. Nó cũng có thể tích hợp với các công nghệ như .NET Framework, cho phép phát triển các ứng dụng đa nền tảng.

Một trong những ưu điểm lớn của SQL Server là khả năng quản lý cơ sở dữ liệu lớn và đa dạng, cho phép lưu trữ và quản lý các loại dữ liệu khác nhau như văn bản, hình ảnh và âm thanh. Nó cũng cung cấp các công cụ quản lý dữ liệu tiên tiến, bao gồm cả công cụ phục hồi dữ liệu để đảm bảo an toàn và tính khả dụng của dữ liệu.

Tuy nhiên, nhược điểm của SQL Server là giá thành cao hơn so với các hệ quản trị cơ sở dữ liệu khác và yêu cầu kỹ năng cao để triển khai và quản lý. Ngoài ra, SQL Server không hỗ trợ một số tính năng như các hệ quản trị cơ sở dữ liệu NoSQL và không phù hợp cho các ứng dụng web với khối lượng truy cập lớn và tốc độ truy xuất nhanh.



**Hình 1.8 Hệ quản trị cơ sở dữ liệu PostgreSQL**

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến được sử dụng rộng rãi trong các ứng dụng web và doanh nghiệp. Nó hỗ trợ các tính năng quản lý cơ sở dữ liệu quan trọng như đa người dùng, đồng bộ hóa, sao lưu và phục hồi, giao dịch ACID, xử lý đám mây và nhiều tính năng khác.

PostgreSQL có khả năng mở rộng tốt, giúp cho nó có thể xử lý tải lớn và phục vụ cho nhiều người dùng cùng một lúc mà không ảnh hưởng đến hiệu suất hoạt động. Ngoài ra, PostgreSQL còn cung cấp các chức năng nâng cao như truy vấn đa chiều, hỗ trợ JSON và XML, và có khả năng mở rộng để đáp ứng các yêu cầu kinh doanh cụ thể.

Một trong những ưu điểm lớn của PostgreSQL so với các hệ quản trị cơ sở dữ liệu khác là tính bảo mật. PostgreSQL có khả năng bảo vệ dữ liệu với cơ chế phân quyền và đăng nhập, giúp người quản trị cơ sở dữ liệu có thể quản lý người dùng và quyền truy cập dữ liệu một cách chặt chẽ. Ngoài ra, PostgreSQL cũng có thể xử lý các vấn đề bảo mật như mã hóa dữ liệu và hạn chế các lỗ hổng bảo mật khác.

Tuy nhiên, PostgreSQL cũng có một số nhược điểm. Vì là một hệ quản trị cơ sở dữ liệu mạnh mẽ, PostgreSQL có thể có một học hỏi dốt để làm quen và tối ưu hóa cho việc triển khai ban đầu. Ngoài ra, các tính năng bảo mật mạnh mẽ của PostgreSQL có thể gây khó khăn trong việc triển khai cũng như sử dụng cho người dùng không có kinh nghiệm.



**Hình 1.8 Hệ quản trị cơ sở dữ liệu SQLite**

SQLite là một hệ quản trị cơ sở dữ liệu quan hệ miễn phí và mã nguồn mở được phát triển bởi D. Richard Hipp vào năm 2000. Nó là một công cụ nhẹ, nhanh và đáng tin cậy cho các ứng dụng có nhu cầu lưu trữ dữ liệu nhỏ đến trung bình. SQLite được thiết kế để có khả năng chạy độc lập với bất kỳ ứng dụng nào mà không cần một hệ thống quản trị cơ sở dữ liệu riêng biệt.

SQLite cung cấp các tính năng quản lý cơ bản cho dữ liệu, bao gồm các câu lệnh SQL để tạo, thêm, sửa và xóa dữ liệu. Nó hỗ trợ hầu hết các loại dữ liệu như số, chuỗi, ngày tháng, v.v. SQLite cũng hỗ trợ các tính năng nâng cao như tạo chỉ mục, trigger và khóa để đảm bảo tính toàn vẹn của dữ liệu.

Một trong những ưu điểm của SQLite là tốc độ truy xuất dữ liệu nhanh và hiệu quả. SQLite không yêu cầu các quy trình đồng bộ hóa, do đó nó không bị ảnh hưởng bởi các tình huống xung đột và lỗi tương tranh. Điều này làm cho SQLite trở thành một giải pháp lý tưởng cho các ứng dụng đơn giản và nhẹ. Ngoài ra, việc triển khai và bảo trì của SQLite cũng rất đơn giản và ít tốn kém.

Tuy nhiên, nhược điểm của SQLite là không được thiết kế để xử lý các tập dữ liệu lớn hoặc ứng dụng có tính phân tán. Do đó, nếu bạn đang tìm kiếm một hệ quản trị cơ sở dữ liệu cho các ứng dụng lớn, phức tạp, phân tán và đòi hỏi khả năng mở rộng cao, thì SQLite có thể không phải là lựa chọn tốt nhất.

Với sự hỗ trợ rất tốt của các hệ quản trị cơ sở dữ liệu như đã đề cập ở trên thì việc áp dụng cơ sở dữ liệu quan hệ là một sự lựa chọn tốt cho hệ thống hỗ trợ xếp lịch hội đồng bảo vệ luận văn thạc sĩ.

## Một vài framework hỗ trợ

## Hibernate framework

Hibernate là một framework ORM (Object-Relational Mapping) mã nguồn mở được sử dụng để giải quyết các vấn đề liên quan đến việc lưu trữ đối tượng trong cơ sở dữ liệu quan hệ. Hibernate cho phép lập trình viên tập trung vào thiết kế đối tượng mà không cần phải quan tâm đến cách thức lưu trữ dữ liệu.

Mặc dù Hibernate được phát triển cho Java, nhưng các nhà phát triển C# cũng có thể sử dụng nó thông qua các thư viện và framework tương tự. Các thư viện như NHibernate và Entity Framework cho phép sử dụng Hibernate trong C#.

Sử dụng Hibernate trong C# có nhiều ưu điểm. Việc sử dụng Hibernate giúp giảm thời gian và công sức cần thiết để xây dựng lớp dữ liệu, giúp tiết kiệm thời gian và nỗ lực của lập trình viên. Hibernate cũng giúp tối ưu hóa hiệu suất của ứng dụng bằng cách tự động tối ưu hóa các câu truy vấn và cung cấp một số tính năng như caching dữ liệu.

Tuy nhiên, việc sử dụng Hibernate trong C# cũng có một số hạn chế, chẳng hạn như sự thiếu khả năng tùy biến và độ linh hoạt trong việc định nghĩa lớp đối tượng. Ngoài ra, những thay đổi trong cấu trúc cơ sở dữ liệu có thể gây ra khó khăn trong việc cập nhật mã lập trình.

Tóm lại, sử dụng Hibernate trong C# là một giải pháp hữu hiệu để giảm thiểu công sức và thời gian cần thiết cho việc lập trình và tối ưu hóa hiệu suất của ứng dụng. Tuy nhiên, lập trình viên cần lưu ý về những hạn chế của Hibernate để có thể áp dụng framework một cách hiệu quả trong dự án.

**Hình 1.5 NHIBERNATE Framework**

## Ant Design Blazor

Ant Design là một bộ UI component được phát triển bởi Ant Financial, một phần của tập đoàn Alibaba. Được phát hành dưới dạng open-source, Ant Design đã trở thành một trong những bộ UI component phổ biến nhất được sử dụng bởi các nhà phát triển frontend trên toàn thế giới.

Ant Design được thiết kế để giúp các nhà phát triển tạo ra các ứng dụng web và mobile với giao diện đẹp mắt, tương tác đơn giản và trải nghiệm người dùng tốt. Bộ component này cung cấp các thành phần cơ bản như button, input, select, table và form, cũng như các thành phần nâng cao hơn như dropdown, notification, modal và menu. Tất cả các thành phần đều được thiết kế để có thể tùy chỉnh linh hoạt theo nhu cầu của người dùng.

Với Ant Design, người dùng có thể tạo ra các ứng dụng web và mobile chất lượng cao với chi phí phát triển thấp hơn so với việc phát triển từ đầu. Bộ component này được phát hành dưới dạng open-source, cho phép các nhà phát triển truy cập mã nguồn và tùy chỉnh theo ý muốn của mình. Điều này giúp các nhà phát triển có thể tiết kiệm thời gian và chi phí phát triển, đồng thời cũng tạo ra cộng đồng lớn các nhà phát triển đang sử dụng và đóng góp cho Ant Design.

Tuy nhiên, như bất kỳ công nghệ nào khác, Ant Design cũng có những hạn chế. Một số thành phần có thể không phù hợp với yêu cầu cụ thể của người dùng và phải được tùy chỉnh lại để phù hợp. Ngoài ra, do sự phổ biến của Ant Design, một số ứng dụng có thể trông giống nhau và khó phân biệt với nhau. Tuy nhiên, các nhà phát triển có thể tùy chỉnh Ant Design để tạo ra các ứng dụng độc đáo và phù hợp với yêu cầu của mình.

Tóm lại, Ant Design là một bộ UI component mạnh mẽ và phổ biến được sử dụng bởi các nhà phát triển trên toàn thế giới. Nó giúp các nhà phát triển tạo ra các ứng dụng web và mobile chất lượng cao với chi phí phát triển thấp hơn, tuy nhiên cũng có những hạn chế cần được cân nh



Hình 1.6 Ant Design Blazor

## CHƯƠNG 2: PHÂN TÍCH YÊU CẦU

## 2.1 Lập kế hoạch và khám phá yêu cầu

## 2.1 Giới thiệu

### **2.1.1 Mục đích**

Mục đích của chương này là trình bày quá trình lập kế hoạch và khám phá yêu cầu cho dự án phần mềm, giúp đảm bảo rằng các yêu cầu của khách hàng được hiểu đúng và đầy đủ, từ đó phát triển một kế hoạch chi tiết để đáp ứng các yêu cầu này. Trong phần này, chúng tôi sẽ trình bày chi tiết về quá trình lập kế hoạch, bao gồm cách thức tiếp cận để xác định mục tiêu, tài nguyên và lộ trình cần thiết để thực hiện dự án phần mềm. Ngoài ra, phần này cũng sẽ giới thiệu về quá trình khám phá yêu cầu, bao gồm phương pháp tiếp cận để hiểu rõ yêu cầu của khách hàng và đưa ra các giải pháp phù hợp để thực hiện dự án.

### **2.1.2 Phạm vi**

* Đăng nhập, đăng xuất: cung cấp cho người dùng khả năng truy cập vào hệ thống và đăng xuất khi không sử dụng nữa.
* Quản lý danh sách học viên: cho phép thêm, sửa, xóa và tìm kiếm thông tin của các học viên trong hệ thống, bao gồm các thông tin như tên, địa chỉ, số điện thoại, email, khoa, ngành học và các thông tin khác liên quan.
* Quản lý danh sách nhà khoa học: cho phép thêm, sửa, xóa và tìm kiếm thông tin của các nhà khoa học trong hệ thống, bao gồm các thông tin như tên, địa chỉ, số điện thoại, email, đơn vị công tác và các thông tin khác liên quan.
* Quản lý hội đồng đánh giá: cho phép thêm, sửa, xóa và tìm kiếm thông tin về các hội đồng đánh giá trong hệ thống, bao gồm các thông tin như tên, thành viên, thời gian và địa điểm tổ chức.
* Quản lý đợt bảo vệ: cho phép thêm, sửa, xóa và tìm kiếm thông tin về các đợt bảo vệ trong hệ thống, bao gồm các thông tin như tên, thời gian, địa điểm và các thông tin khác liên quan.
* Xuất ra file đơn đăng kí cho từng học viên: cho phép xuất thông tin đăng kí của từng học viên thành một file đơn đăng kí để in hoặc lưu trữ.
* Lấy danh sách học viên đăng kí và thống kê các thông tin: cho phép lấy danh sách các học viên đăng kí và thống kê các thông tin liên quan, bao gồm số lượng học viên đăng kí, số lượng học viên đăng kí theo khoa và ngành học, số lượng học viên đăng kí theo đợt bảo vệ và các thông tin khác liên quan.

## 2.2. Mô tả chung

## 2.2.1 bối cảnh phát triển hệ thống phần mềm

### **2.2.1 Các chức năng thực hiện**

Chức năng được phát triển:

* **Đăng nhập**: cho phép người dùng truy cập vào hệ thống với tài khoản và mật khẩu tương ứng. Các tài khoản có các mức quyền hạn khác nhau để đảm bảo bảo mật và kiểm soát quyền truy cập vào các chức năng trong hệ thống.Đ**ăng xuất**: thực hiện đăng xuất khỏi ứng dụng.
* **Đăng xuất**: cho phép người dùng đăng xuất khỏi hệ thống và kết thúc phiên làm việc.
* **Quản lý danh sách học viên** cho phép lưu trữ, tìm kiếm, thêm, cập nhật và xóa thông tin học viên. Hỗ trợ nhập danh sách học viên từ file Excel.
* **Quản lý các nhà khoa học**: cho phép lưu trữ, tìm kiếm, thêm, cập nhật và xóa thông tin về các nhà khoa học.
* **Quản lý hội đồng đánh giá**: cho phép lưu trữ, tìm kiếm, thêm, cập nhật và xóa thông tin về hội đồng đánh giá.
* **Quản lý đợt bảo vệ:** cho phép lưu trữ, tìm kiếm, thêm, cập nhật và xóa thông tin về các đợt bảo vệ.
* **Xuất file đơn đăng kí cho học viên:** xuất file đơn đăng kí cho học viên.
* **Thống kê thông tin:** cho phép thống kê các thông tin về danh sách học viên, nhà khoa học và các hội đồng đánh giá.
* **2.2.2 Các lớp người dùng**
* **Ban chủ nhiệm khoa: được phép xem danh sách học viên của khoa mình, thêm nhà khoa học vào hội đồng của khoa và lấy danh sách học viên đăng kí theo khoa.**
* **Trợ lý sau đại học: được phép xem danh sách học viên của tất cả các khoa và thêm nhà khoa học vào hội đồng.**
* **Phòng sau đại học: có quyền của trợ lý sau đại học và được phép nhập danh sách học viên, chọn hội đồng 5 người, xuất file đăng kí cho từng học viên và lấy danh sách học viên đăng kí của tất cả các khoa.**
* **Admin: có tất cả các quyền của phòng sau đại học.**

### **Môi trường vận hành**

Hệ thống được thiết kế để có thể hoạt động trên nhiều phiên bản hệ điều hành khác nhau, bao gồm Windows 7, 8, 10, 11, macOS và Linux. Ngoài ra, hệ thống cũng tương thích với nhiều trình duyệt phổ biến như Chrome, Firefox, Edge, Safari, CocCoc và Opera.

# **2.3.** Xác định tập yêu cầu

### **2.3.1 Các yêu cầu chức năng**

1. chức năng đăng nhập, đăng xuất

* Giới thiệu:
* Chức năng thực hiện việc cho phép đăng nhập, đăng xuất vào hệ thống với tài khoản đã cấp.
* Quản lí từng phiên đăng nhập của các tài khoản.
* Thông tin lưu trữ gồm: Tên đăng nhập, mật khẩu, role của tài khoản đăng nhập.
* Đầu vào:
* Tài khoản và mật khẩu.
* Xử lí:
* Người sử dụng nhập tài khoản và mật khẩu để đăng nhập.
* Kiểm tra tài khoản mật khẩu xem có chính xác không, nếu đúng thì lưu phiên đăng nhập và cho phép đăng nhập, hiển thị chức năng, thông tin tương ứng với quyền hạn. Nếu sai hiển thị thông báo sai tài khoản mật khẩu.
* Sau khi đăng nhập, người sử dụng chọn đăng xuất để kết thúc phiên sử dụng.
* Đầu ra:
  + Đăng nhập thành công vào hệ thống.
* Lỗi:
  + Khi tài thông tin khoản hoặc mật khẩu không xác thực được.

1. Quản lí danh sách học viên

* Giới thiệu:
* Danh sách học viên bảo vệ luận văn được nhập từ file excel và lưu trữ trong hệ thống.
* Đầu vào:
* Danh sách học viên được lưu trong hệ thống.
* Chức năng người dùng chọn như tìm kiếm, xem chi tiết, cập nhật thông tin, xóa học viên, đưa vào danh sách bảo vệ luận văn.
* Xử lí:
* Hệ thống lấy thông tin của danh sách học viên để đưa ra kết quả phù hợp với các chức năng mà người dùng chọn.
* Đầu ra:
  + Thông tin học viên theo chức năng mà người dùng lựa chọn.

1. Quản lí các nhà khoa học

* Giới thiệu:
* Các nhà khoa học sẽ được người dùng nhập tay trên hệ thống.
* Các nhà khoa học sẽ có 2 loại: nhà khoa học trong trường và nhà khoa học ngoài trường.
* Đầu vào:
* Danh sách nhà khoa học được lưu trong cơ sở dữ liệu của hệ thống.
* Chức năng người dùng chọn như tìm kiếm, xem chi tiết, cập nhật thông tin, xóa học viên, đưa vào hội đồng bảo vệ luận văn.
* Xử lí:
* Hệ thống lấy thông tin của các nhà khoa học để đưa ra kết quả phù hợp với các chức năng mà người dùng chọn.
* Đầu ra:
  + Thông tin nhà khoa học theo chức năng mà người dùng lựa chọn.

1. Quản lý hội đồng đánh giá

* Giới thiệu:
* Hội đồng đánh giá sẽ có 7 người do khoa chọn sau đó phòng sau đại học sẽ chọn lấy 5 người trong 7 người này.
* Hội đồng đánh giá gồm có: 1 chủ tịch, 3 phản biện, 1 thư ký, 2 nhà khoa học (một nhà khoa học thuộc cơ sở đào tạo, một nhà khoa học ngoài cơ sở đào tạo).
* Đầu vào:
* Danh sách hội đồng đánh giá được lưu trong cơ sở dữ liệu của hệ thống.
* Chức năng người dùng chọn như lập hội đồng, tìm kiếm, xem chi tiết, cập nhật thông tin hội đồng, xóa hội đồng.
* Xử lí:
* Hệ thống lấy thông tin của hội đồng để đưa ra kết quả phù hợp với các chức năng mà người dùng chọn.
* Đầu ra:
  + Thông tin hội đồng đánh giá theo chức năng mà người dùng lựa chọn.

1. Quản lý đợt bảo vệ

* Giới thiệu:
* Mỗi năm sẽ có 2 đợt bảo vệ.
* Danh sách học viên bảo vệ được lấy từ đầu năm.
* Đầu vào:
* Đợt bảo vệ được lưu trong cơ sở dữ liệu của hệ thống.
* Chức năng người dùng chọn như tìm kiếm, xem chi tiết, cập nhật đợt bảo vệ.
* Xử lí:
* Hệ thống lấy thông tin đợt bảo vệ để đưa ra kết quả phù hợp với các chức năng mà người dùng chọn.
* Đầu ra:
  + Thông tin đợt bảo vệ theo chức năng mà người dùng lựa chọn.

1. Xuất file đơn đăng kí cho học viên

* Giới thiệu:
* Xuất file đơn đăng kí bảo vệ luận văn Thạc sĩ cho từng học viên.
* Đầu vào:
* Các thông tin cần thiết của học viên dùng để thêm vào file đơn đăng ký.
* Xử lí:
* Hệ thống lấy thông tin học viên nhập để xuất ra file đơn đăng kí cho học viên.
* Đầu ra:
  + File đơn đăng kí cho học viên.

1. Thống kê thông tin

* Giới thiệu:
* Thống kê số lượng xuất hiện của các nhà khoa học trong các hội đồng đánh giá, thống kê số lượng học viên tham gia bảo vệ đợt 1, đợt 2­­­.
* Đầu vào:
* Thông tin dữ liệu của hệ thống.
* Xử lý:
* Hệ thống tính toán và trả ra kết quả thống kê.
* Đầu ra:
  + Các kết quả thống kê.

### **Các yêu cầu phi chức năng**

* + Yêu cầu về hoạt động

Hệ thống cần hoạt động 24/7 và đảm bảo thời gian chết không quá 48 giờ trong một năm.

Hệ thống cần có khả năng tự động phục hồi khi gặp sự cố để giảm thiểu thời gian chết và giảm thiểu tác động đến người dùng.

* + Yêu cầu về hiệu suất

Hệ thống cần đáp ứng được số lượng người dùng truy cập đồng thời lên tới 150 người, đảm bảo tốc độ truy cập và thời gian phản hồi nhanh chóng.

Hệ thống cần được thiết kế để có thể nâng cấp và mở rộng trong tương lai để đáp ứng nhu cầu sử dụng ngày càng tăng.

* + Yêu cầu về bảo mật

Hệ thống cần có các biện pháp bảo mật như phân quyền truy cập và quản lý tài khoản để giảm thiểu rủi ro về an ninh thông tin.

Các tính năng và dữ liệu trên hệ thống chỉ được sử dụng và truy cập trong phạm vi trường đại học sư phạm Hà Nội.

* + Yêu cầu về văn hóa chính trị

Hệ thống cần đảm bảo tuân thủ các quy định về văn hóa chính trị và pháp luật của Việt Nam.

* + Yêu cầu về giao diện.

Giao diện của hệ thống cần được thiết kế thân thiện, dễ sử dụng và hiển thị đầy đủ các chức năng để giúp người dùng tương tác với hệ thống một cách thuận tiện và nhanh chóng.

* + Yêu cầu về tính khả dụng

Hệ thống cần đáp ứng được trên nhiều trình duyệt và nền tảng khác nhau, bao gồm Chrome, FireFox, Edge, Safari, CocCoc và Opera trên các hệ điều hành Window, macos, Linux.

Hệ thống cần có khả năng xử lý các lỗi phát sinh và đảm bảo tính khả dụng cao để giảm thiểu thời gian chết và giảm thiểu tác động đến người dùng.

* + Yêu cầu về tài nguyên ứng dụng

Hệ thống yêu cầu thiết bị có tối thiểu 2GB RAM để đảm bảo khả năng xử lý dữ liệu và tương tác người dùng một cách trơn tru.

Hệ thống cần có dung lượng ổ đĩa trống tối thiểu 2GB để lưu trữ dữ liệu và các tệp đính kèm liên quan đến hoạt động của người dùng.

* + Yêu cầu về độ tin cậy

Hệ thống phải đảm bảo rằng không xảy ra các trường hợp lưu sai dữ liệu trong quá trình thao tác và xử lý dữ liệu của người dùng. Nếu có lỗi xảy ra, hệ thống phải tự động khôi phục lại trạng thái trước đó để tránh làm sai lệch dữ liệu và gây thiệt hại cho người dùng và cơ quan sử dụng hệ thống.

Hệ thống cũng cần đảm bảo tính khả dụng cao, tức là hệ thống sẽ hoạt động liên tục 24/7 để đáp ứng nhu cầu sử dụng của người dùng. Thời gian hệ thống bị sập phải ít hơn 48 giờ/năm để đảm bảo sự liên tục trong hoạt động của hệ thống.

Hệ thống cần có khả năng chịu tải cao và đáp ứng tối đa 150 người dùng truy cập đồng thời vào cơ sở dữ liệu trung tâm ở bất cứ thời điểm nào để đảm bảo khả năng sử dụng của người dùng không bị gián đoạn.

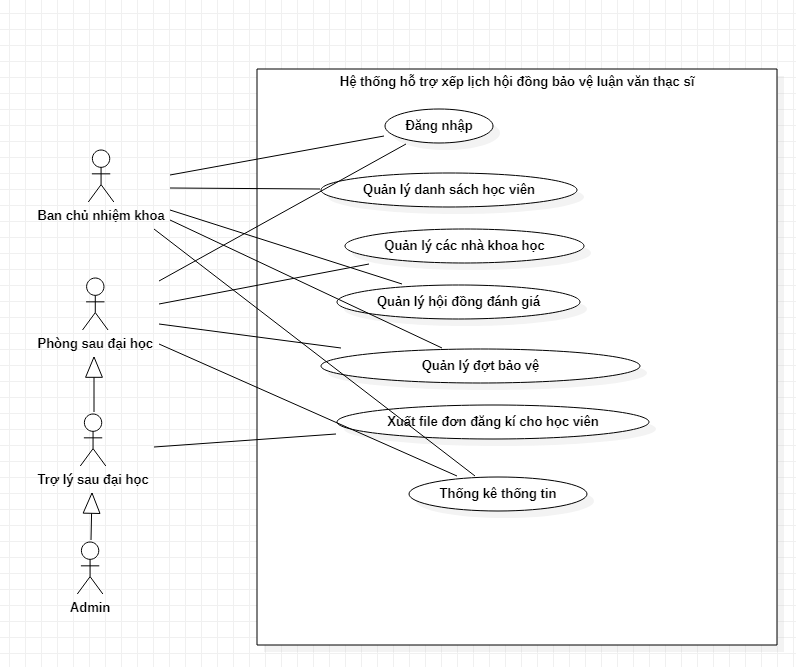
\*Yêu cầu về bảo mật

Hệ thống cần có phân quyền truy cập để chỉ cho phép những người dùng được cấp quyền sử dụng hệ thống và các tính năng liên quan. Hệ thống chỉ sử dụng trong phạm vi của trường Đại học Sư phạm Hà Nội và không được sử dụng cho mục đích khác.

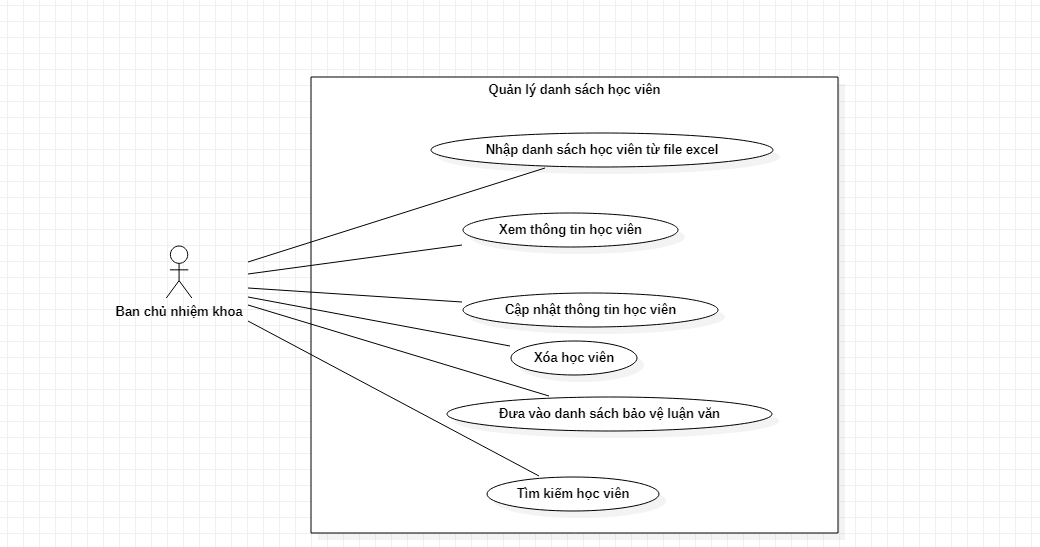
Các tài khoản được sử dụng để truy cập hệ thống cần được phân quyền và chỉ có thể sử dụng các tính năng tương ứng với mức quyền hạn của mình để đảm bảo an toàn và bảo mật thông tin của người dùng.

## 2.2 Phân tích yêu cầu

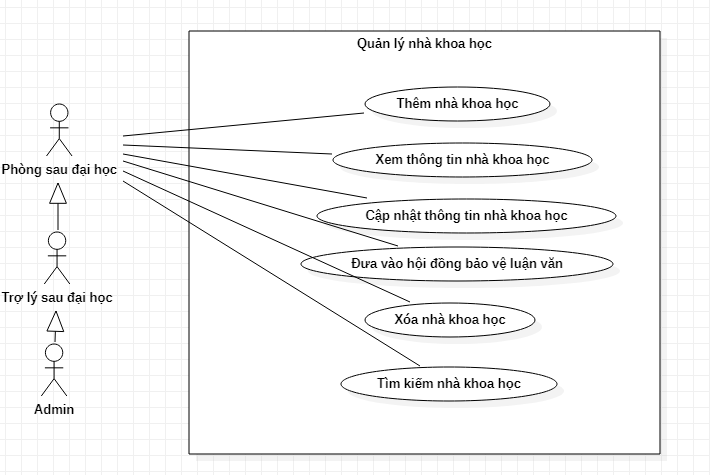
### **2.2.1 Mô hình hóa ca sử dụng**



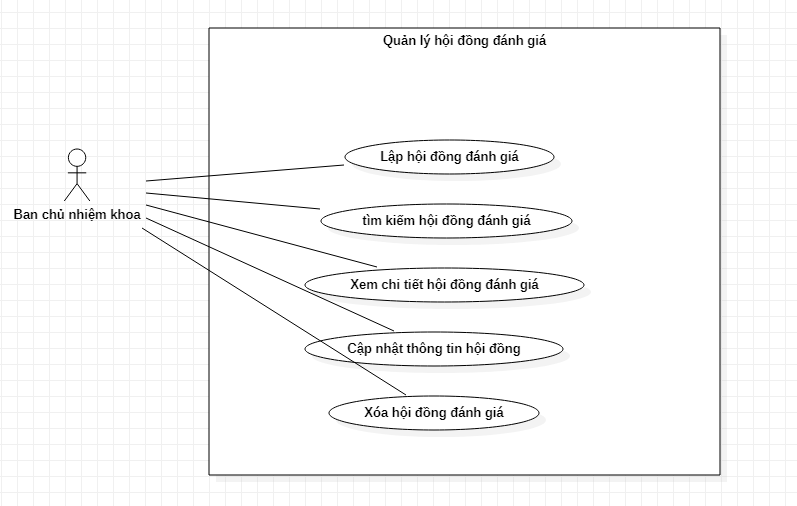
Hình 2.1 - Biểu đồ ca sử dụng ở mức tổng quan



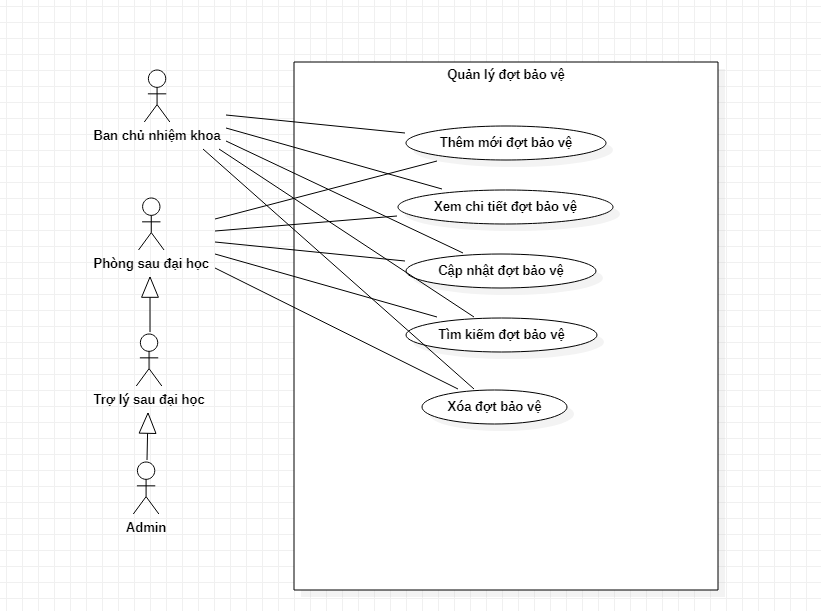
Hình 2.2 - Biểu đồ ca sử dụng Quản lý danh sách học viên



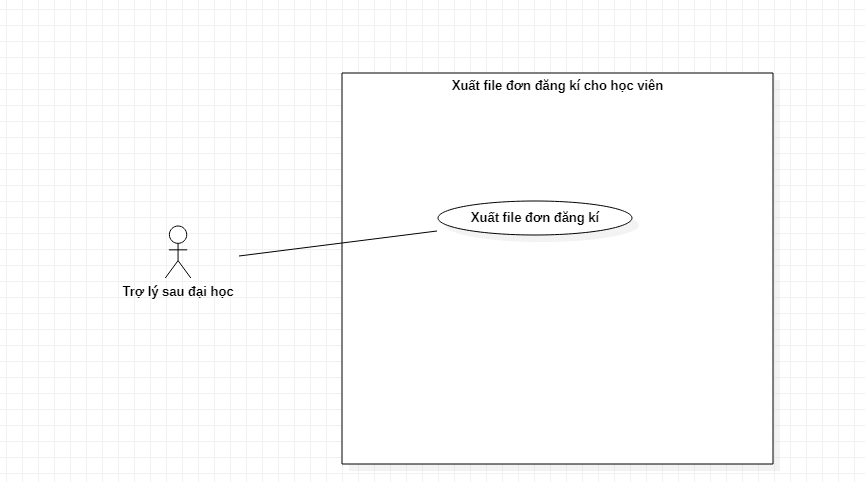
Hình 2.3 - Biểu đồ ca sử dụng Quản lý nhà khoa học



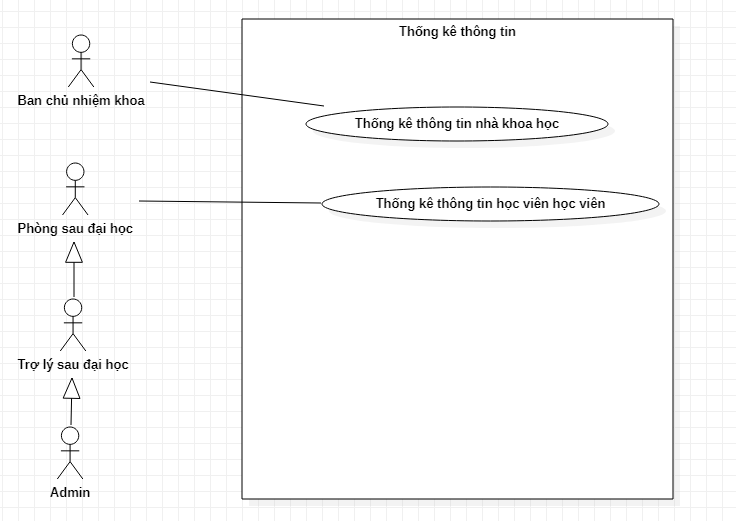
Hình 2.4 - Biểu đồ ca sử dụng Quản lý hội đồng đánh giá



Hình 2.5 - Biểu đồ ca sử dụng Quản lý đợt bảo vệ

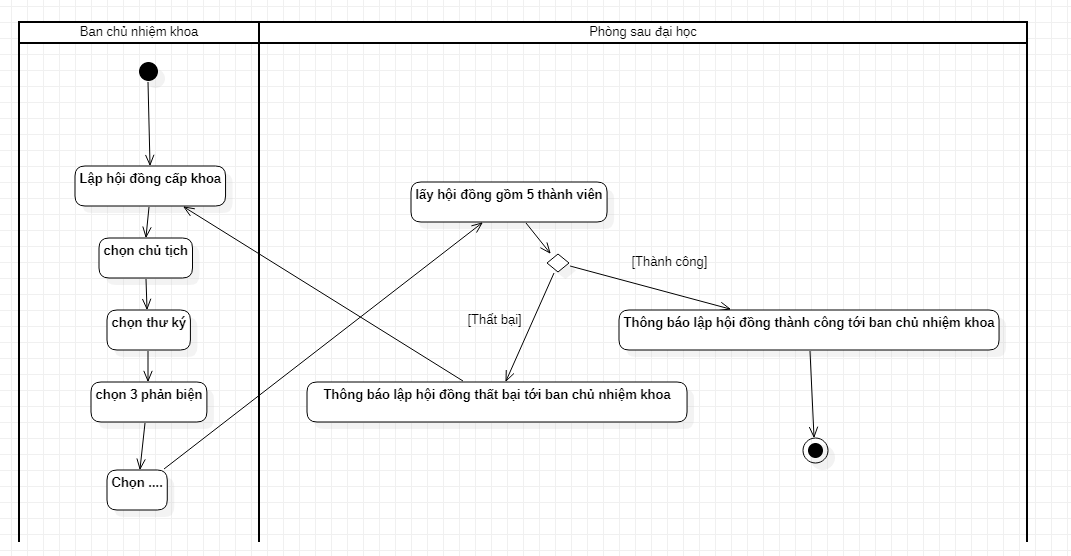


Hình 2.6 - Biểu đồ ca sử dụng xuất đơn đăng kí cho học viên



Hình 2.7 - Biểu đồ ca sử dụng thống kê thông tin

### **2.2.2 Mô hình hóa các tiến trình nghiệm vụ chính của hệ thống sử dụng biểu đồ hoạt động**



Hình 2.8 - Biểu đồ hoạt động thể hiện quá trình lập hội đồng bảo vệ luận văn thạc sĩ

### **Thiết kế tầng lưu trữ dữ liệu**

* + Chọn hệ quản trị cơ sở dữ liệu SQL server vì những ưu điểm sau:
    - Hỗ trợ tốt cho C# .Net Core.
    - Bảo mật cao.
    - Có thể cài đặt nhiều phiên bản MS SQL cho một máy chủ.
    - Duy trì riêng biệt các môi trường (sản xuất, phát triển và thử nghiệm).
    - Tách biệt những đặc quyền bảo mật.
  + Thiết kế cấu trúc các bảng dữ liệu lưu trữ trong hệ quản trị cơ sở dữ liệu:

Bảng HocVien

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên cột | Kiểu dữ liệu | Kích Thước | Khóa chính | Khóa Ngoại | Tự động tăng | Cho phép  null | Mô tả |
| 1 | Id | char | 36 | X |  |  |  | Id gồm 36 kí tự và là duy nhất. |
| 2 | Code | char | 20 |  |  | x |  | Mã học viên. |
| 3 | Name | varchar | 100 |  |  |  | x | Tên học viên. |
| 4 | DateOfBirth | datetime |  |  |  |  | x | Ngày sinh học viên. |
| 5 | Email | varchar | 100 |  |  |  |  | Email của học viên. |
| 6 | PhoneNumber | varchar | 20 |  |  |  |  | Số điện thoại học viên. |
| 7 | TopicName | varchar | 255 |  |  |  | x |  |
| 8 | InstructorOne | Varchar | 100 |  |  |  |  | Người hướng dẫn thứ nhất. |
| 9 | OnstructorTwo | Varchar | 100 |  |  |  |  | Người hướng dẫn thứ hai. |

Bảng HocVien

Bảng NhaKhoaHoc

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên cột | Kiểu dữ liệu | Kích Thước | Khóa chính | Khóa Ngoại | Tự động tăng | Cho phép  null | Mô tả |
| 1 | Id | char | 36 | X |  |  |  | Id gồm 36 kí tự và là duy nhất. |
| 2 | Code | char | 20 |  |  | x |  | Mã nhà khoa học. |
| 3 | Name | varchar | 100 |  |  |  | x | Tên nhà khoa học. |
| 4 | DateOfBirth | datetime |  |  |  |  | x | Ngày sinh nhà khoa học. |
| 5 | Email | varchar | 100 |  |  |  |  | Email của nhà khoa học. |
| 6 | PhoneNumber | varchar | 20 |  |  |  |  | Số điện thoại của nhà khoa học. |
| 7 | AcademicRank | varchar | 255 |  |  |  |  | Học hàm của nhà khoa học. |
| 8 | Degree | Varchar | 255 |  |  |  |  | Học vị của nhà khoa học. |
| 9 | CV | Varchar | 255 |  |  |  |  | CV của nhà khoa học. |

Bảng NhaKhoaHoc

Bảng TaiKhoan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên cột | Kiểu dữ liệu | Kích Thước | Khóa chính | Khóa Ngoại | Tự động tăng | Cho phép  null | Mô tả |
| 1 | Id | char | 36 | X |  |  |  | Id gồm 36 kí tự và là duy nhất. |
| 2 | UserName | Varchar | 255 |  |  | x |  | Tên tài khoản đăng nhập. |
| 3 | PassWord | varchar | 255 |  |  |  | x | Mật khẩu tài khoản đăng nhập. |
| 4 | Role | Int |  |  |  |  | x | Quyền hạn của tài khoản. |

Bảng TaiKhoan