**TRƯỜNG ĐẠI HỌC LÂM NGHIỆP**

KHOA KINH TẾ VÀ QUẢN TRỊ KINH DOANH

**===o0o===**

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**MÔN: LẬP TRÌNH C#**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ THƯ VIỆN**



**Giảng viên hướng dẫn: Ths. Trần Xuân Hòa**

**Sinh viên thực hiện:**

**Họ và Tên: Nguyễn Khả Đăng**

**Mã sinh viên: 2041070501**

**Lớp: K65 - HTTT**

**Hà Nội, 2020**

**MỤC LỤC**

LỜI CẢM ƠN

[CHƯƠNG I: TỔNG QUAN ĐỀ TÀI NGHIÊN CỨU 1](#_Toc119018903)

[1.1. Lý do, mục đích chọn đề tài 1](#_Toc119018904)

[1.2. Mục tiêu và phạm vi nghiên cứu 1](#_Toc119018905)

[1.2.1Mục tiêu 1](#_Toc119018906)

[1.2.2Phạm vi nghiên cứu 1](#_Toc119018907)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 2](#_Toc119018908)

[2.1. Công nghệ .Net 2](#_Toc119018909)

[Định nghĩa .Net là gì? 2](#_Toc119018910)

[Một số ưu điểm nổi bật của .NET 2](#_Toc119018911)

[2.2 Ngôn ngữ C# 3](#_Toc119018912)

[2.2.1 Mục tiêu của việc phát triển C# 3](#_Toc119018913)

[2.2.2 Ứng dụng của C# 4](#_Toc119018914)

[2.2.3 Đặc điểm của ngôn ngữ C# 4](#_Toc119018915)

[2.2.4 Soạn thảo và biên dịch C# 6](#_Toc119018916)

[2.3 ADO.NET 8](#_Toc119018917)

[2.3.1 ADO.NET là gì? 8](#_Toc119018918)

[2.3.2 Những loại ứng dụng nào sử dụng ADO.NET 8](#_Toc119018919)

[2.3.3 Các thành phần của ADO.NET 8](#_Toc119018920)

[2.3.4 .NET Data Providers là gì? 9](#_Toc119018921)

[2.3.5 DataSet là gì? 9](#_Toc119018922)

[2.4 SQL Server 9](#_Toc119018923)

[2.4.1 Định nghĩa SQL sever là gì? 9](#_Toc119018924)

[2.4.2 Những thành phần của SQL Server là gì? 10](#_Toc119018925)

[2.4.3 Mục đích khi sử dụng SQL Server là gì? 10](#_Toc119018926)

[CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 11](#_Toc119018927)

[3.1 Thiết kế mô hình UML 11](#_Toc119018928)

[3.1.1 Danh sách các đối tượng sử dụng (Actor) 11](#_Toc119018929)

[3.1.2 Mô tả các use case 11](#_Toc119018930)

[3.1.3 Cơ sở dữ liệu 13](#_Toc119018931)

[3.1.4. Biểu đồ phân rã các UC 13](#_Toc119018932)

[3.1.5. Biểu đồ hoạt động 13](#_Toc119018933)

[3.1.6. Biểu đồ trình tự 13](#_Toc119018934)

[CHƯƠNG IV: LẬP TRÌNH VÀ KẾT QUẢ 14](#_Toc119018935)

[4.1. Các hàm chức năng 14](#_Toc119018936)

[4.2. Kết quả lập trình 14](#_Toc119018937)

[4.2.1 Giao diện đăng nhập 14](#_Toc119018938)

[4.2.2. Giao diện quên mật khẩu 14](#_Toc119018939)

[CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 15](#_Toc119018940)

[5.1. Kết luận 15](#_Toc119018941)

[5.2. Hướng phát triển 15](#_Toc119018942)

TÀI LIỆU THAM KHẢO

**LỜI CẢM ƠN**

Chúng em được gửi lời cảm ơn trân thành đến các thầy cô trong ngành công nghệ thông tin Trường đại học lâm nghiệp và đặc biệt là thầy **Trần Xuân Hòa** là người đã trực tiếp giảng dạy và hướng dẫn, giúp đỡ về kiến thức và phương pháp nghiên cứu để chúng em có thể hoàn thành đồ án kết thúc môn học Lập trình C# với đề tài “***Xây dựng hệ thống quản lý thư viện.***”.

Chúng em xin chân thành cảm ơn!

*Hà Nội, ngày 20 tháng 11 năm 2020*

**Sinh viên thực hiện**

**Nguyễn Khả Đăng**

# TỔNG QUAN ĐỀ TÀI NGHIÊN CỨU

## 1. Lý do, mục đích chọn đề tài

Hiện nay, thư viện Trường Đại học Lâm Nghiệp chưa có một hệ thống chuyên biệt nào để lưu trữ thông tin sách và mượn trả sách tại thư viện trường. Việc quản lý sinh viên mượn trả sách bằng thủ công còn nhiều bất cập như mất nhiều thời gian, sổ sách nhiều, dễ sai sót trong thống kê... Thư viện quản lý khoảng hàng trăm ngàn đầu sách và tạp chí, phục vụ cho sinh viên của trường học tập tham khảo. Sinh viên có thể mượn sách đọc thông qua thẻ sinh viên. Để phục vụ sinh viên nhanh, gọn và chính xác, thư viện cần tin học hóa công việc quản lý danh mục sách và quản lý sinh viên của mình.... Bên cạnh đó cùng với sự phát triển của công nghệ thông tin, như cầu úng dụng công nghệ vào trong quản lý ngày càng phát triển mạnh và đem lại rất nhiều lợi ích. Do đó chương trình quản lí thư viện ra đời sẽ giúp cho các thủ thư đỡ mất công tìm kiếm thủ công trong mớ tài liệu đồ sộ, giúp việc quản lí sách và sinh viên dễ dàng,thuận tiện hơn. Từ những yêu cầu trên cùng với sự góp ý của các thầy cô giáo, em quyết định chọn đề tài là “xây dụng phần mềm quản lý thư viện trường Đại học Lâm Nghiệp”.

## 1.2 Mục tiêu và phạm vi nghiên cứu

### 1.2.1 Mục tiêu

Xây dựng được phần mềm quản lý thư viện cho trường Đại học Lâm Nghiệp

### 1.2.2 Phạm vi nghiên cứu

Thư viện Trường Đại học Lâm Nghiệp (cơ sở Xuân Mai)

# CƠ SỞ LÝ THUYẾT

## Công nghệ .Net

### 2.1.1 Định nghĩa .Net là gì?

.NET Framework là một nền tảng lập trình và cũng là một nền tảng thực thi ứng dụng chủ yếu trên hệ điều hành Microsoft Windows được phát triển bởi Microsoft từ năm 2002, kết thúc phát triển vào năm 2019 ở phiên bản 4.8. Các chương trình được viết trên nền .NET Framework sẽ được triển khai trong môi trường phần mềm (ngược lại với môi trường phần cứng) được biết đến với tên Common Language Runtime (CLR). Môi trường phần mềm này là một máy ảo trong đó cung cấp các dịch vụ như an ninh phần mềm (security), quản lý bộ nhớ (memory management), và các xử lý lỗi ngoại lệ (exception handling).

.NET framework bao gồm tập các thư viện lập trình lớn, và những thư viện này hỗ trợ việc xây dựng các chương trình phần mềm như lập trình giao diện; truy cập, kết nối cơ sở dữ liệu; ứng dụng web; các giải thuật, cấu trúc dữ liệu; giao tiếp mạng... CLR cùng với bộ thư viện này là 2 thành phần chính của .NET framework.

.NET framework đơn giản hóa việc viết ứng dụng bằng cách cung cấp nhiều thành phần được thiết kế sẵn, người lập trình chỉ cần học cách sử dụng và tùy theo sự sáng tạo mà gắn kết các thành phần đó lại với nhau. Nhiều công cụ được tạo ra để hỗ trợ xây dựng ứng dụng .NET, và IDE (Integrated Developement Environment) được phát triển và hỗ trợ bởi chính Microsoft là Visual Studio.

### 2.1.2 Một số ưu điểm nổi bật của .NET

* .Net có khả năng tương thích với đạt đa số các mã thực hiện cũng như lưu trữ của đối tượng.
* .Net có thể tạo ra sự nhất quán dành cho mọi trải nghiệm của nhà phát triển với các ứng dụng khác nhau ví dụ như: App web, App windows,..
* .Net tạo ra được những ứng dụng đơn giản dựa trên Web-based, Form-based dựa trên .NET framework.
* .Net có khả năng xây dựng được toàn bộ thông tin liên lạc cũng như các tiêu chuẩn của ngành công nghiệp dựa vào những khả năng tích hợp của hầu hết các mã thuộc nền tảng .Net.
* .Net có thể cung cấp một môi trường thực thi với mục đích giải quyết các rắc rối, xung đột liên quan đến hiệu suất. Từ đó, thúc đẩy cho quá trình phát triển và triển khai cho mã an toàn.
* Ngoài ra, .Net còn sở hữu vô số các điểm cộng khác như: rất đáng tin cậy nhờ có tính bảo mật cao, chi phí sẽ được giảm khi triển khai ứng dụng, ngôn ngữ sử dụng đa nền tảng,...

## 2.2 Ngôn ngữ C#

C# là một ngôn ngữ lập trình hướng đối tượng đa năng, mạnh mẽ được phát triển bởi Microsoft, C# là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

C# được thiết kế chủ yếu bởi Anders Hejlsberg kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, J++, WFC. Phiên bản gần đây nhất là 9.0, được phát hành vào năm 2020 cùng với Visual Studio 2019 phiên bản 16.8.[14]

### 2.2.1 Mục tiêu của việc phát triển C#

Tiêu chuẩn ECMA liệt kê các mục tiêu của việc thiết kế ngôn ngữ C#[15]:

* Ngôn ngữ được dự định là một ngôn ngữ lập trình đơn giản, hiện đại, hướng đến nhiều mục đích sử dụng, và là một ngôn ngữ lập trình hướng đối tượng.
* Ngôn ngữ và việc triển khai đáp ứng các nguyên tắc của ngành kỹ thuật phần mềm như kiểm tra chặt chẽ kiểu dữ liệu, kiểm tra giới hạn mảng, phát hiện các trường hợp sử dụng các biến chưa có dữ liệu, và tự động thu gom rác. Tính mạnh mẽ, sự bền bỉ, và năng suất của việc lập trình là rất quan trọng đối với ngôn ngữ này.
* Ngôn ngữ sẽ được sử dụng để phát triển các thành phần của phần mềm theo hướng thích hợp cho việc triển khai trong các môi trường phân tán.
* Khả năng di chuyển (portability) là rất quan trọng, đặc biệt là đối với những lập trình viên đã quen với C và C++.
* Hỗ trợ quốc tế hóa (i18n).
* Ngôn ngữ sẽ được thiết kế để phù hợp với việc viết các ứng dụng cho cả hai hệ thống: hosted và nhúng, từ các phần mềm quy mô lớn, đến các phần mềm chỉ có các chức năng đơn giản.
* Mặc dù các ứng dụng C# có tính kinh tế đối với các yêu cầu về bộ nhớ và chế độ xử lý, ngôn ngữ này không cạnh tranh trực tiếp về hiệu năng và kích thước đối với ngôn ngữ C hoặc assembly.

### 2.2.2 Ứng dụng của C#

* Phát triển web backend (ASP.NET MVC, ASP.NET core, Web API,Graph API,gPRC, Blazor sevver,Uno platform, Mono)
* Phát triển web front end (Blazor WebAssembly,Uno platform)
* Phát triển desktop app (Winform, WPF, UWP,WinUI, Mono, Uno, MAUI,Blazor desktop...)
* Phát triển game 2D, 3D đa nền tảng (Game engine:Unity, Monogame, Godot, Stride, CryEngine, Flax Engine, Evergine, NeoAxis, XNA ..)
* Phát triển thực tế ảo (VR), thực tế tăng cường(AR), thực tế hỗn hợp (MR) (HoloLens,Unity,CryEngine, Oculus quest..)
* Phát triển ứng dụng đồ họa 2D,3D đa nền tảng (2D: SkiaSharp, ImageSharp...; 3D: OpenTK, SharpDX, SharpVulkan, Vulkan.NET, Veldrid, Silk.NET, Helix Toolkit, Aspose..)
* Phát triển mobile app, IOS native, Android native (Xamarin,MAUI,Uno platform)
* Phát triển đám mây (Azure,AWS,Google Cloud...)
* Học máy và trí tuệ nhân tạo (ML.Net, TensorFlow, csiSharp..)
* Data science, bigdata (csiSharp, Apache Spark)
* Blockchain (NEO, Stratis)
* Microservices and containers
* Internet of thing (IoT,5G)
* Hệ thống nhúng (Raspberry pi, PLC)

### 2.2.3 Đặc điểm của ngôn ngữ C#

* C# là ngôn ngữ đơn giản, mạnh mẽ
  + C# được dựng trên nền tảng C++ và Java, ảnh hưởng bởi Delphi, VisualBasic nên ngôn ngữ C# được thừa hưởng các ưu điểm vào loại bỏ các yếu điểm của các ngôn ngữ trên, vì vậy nó khá đơn giản, đồng thời loại bỏ các cú pháp dư thừa và thêm vào đó các cú pháp cải tiến hơn
  + C# là ngôn ngữ lập trình bậc cao, đa nền tảng vì vậy nó dễ dàng tiếp cận và phù hợp cho người mới bắt đầu học, ví dụ câu lệnh kinh điển dành cho người mới bắt đầu học là in ra dòng chữ "Hello world", với C# ta chỉ cần 1 câu lệnh: System.Console.WriteLine("Hello world");
* C# là ngôn ngữ đa năng và hiện đại
  + C# phù hợp cho việc phát triển trong thời đại 4.0, bao gồm việc phát triển web, mobile app, game, học máy và trí tuệ nhân tạo, phát triển đám mây, IoT, blockchain, microservices...
* C# là một ngôn ngữ lập trình hướng đối tượng đồng thời hỗ trợ lập trình chức năng
  + C# hỗ trợ mạnh mẽ cho phương pháp lập trình hướng đối tượng, ngoài ra C# còn hỗ trợ các phương pháp lập trình chức năng thông qua các biểu thức lamba, khớp mẫu, functions, các thuộc tính bất biến.
* C# là ngôn ngữ gõ tĩnh, định kiểu mạnh, hỗ trợ gõ động.
  + C# được gõ tĩnh nên nó mang đầy đủ các ưu việt của phương pháp gõ tĩnh như bảo đảm an toàn kiểu, tự động phân tích và nhận biết lỗi cú pháp ngay trong quá trình viết mã...
  + Ngoài ra khi sử dụng C# kết hợp với IDE Visual Studio, C# được hỗ trợ gợi ý code bởi Visual Studio IntelliCode sử dụng trí tuệ nhân tạo giúp cho việc viết code trở nên nhanh chóng và dễ dàng hơn
* C# là một ngôn ngữ ít từ khóa
  + C# có khoảng hơn 80 từ khóa
* C# là một trong các ngôn ngữ lập trình phổ biến và phát triển nhất
  + Theo TIOBE Index, tính đến tháng 10/2020, C# là ngôn ngữ phổ biến thứ 5 thế giới.
  + Theo PYPL, tính đến tháng 10/2020, C# là ngôn ngữ được cộng đồng quan tâm và chia sẻ nhiều thứ 4 thế giới.
  + Theo [./Https://github.com/ Github] (Kho lưu trữ mã nguồn lớn nhất thế giới), tính đến tháng 10/2020, C# là ngôn ngữ hoạt động nhiều thứ 4.
  + Cộng đồng phát triển và số người theo học ngôn ngữ C# tăng khônng ngừng theo mỗi năm.Theo ước tính 10/2020, cộng đồng phát triển C# là hơn 6 triệu người
* C# kết hợp chặt chẽ với nền tảng.NET - một khung nền tảng được đầu tư rất mạnh của Microsoft.
* Ngoài ra C# còn có những ưu điểm:
  + C# là ngôn ngữ lập trình mã nguồn mở,vì vậy C# là miễn phí với tất cả mọi người, đồng thời mọi người đều có thể cùng tham gia phát triển, đề xuất thiết kế ngôn ngữ C#
  + C# là ngôn ngữ đa nền tảng vì vậy có thể biên dịch trên nhiều nền tảng máy tính khác nhau (Windows, Linux, MacOS)
  + C# có hiệu suất cao và tốc độ thực thi nhanh do sử dụng trình biên dich trung gian (CLR), điểm cộng nữa là tốc độ phát triển phần mềm nhanh chóng so với đa số các ngôn ngữ hiện tại.
  + C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ. ngoài ra có thể viết C# bằng bất kỳ text editor nào khác như Visual Studio Code, Vim, Netbeam...
  + C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, song cũng được bổ sung các yếu tố mang tính hiện đại nên dễ dàng tiếp cận cho người mới học và học nhanh với C#.
  + C# có cộng đồng nhà phát triển vô cùng lớn mạnh.
  + C# được phát triển và cải tiến không ngừng với tần suất 1 phiên bản/ 1 năm, đáp ứng các mong muốn cải thiện, cải tiến cho phù hợp với nhu cầu công nghệ của các nhà phát triển.
  + C# có tài liệu tham khảo và hướng dẫn vô cùng phong phú và chất lượng, đồng thời có các buổi hội thảo giới thiệu tính năng mới và định hướng phát triển ngôn ngữ trong tương lai.
  + C# và.NET được đánh giá là có design tốt, vì vậy cú pháp và logic rất nhất quán, mã nguồn C# dễ đọc và mở rộng.
  + C# được thiết kế và phát triển bởi Microsoft nên rất được Microsoft quan tâm và hỗ trợ.

### 2.2.4 Soạn thảo và biên dịch C#

Có hai cách để soạn thảo, biên dịch và thực thi cho chương trình C# đó là:

* Sử dụng chương trình soạn thảo văn bản bất kỳ rồi sử dụng command line để biên dịch và thực thi
* Sử dụng môi trường phát triển tích hợp (IDE).

**Biên dịch chương trình C#**

* Ngôn ngữ C# biên dịch chương trình ra mã trung gian (MSIL).
* Sau đó thực hiện bằng cách thông dịch hoặc biên dịch trong từng máy ảo

tương ứng.

**Sử dụng chương trình soạn thảo văn bản**

* Dùng bất kỳ một trình soạn thảo văn bản để lập trình.
* Lưu tập tin dưới dạng \*.cs.
* Dùng command line để biên dịch và thực thi tập tin nguồn. Mẫu dòng lệnh biên

dịch là: csc.exe [/out: <file thực thi>] <file nguồn>

*Ví dụ: csc.exe /out:C:\QLSV.exe d:\QLSV.cs*

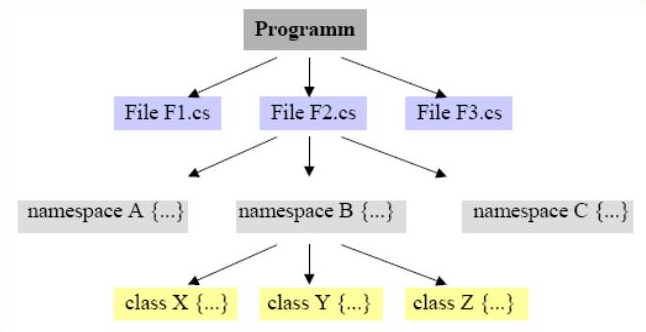
**Sử dụng IDE**

* Dùng bộ IDE của VS.NET
* Mở chương trình VS.NET đã được cài trên máy.
* Tạo mới một chương trình.
* Soạn thảo chương trình trong cửa sổ.
* Lưu tập tin.
* Biên dịch và chạy chương trình.

**Các loại ứng dụng C#**

* Chương trình Console (TUI)
* Giao tiếp với người dùng bằng bàn phím
* Không có giao diện đồ họa (GUI)
* Chương trình Windows Form
* Giao tiếp với người dùng bằng bàn phím và mouse
* Có giao diện đồ họa và xử lý sự kiện
* Chương trình Web Form
* Kết hợp với ASP .NET, C# đóng vài trò xử lý bên dưới (underlying code)
* Có giao diện đồ họa và xử lý sự kiện

**Cấu trúc chương trình C#**



## **2.3 ADO.NET**

### 2.3.1 ADO.NET là gì?

ADO là viết tắt của Microsoft ActiveX Data Objects. ADO.NET là một trong những công nghệ truy cập dữ liệu của Microsoft. Nó là một phần của .Net Framework được sử dụng để thiết lập kết nối giữa ứng dụng .NET và các nguồn dữ liệu.

Các nguồn dữ liệu có thể là SQL Server, Oracle, MySQL, XML, .... ADO.NET bao gồm một tập hợp các lớp có thể được sử dụng để kết nối, truy xuất, chèn và xóa dữ liệu khỏi nguồn dữ liệu. ADO.NET chủ yếu sử dụng System.Data.dll và System.Xml.dll.

### 2.3.2 Những loại ứng dụng nào sử dụng ADO.NET

ADO.NET có thể được sử dụng để phát triển bất kỳ loại ứng dụng .NET nào. Sau đây là một số ứng dụng .NET mà bạn có thể sử dụng công nghệ truy cập dữ liệu ADO.NET để tương tác với nguồn dữ liệu.

* ASP.NET Web Form Applications
* Windows Applications
* ASP.NET MVC Applications
* Console Applications
* ASP.NET Web API Applications

### 2.3.3 Các thành phần của ADO.NET

Các Compnents của ADO.NET được thiết kế để thao tác dữ liệu và truy cập nhanh vào dữ liệu. Connecton, Command, DataReader, DataAdapter, DataSet và DataView là các Components của ADO.NET được sử dụng để thực hiện các hoạt động cơ sở dữ liệu. ADO.NET có hai thành phần chính được sử dụng để truy cập và thao tác dữ liệu như sau:

* Data Provider
* DataSet

### 2.3.4 .NET Data Providers là gì?

Cơ sở dữ liệu không thể thực thi trực tiếp code C# của chúng ta, nó chỉ hiểu SQL. Vì vậy, nếu ứng dụng .NET muốn truy xuất dữ liệu hoặc thực hiện một số thao tác chèn, cập nhật, xóa từ ứng ứng vào CSDL thì cần:

1. Kết nối với CSDL.
2. Chuẩn bị câu lệnh SQL.
3. Thực hiện lệnh.
4. Lấy kết quả và hiển thị chúng trong ứng dụng.

### 2.3.5 DataSet là gì?

DataSet không dành riêng cho một Data Provider (nhà cung cấp) nào cả. Khi bạn kết nối với cơ sở dữ liệu, hãy thực thi lệnh và truy xuất dữ liệu vào ứng dụng .NET. Dữ liệu sau đó có thể được lưu trữ trong DataSet và hoạt động độc lập với CSDL. Vì vậy, nó được sử dụng để truy cập dữ liệu độc lập từ bất kỳ nguồn dữ liệu nào. DataSet chứa một tập hợp một hoặc nhiều đối tượng trong DataTable.

## 2.4 SQL Server

### 2.4.1 Định nghĩa SQL sever là gì?

SQL server hay còn được gọi là Microsoft SQL Server, nó từ viết tắt của MS SQL Server. Đây chính là một loại phần mềm đã được phát triển bởi Microsoft và nó được sử dụng để có thể dễ dàng lưu trữ cho những dữ liệu dựa theo tiêu chuẩn RDBMS.

Từ đó, người ta sẽ lưu trữ dữ liệu dựa vào tiêu chuẩn RDBMS và nó cũng là một trong những hệ quản trị cơ sở dữ liệu dạng quan hệ đối tượng.

SQL Server có khả năng cung cấp đầy đủ các công cụ cho việc quản lý từ giao diện GUI đến sử dụng ngôn ngữ cho việc truy vấn SQL. Điểm mạnh của SQL điểm mạnh của nó là có nhiều nền tảng được kết hợp cùng như: ASP.NET, C# để xây dựng Winform cũng chính nó có khả năng hoạt động độc lập.

Tuy nhiên, SQL Server thường đi kèm với việc thực hiện riêng các ngôn ngữ SQL, T-SQL,...

Cụ thể như sau:

* T-SQL là một trong những loại ngôn ngữ thuộc quyền sở hữu của Microsoft và được gọi với cái tên Transact-SQL. Nó thường cung cấp thêm rất nhiều cho các khả năng khai báo biến, thủ tục lưu trữ và xử lý ngoại lệ,...
* SQL Server Management Studio là một loại công cụ giao diện chính cho máy chủ cơ sở của chính dữ liệu SQL, thông thường thì nó hỗ trợ cho cả môi trường 64 bit và 32 bit.

### 2.4.2 Những thành phần của SQL Server là gì?

SQL Server đã trải qua hơn 20 năm phát triển và đã đề ra những version cụ thể khác nhau. Các mô hình Client - Server được chia làm 2 thành phần chính bao gồm:

Workstation: Nó được cài đặt trên các thiết bị vận hành để trở thành phần mềm tương tác với hệ thống máy chủ Server.

Server: Được cài đặt ở trên máy chủ chính, nó có thể là: SQL server, SQL Server Agent, SSIS, SSAS,...

Ngoài ra, bạn hoàn toàn có thể cài đặt nhiều phiên bản của SQL Server trên cùng một máy chủ và điều này sẽ giúp tiết kiệm được các chi phí mua Server để hoạt động và cần nhiều phiên bản khác nhau. Nó bảo mật và cũng được tách biệt hoàn toàn giúp cho hệ thống được an toàn hơn.

### 2.4.3 Mục đích khi sử dụng SQL Server là gì?

SQL Server thông thường được sử dụng cho mục đích lưu trữ dữ liệu. Ngoài ra, nó còn mang lại những tính năng làm việc giúp người dùng làm việc hiệu quả hơn như sau:

* Giúp người sử dụng có thể duy trì việc lưu trữ bền vững.
* Cho phép bạn tạo ra nhiều cơ sở dữ liệu hơn.
* Có khả năng phân tích dữ liệu bằng SSAS
* Nó có khả năng bảo mật cao
* Việc tạo ra được các báo cáo bằng SSRS — SQL Server Reporting Services sẽ được dễ dàng hơn.
* Các quá trình sẽ được thực hiện bằng SSIS — SQL Server Integration Services.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 3.1 Thiết kế mô hình UML

### 3.1.1 Danh sách các đối tượng sử dụng (Actor)

|  |  |  |
| --- | --- | --- |
| STT | Tên Actor | Mô tả |
| 1 | Tác giả | Người thực hiện đề tài. |
| 2 | Nhân viên | Là nhân viên của hệ thống có quyền đăng nhập vào hệ thống, có thể thêm đề tài, duyệt các đề tài đăng ký. |
| 3 | Người quản trị | Là nhân viên của hệ thống có quyền đăng nhập vào hệ thống, thực hiện tất cả các chức năng của hệ thống bao gồm: Quản lý đề tài, quản lý phê duyệt đề tài đăng ký, quản lý tài khoản trong hệ thống và phân quyền |

Bảng 1: Danh sách các đối tượng sử dụng (Actor)

### 3.1.2 Mô tả các use case

|  |  |  |
| --- | --- | --- |
| STT | Use case | Mô tả |
| 1 | Đăng nhập | Người dùng cần đăng nhập dùng tài khoản và mật khẩu của mình để có thể thực hiện các thao tác ứng với từng chức năng của mình |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |

Bảng 2: Danh sách các use case chính

## 3.1.3 Cơ sở dữ liệu

* **Bảng đề tài (Topic):**
* **Bảng tài khoản đăng nhâp (Account):**

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| ID | Int | Primary, NOT NULL |
| Username | Nvarchar | Tên đăng nhập |
| Password | Nvarchar | Mật khẩu đăng nhập |
| UserType | Nvarchar | Phân quyên đăng nhập |
| Email | Nvarchar | Email |

Bảng 4: Cơ sở dữ liệu đăng nhập

* **Bảng chờ phê duyệt đề tài đăng ký (Queue):**

### 3.1.4. Biểu đồ phân rã các UC

* **Biểu đồ UC hệ thống**

### 3.1.5. Biểu đồ hoạt động

* **Biểu đồ hoạt động đăng nhập**

### 3.1.6. Biểu đồ trình tự

* **Biểu đồ trình tự đăng nhập**
* **Biểu đồ trình tự đăng ký tài khoản**

# LẬP TRÌNH VÀ KẾT QUẢ

## 4.1. Các hàm chức năng

## 4.2. Kết quả lập trình

### 4.2.1 Giao diện đăng nhập

### 4.2.2. Giao diện quên mật khẩu

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

## 5.2. Hướng phát triển

TÀI LIỆU THAM KHẢO

[1]. ***.Net Framework*** ([.NET Framework – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/.NET_Framework))

[2]. **C#** ([C Sharp (ngôn ngữ lập trình) – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/C_Sharp_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)))

[3]. ***ADO.Net*** ([ADO.NET là gì? Các thành phần quan trọng của ADO .NET (freetuts.net)](https://freetuts.net/gioi-thieu-ve-adonet-5440.html))

[4]. ***SQL Server*** ([SQL Server là gì? Mục đích của việc sử dụng SQL Server (itnavi.com.vn)](https://itnavi.com.vn/blog/sql-server))