**TRƯỜNG ĐẠI HỌC LÂM NGHIỆP**

KHOA KINH TẾ VÀ QUẢN TRỊ KINH DOANH

**===o0o===**

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**MÔN: AN TOÀN BẢO MẬT THÔNG TIN**

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



**Giảng viên hướng dẫn: Mai Hà An**

**Sinh viên thực hiện:**

**Họ và Tên: Nguyễn Khả Đăng**

**Mã sinh viên: 2041070501**

**Lớp: K65 - HTTT**

**Hà Nội, 2022**

**MỤC LỤC**

[CHƯƠNG I. TỔNG QUAN ĐỀ TÀI NGHIÊN CỨU 1](#_Toc120128068)

[1.1 Lý do, mục đích chọn đề tài 1](#_Toc120128069)

[1.2 Mục tiêu và phạm vi nghiên cứu 1](#_Toc120128070)

[1.2.1 Mục tiêu 1](#_Toc120128071)

[1.2.2 Phạm vi nghiên cứu 1](#_Toc120128072)

[CHƯƠNG II. CƠ SỞ LÝ THUYẾT 2](#_Toc120128073)

[2.1. Công nghệ .Net 2](#_Toc120128074)

[2.1.1 Định nghĩa? 2](#_Toc120128075)

[2.1.2 Một số ưu điểm nổi bật của .NET 2](#_Toc120128076)

[2.2 Ngôn ngữ C# 3](#_Toc120128077)

[2.2.1 Định nghĩa 3](#_Toc120128078)

[2.2.2 Mục tiêu của việc phát triển C# 3](#_Toc120128079)

[2.2.3 Ứng dụng của C# 4](#_Toc120128080)

[2.2.4 Đặc điểm của ngôn ngữ C# 4](#_Toc120128081)

[2.2.5 Soạn thảo và biên dịch C# 6](#_Toc120128082)

[2.3 ADO.NET 8](#_Toc120128083)

[2.3.1 ADO.NET là gì? 8](#_Toc120128084)

[2.3.2 Những loại ứng dụng nào sử dụng ADO.NET 8](#_Toc120128085)

[2.3.3 Các thành phần của ADO.NET 9](#_Toc120128086)

[2.3.4 .NET Data Providers là gì? 9](#_Toc120128087)

[2.3.5 DataSet là gì? 9](#_Toc120128088)

[2.4 SQL Server 9](#_Toc120128089)

[2.4.1 Định nghĩa SQL sever là gì? 9](#_Toc120128090)

[2.4.2 Những thành phần của SQL Server là gì? 10](#_Toc120128091)

[2.4.3 Mục đích khi sử dụng SQL Server là gì? 10](#_Toc120128092)

[CHƯƠNG III. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 12](#_Toc120128093)

[3.1 Thiết kế mô hình UML 12](#_Toc120128094)

[3.1.1 Danh sách các đối tượng sử dụng (Actor) 12](#_Toc120128095)

[3.1.2 Mô tả các use case 12](#_Toc120128096)

[3.1.3 Cơ sở dữ liệu 14](#_Toc120128097)

[3.1.4. Biểu đồ phân rã các UC 16](#_Toc120128098)

[3.1.5. Biểu đồ hoạt động 19](#_Toc120128099)

[CHƯƠNG IV. LẬP TRÌNH VÀ KẾT QUẢ 21](#_Toc120128100)

[4.1. Các hàm chức năng 21](#_Toc120128101)

[4.1.1 Kết nối tới Sql 21](#_Toc120128102)

[4.1.2 Kiểm tra trùng lặp mã 21](#_Toc120128103)

[4.1.3 Kiểm tra số lượng sách 21](#_Toc120128104)

[4.1.4 Hiển thị data từ sql lên datagridview 22](#_Toc120128105)

[4.1.5 Insert 22](#_Toc120128106)

[4.1.6 Update 23](#_Toc120128107)

[4.1.7 Delete 24](#_Toc120128108)

[4.1.8 Search 24](#_Toc120128109)

[4.1.9 Export 25](#_Toc120128110)

[4.1.10 Import 25](#_Toc120128111)

[4.1.11 MD5\_algorithm 26](#_Toc120128112)

[4.2. Kết quả lập trình 27](#_Toc120128113)

[4.2.1 Form đăng nhập 27](#_Toc120128114)

[4.2.2. Form quên mật khẩu 27](#_Toc120128115)

[4.2.3. Form cho admin 27](#_Toc120128116)

[4.2.4. Form cho nhân viên 28](#_Toc120128117)

[4.2.5. Form quản lý mượn sách 28](#_Toc120128118)

[4.2.6. Form quản lý trả sách 29](#_Toc120128119)

[4.2.7. Form quản lý sách 29](#_Toc120128120)

[4.2.8. Form quản lý loại sách 30](#_Toc120128121)

[4.2.9. Form quản lý đọc giả 30](#_Toc120128122)

[4.2.10. Form quản lý chức vụ nhân viên 31](#_Toc120128123)

[4.2.11. Form quản lý nhân viên thư viện 31](#_Toc120128124)

[4.2.12. Form báo cáo thống kê 32](#_Toc120128125)

[4.2.13. Form quản lý tài khoản 32](#_Toc120128126)

[CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 33](#_Toc120128127)

[5.1. Kết luận 33](#_Toc120128128)

[5.1.1. Kết quả đạt được 33](#_Toc120128129)

[5.1.2. Hạn chế 33](#_Toc120128130)

[5.2. Hướng phát triển 33](#_Toc120128131)

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 **Mai Hà An** 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 An Toàn Bảo Mật Thông Tin 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**

# CHƯƠNG I. TỔNG QUAN ĐỀ TÀI NGHIÊN CỨU

## 1.1 Lý do, mục đích chọn đề tài

Hiện nay, thư viện Trường Đại học ABC 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. Đọc giả có thể mượn sách đọc thông qua thẻ sinh viên. Để phục vụ đọc giả nhanh, gọn và chính xác, thư viện cần tin học hóa công việc quản lý danh mục sách và quản lý đọc giả 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 đọc giả 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”.

## 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 ABC

### 1.2.2 Phạm vi nghiên cứu

Thư viện Trường ABC (cơ sở XM)

# CHƯƠNG II. CƠ SỞ LÝ THUYẾT

## Công nghệ .Net

### 2.1.1 Định nghĩa?

.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#

### 2.2.1 Định nghĩa

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.

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

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

* 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.

### 2.2.3 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)

### 2.2.4 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.

### 2.2.5 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*

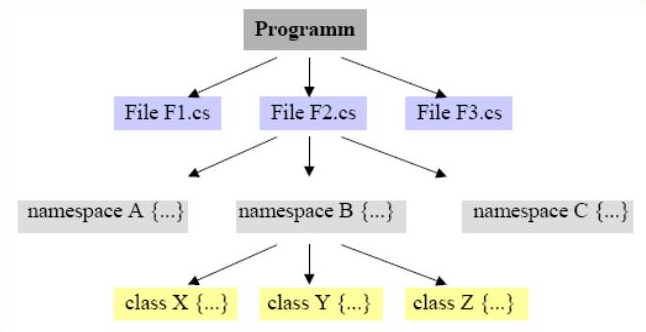
**Sử dụng IDE**

* 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.

# CHƯƠNG III. 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 quản lý đọc giả, quản lý sách, quản lý mượn trả, tìm kiếm thông tin, thống kê mượn trả. |
| 3 | Người quản trị (admin) | 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 đọc giả, quản lý sách, quản lý mượn trả, tìm kiếm thông tin, thống kê mượn trả, quản lý nhân viên. |

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 đăng nhập vào hệ thống |
| 2 | Quên mật khẩu | Cho phép người dùng đổi lại mật khẩu khi quên bằng cách xác nhập lại email đã đăng kí trên hệ thống |
| 3 | Quản lý người dùng | Cho phép admin thêm, sửa, xoá người dùng |
| 4 | Quản lý chức vụ | Cho phép admin thêm, sửa, xoá, tìm kiếm và xuất danh sách excel chức vụ trong thư viện |
| 5 | Quản lý nhân viên | Cho phép admin thêm, sửa, xoá, tìm kiếm và xuất danh sách excel nhân viên trong thư viện |
| 6 | Quản lý đọc giả | Cho phép người dùng thêm, sửa, xoá, tìm kiếm và xuất danh sách excel đọc giả ra vào, mượn sách trong thư viện |
| 7 | Quản lý loại sách | Cho phép người dùng thêm, sửa, xoá, tìm kiếm, xuất danh sách ra excel, chèn danh sách excel vào danh sách các loại sách |
| 8 | Quản lý sách | Cho phép người dùng thêm, sửa, xoá, tìm kiếm, xuất danh sách ra excel, chèn danh sách excel vào danh sách sách có trong thư viện |
| 9 | Quản lý mượn sách | Cho phép người dùng thêm, sửa, tìm kiếm, xuất danh sách ra excel, chèn danh sách excel vào danh sách mượn sách |
| 10 | Quản lý trả sách | Cho phép người dùng thêm, sửa, tìm kiếm, xuất danh sách ra excel, chèn danh sách excel vào danh sách trả sách |
| 11 | Thống kê | Cho người dùng biết được sách nào được mượn nhiều nhất, còn đọc giả nào chưa trả sách, đọc giả nào mượn quá hạn |

Bảng 2: Danh sách các use case chính

### 3.1.3 Cơ sở dữ liệu

1. Bảng Users

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| user\_\_id | Int | Khoá chính |
| user\_name | varchar | Tên đăng nhập |
| password | varchar | Mật khẩu đăng nhập |
| email | Nvarchar | Email |

1. Bảng Sách

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| book\_id | Nvarchar | Mã sách |
| book\_name | Nvarchar | Tên sách |
| Type\_id | Nvarchar | Mã loại sách |
| quantity | smallint | Số lượng |
| Author\_name | Nvarchar | Tên tác giả |

1. Bảng Loại Sách

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Type\_id | Nvarchar | Mã loại sách |
| Type\_name | Nvarchar | Tên loại sách |

1. Bảng Nhân Viên

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Staff\_id | Nvarchar | Mã nhân viên |
| name | Nvarchar | Tên nhân viên |
| gender | Nvarchar | Giới tính |
| Designation\_id | Nvarchar | Mã chức vụ |
| address | Nvarchar | Địa chỉ |
| phone | Nvarchar | Số điện thoại |

1. Bảng Chức Vụ NV

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Designation\_id | Nvarchar | Mã chức vụ |
| Designation | Nvarchar | Tên chức vụ |

1. Bảng Đọc Giả

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| Student\_id | Nvarchar | Mã đọc giả |
| Student\_name | Nvarchar | Tên đọc giả |
| phone | Nvarchar | Số điện thoại |

1. Bảng Mượn

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| issue\_id | Nvarchar | Mã mượn |
| book\_id | Nvarchar | Mã sách |
| Date\_issue | date | Ngày mượn |
| Date\_expirary | date | Hạn trả |
| Student\_id | Nvarchar | Mã đọc giả |
| Satff\_id | Nvarchar | Mã nhân viên |

1. Bảng Trả

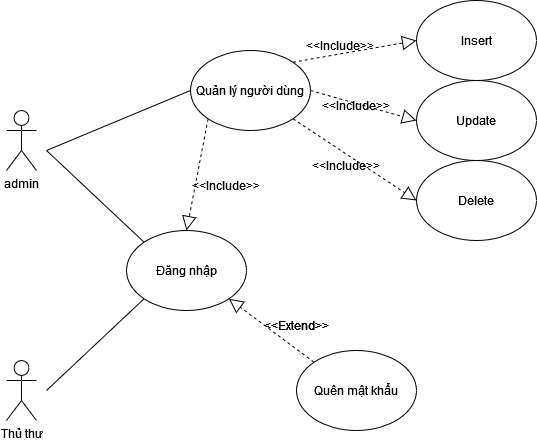
|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| return\_id | Nvarchar | Mã trả |
| issue\_id | Nvarchar | Mã mượn |
| Staff\_id | Nvarchar | Mã nhân viên |
| Date\_return | date | Ngày trả |
| Book\_id | Nvarchar | Mã sách |
| Date\_expirary | date | Hạn trả |

### 3.1.4. Biểu đồ phân rã các UC

* **Biểu đồ Usecase tổng quát**

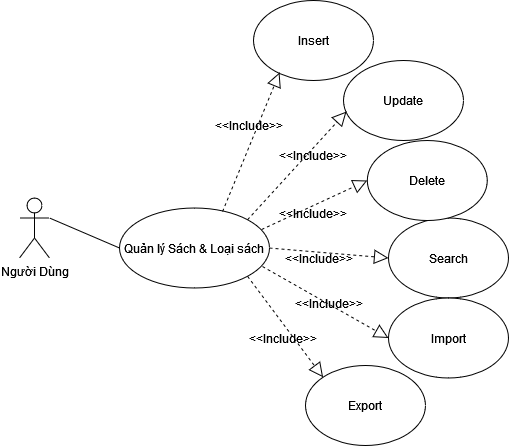
****

* **Biểu đồ Usecase “Quản trị hệ thống”**

****

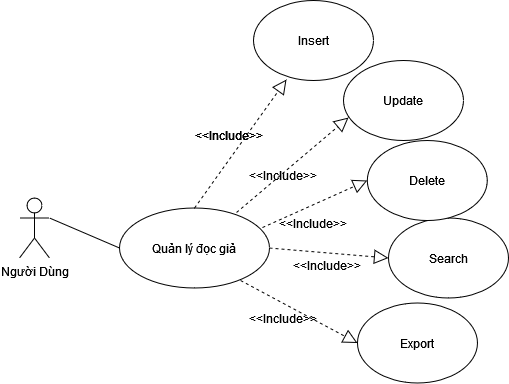
* **Biểu đồ Usecase “Quản lý sách & loại sách”**

Tác nhân “người dùng” bao gồm admin và thủ thư

****

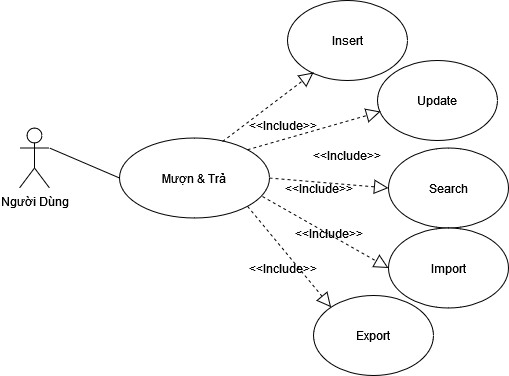
* **Biểu đồ Usecase “Quản lý độc giả”**

Tác nhân “người dùng” bao gồm admin và thủ thư

****

* **Biểu đồ Usecase “Quản lý mượn, trả tài liệu”**

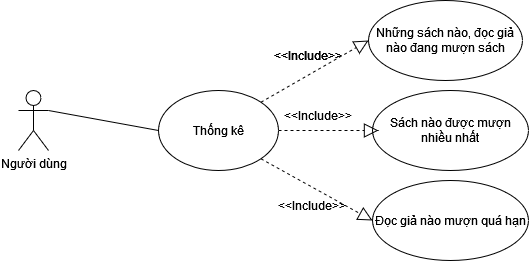
Tác nhân “người dùng” bao gồm admin và thủ thư

****

**Biểu đồ Usecase “Quản lý nhân sự thư viện”**

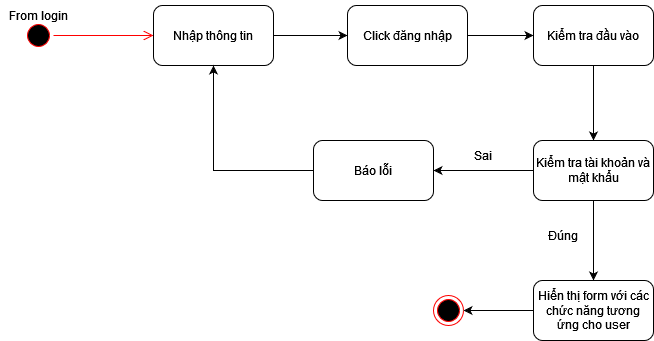
* **Biểu đồ Usecase “Báo cáo thông kê”**

Tác nhân “người dùng” bao gồm admin và thủ thư

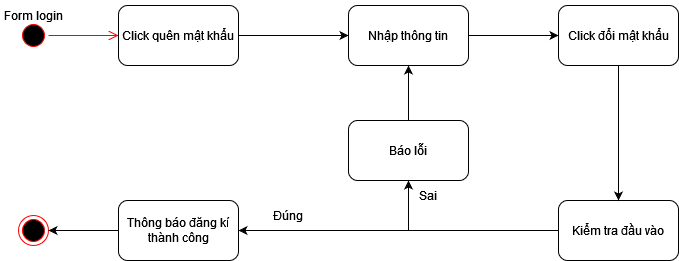
****

### 3.1.5. Biểu đồ hoạt động

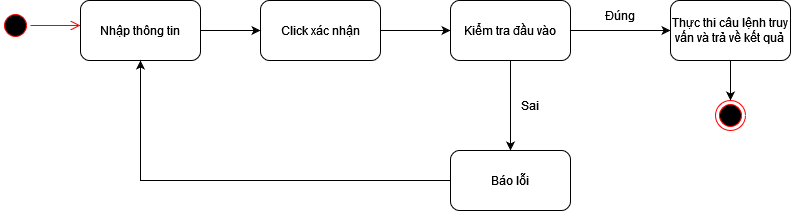
* **Biểu đồ hoạt động đăng nhập**

****

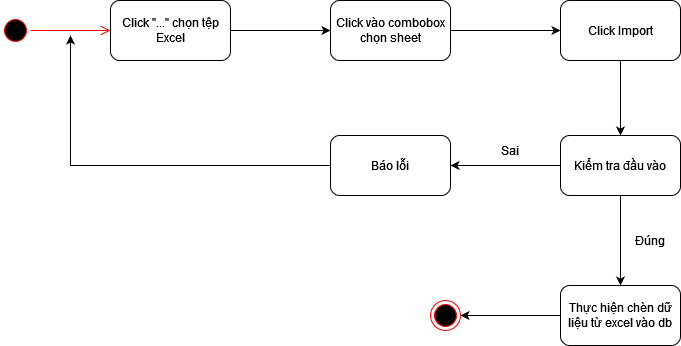
* **Biểu đồ hoạt động quên mật khẩu**



* **Biểu đồ hoạt động CRUD(Create-Read-Update-Delete)**



* **Biểu đồ hoạt động Import**

**c**

* **Biểu đồ hoạt động Export**

****

# CHƯƠNG IV. LẬP TRÌNH VÀ KẾT QUẢ

## 4.1. Các hàm chức năng

### 4.1.1 Kết nối tới Sql

public static SqlConnection GetDBConnection()

{

string connString = @"Data Source=DG;Initial Catalog=QLThuVien;Persist Security Info=True;User ID=sa;Password=a12345678";

SqlConnection con = new SqlConnection(connString);

return con;

}

### 4.1.2 Kiểm tra trùng lặp mã

public class ID\_Check

{

SqlConnection con = DBConnect.GetDBConnection();

public int KiemTraMa(string TenBang, string TenField, string DieuKien)

{

int dieuKien = 0;

DataSet ds = new DataSet();

string strSQL = " Select \* From " + TenBang;

if (TenField != "" && DieuKien != "")

{

strSQL += " Where " + TenField + "='" + DieuKien + "'";

}

SqlDataAdapter da = new SqlDataAdapter(strSQL, con);

da.Fill(ds, TenBang);

DataTable table = ds.Tables[0];

foreach (DataRow row in table.Rows)

{

dieuKien++;

}

return dieuKien;

}

}

### 4.1.3 Kiểm tra số lượng sách

public class SLSach

{

SqlConnection con = DBConnect.GetDBConnection();

public void SLTang(string book\_id)

{

if (con.State != ConnectionState.Open)

{

con.Open();

}

string sql = "update books " +

"set quantity=quantity + 1 " +

" Where book\_id='" + book\_id + "'";

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

}

public void SLGiam(string book\_id)

{

if (con.State != ConnectionState.Open)

{

con.Open();

}

string sql = "update books " +

"set quantity=quantity - 1 " +

" Where book\_id='" + book\_id + "'";

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

}

public int CheckSL(string book\_id)

{

if (con.State != ConnectionState.Open)

{

con.Open();

}

int sl = 0;

string sql = "Select quantity From books Where book\_id='" + book\_id + "'";

SqlCommand cmd = new SqlCommand(sql, con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

string strSL = dr["quantity"].ToString().Trim();

if (strSL == "")

sl = 0;

else

sl = Convert.ToInt32(strSL);

}

con.Close();

return sl;

}

}

### 4.1.4 MD5\_algorithm

public class MD5\_algorithm

{

public String MaHoaString(String giatriCanMa, String matKhau)

{

byte[] bGiaTriCanMa = Encoding.UTF8.GetBytes(giatriCanMa);

byte[] bMatKhau = Encoding.UTF8.GetBytes(matKhau);

byte[] ketQua = MaHoa(bGiaTriCanMa, bMatKhau);

return Convert.ToBase64String(ketQua);

}

public String GiaiMaSring(String giatriCanGiaiMa, String matKhau)

{

byte[] bGiaTriCanGiaiMa = Convert.FromBase64String(giatriCanGiaiMa);

byte[] bMatKhau = Encoding.UTF8.GetBytes(matKhau);

byte[] ketQua = GiaiMa(bGiaTriCanGiaiMa, bMatKhau);

return Encoding.UTF8.GetString(ketQua);

}

public byte[] MaHoa(byte[] giatriCanMa, byte[] matKhau)

{

//Lấy mật khẩu thông qua md5

MD5CryptoServiceProvider md5 = new MD5CryptoServiceProvider();

byte[] keys = md5.ComputeHash(matKhau);

//Tạo bộ cài đặt mã hoá

TripleDESCryptoServiceProvider trip = new TripleDESCryptoServiceProvider();

trip.Key = keys;

trip.Mode = CipherMode.ECB;

trip.Padding = PaddingMode.PKCS7;

//Tạo bộ mã hoá và mã hoá

ICryptoTransform transform = trip.CreateEncryptor();

return transform.TransformFinalBlock(giatriCanMa, 0, giatriCanMa.Length);

}

public byte[] GiaiMa(byte[] giatriCanGiaiMa, byte[] matKhau)

{

//Lấy mật khẩu thông qua md5

MD5CryptoServiceProvider md5 = new MD5CryptoServiceProvider();

byte[] keys = md5.ComputeHash(matKhau);

//Tạo bộ cài đặt mã hoá

TripleDESCryptoServiceProvider trip = new TripleDESCryptoServiceProvider();

trip.Key = keys;

trip.Mode = CipherMode.ECB;

trip.Padding = PaddingMode.PKCS7;

//Tạo bộ mã hoá và mã hoá

ICryptoTransform transform = trip.CreateDecryptor();

return transform.TransformFinalBlock(giatriCanGiaiMa, 0, giatriCanGiaiMa.Length);

}

}

### 4.1.5 Hiển thị data từ sql lên datagridview

public void HienThi()

{

string sqlSelect = "select \* from staff";

SqlCommand cmd = new SqlCommand(sqlSelect, con);

SqlDataReader dr = cmd.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

//Giải Mã

MD5\_algorithm md5 = new MD5\_algorithm();

foreach (DataRow row in dt.Rows)

{

row["name"] = md5.GiaiMaSring(row["name"].ToString(), "NguyenKhaDang");

row["gender"] = md5.GiaiMaSring(row["gender"].ToString(), "NguyenKhaDang");

row["designation\_id"] = md5.GiaiMaSring(row["designation\_id"].ToString(), "NguyenKhaDang");

row["address"] = md5.GiaiMaSring(row["address"].ToString(), "NguyenKhaDang");

row["phone"] = md5.GiaiMaSring(row["phone"].ToString(), "NguyenKhaDang");

}

dataView.DataSource = dt;

}

### 4.1.6 Insert

private void btInsert\_Click(object sender, EventArgs e)

{

if (String.IsNullOrEmpty(tbStaff\_id.Text) || String.IsNullOrEmpty(tbName.Text) || String.IsNullOrEmpty(tbGender.Text) || String.IsNullOrEmpty(tbDesignation\_id.Text) || String.IsNullOrEmpty(tbAddress.Text) || String.IsNullOrEmpty(tbPhone.Text))

{

MessageBox.Show("Bạn phải nhập đầy đủ dữ liệu vào");

return;

}

if (dieuKien.KiemTraMa("staff", "staff\_id", tbStaff\_id.Text) > 0)

{

MessageBox.Show("Mã nhân viên đã tồn tại");

MessageBox.Show("Hãy nhập mã khác");

return;

}

if (dieuKien.KiemTraMa("designation", "designation\_id", tbDesignation\_id.Text) == 0)

{

MessageBox.Show("Mã chức vụ ko tồn tại");

return;

}

else

{

string sqlThem = "INSERT INTO staff " +

"VALUES (@staff\_id, @name, @gender, @designation\_id, @address, @phone)";

SqlCommand cmd = new SqlCommand(sqlThem, con);

//Mã Hoá

MD5\_algorithm md5 = new MD5\_algorithm();

cmd.Parameters.AddWithValue("staff\_id", tbStaff\_id.Text);

cmd.Parameters.AddWithValue("name", md5.MaHoaString((tbName.Text), "NguyenKhaDang"));

cmd.Parameters.AddWithValue("gender", md5.MaHoaString((tbGender.Text), "NguyenKhaDang"));

cmd.Parameters.AddWithValue("designation\_id", md5.MaHoaString((tbDesignation\_id.Text), "NguyenKhaDang"));

cmd.Parameters.AddWithValue("address", md5.MaHoaString((tbAddress.Text), "NguyenKhaDang"));

cmd.Parameters.AddWithValue("phone", md5.MaHoaString((tbPhone.Text), "NguyenKhaDang"));

cmd.ExecuteNonQuery();

HienThi();

}

}

### 4.1.7 Update

private void btUpdate\_Click(object sender, EventArgs e)

{

if (String.IsNullOrEmpty(tbStaff\_id.Text) || String.IsNullOrEmpty(tbName.Text) || String.IsNullOrEmpty(tbGender.Text) || String.IsNullOrEmpty(tbDesignation\_id.Text) || String.IsNullOrEmpty(tbAddress.Text) || String.IsNullOrEmpty(tbPhone.Text))

{

MessageBox.Show("Bạn phải nhập đầy đủ dữ liệu vào");

return;

}

if (dieuKien.KiemTraMa("designation", "designation\_id", tbDesignation\_id.Text) == 0)

{

MessageBox.Show("Mã chức vụ tồn tại");

return;

}

else

{

string sqlThem = "update staff " +

"set staff\_id=@staff\_id, name=@name, gender=@gender, designation\_id=@designation\_id, address=@address, phone=@phone " +

"where staff\_id=@staff\_id";

SqlCommand cmd = new SqlCommand(sqlThem, con);

MD5\_algorithm md5 = new MD5\_algorithm();

cmd.Parameters.AddWithValue("staff\_id", tbStaff\_id.Text);

cmd.Parameters.AddWithValue("name", md5.MaHoaString((tbName.Text), "NguyenKhaDang"));

cmd.Parameters.AddWithValue("gender", md5.MaHoaString((tbGender.Text), "NguyenKhaDang"));

cmd.Parameters.AddWithValue("designation\_id", md5.MaHoaString((tbDesignation\_id.Text), "NguyenKhaDang"));

cmd.Parameters.AddWithValue("address", md5.MaHoaString((tbAddress.Text), "NguyenKhaDang"));

cmd.Parameters.AddWithValue("phone", md5.MaHoaString((tbPhone.Text), "NguyenKhaDang"));

cmd.ExecuteNonQuery();

HienThi();

}

}

### 4.1.8 Delete

private void btDelete\_Click(object sender, EventArgs e)

{

if (String.IsNullOrEmpty(tbStaff\_id.Text))

{

MessageBox.Show("Bạn phải nhập mã nhân viên vào");

return;

}

string sqlThem = "delete from staff " +

"where staff\_id=@staff\_id";

SqlCommand cmd = new SqlCommand(sqlThem, con);

cmd.Parameters.AddWithValue("staff\_id", tbStaff\_id.Text);

cmd.Parameters.AddWithValue("name", tbName.Text);

cmd.Parameters.AddWithValue("gender", tbGender.Text);

cmd.Parameters.AddWithValue("designation\_id", tbDesignation\_id.Text);

cmd.Parameters.AddWithValue("address", tbAddress.Text);

cmd.Parameters.AddWithValue("phone", tbPhone.Text);

cmd.ExecuteNonQuery();

HienThi();

}

### 4.1.9 Search

private void btTimKiem\_Click(object sender, EventArgs e)

{

if (String.IsNullOrEmpty(tbNoiDungTimKiem.Text))

{

MessageBox.Show("Bạn phải nhập dữ liệu cần tìm vào");

return;

}

(dataView.DataSource as DataTable).DefaultView.RowFilter = String.Format("staff\_id like '%" + tbNoiDungTimKiem.Text + "%' or name like '%" + tbNoiDungTimKiem.Text + "%' or gender like '%" + tbNoiDungTimKiem.Text + "%' or designation\_id like '%" + tbNoiDungTimKiem.Text + "%' or address like '%" + tbNoiDungTimKiem.Text + "%' or phone like '%" + tbNoiDungTimKiem.Text + "%'");

}

### 4.1.10 Export

// khởi tạo Excel Application

Microsoft.Office.Interop.Excel.\_Application app = new Microsoft.Office.Interop.Excel.Application();

// tạo WorkBook mới trong Excel Application

Microsoft.Office.Interop.Excel.\_Workbook workbook = app.Workbooks.Add(Type.Missing);

// tạo Excelsheet mới

Microsoft.Office.Interop.Excel.\_Worksheet worksheet = null;

// mở bảng excel đằng sau chương trình

app.Visible = true;

// lấy tham chiếu của trang tính đầu tiên. Theo mặc định, tên của nó là Sheet1

worksheet = workbook.Sheets["Sheet1"];

// lưu trữ tham chiếu của nó vào sheet

worksheet = workbook.ActiveSheet;

// thay đổi tên của sheet

worksheet.Name = "type";

// lấy tên cột

for (int i = 1; i < dataView.Columns.Count + 1; i++)

{

worksheet.Cells[1, i] = dataView.Columns[i - 1].HeaderText;

}

// lưu trữ từng giá trị hàng và cột vào sheet

for (int i = 0; i < dataView.Rows.Count; i++)

{

for (int j = 0; j < dataView.Columns.Count; j++)

{

worksheet.Cells[i + 2, j + 1] = dataView.Rows[i].Cells[j].Value.ToString();

}

}

### 4.1.11 Import

if (String.IsNullOrEmpty(tbFileName.Text))

{

MessageBox.Show("Bạn phải chọn tệp dữ liệu để nhập vào");

return;

}

try

{

string connecionString = "Server=DG;Database=QLThuVien;User Id=sa;Password=a12345678;";

DapperPlusManager.Entity<DTO\_giveback>().Table("giveback");

List<DTO\_giveback> temp = givebackBindingSource.DataSource as List<DTO\_giveback>;

if (temp != null)

{

using (IDbConnection db = new SqlConnection(connecionString))

{

db.BulkInsert(temp);

}

MessageBox.Show("Import thành công");

HienThi();

}

}

catch (Exception ex)

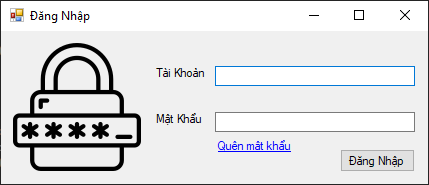
{

MessageBox.Show(ex.Message, "Message", MessageBoxButtons.OK, MessageBoxIcon.Error);

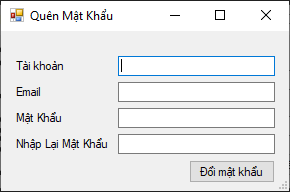
}

## 4.2. Kết quả lập trình

### 4.2.1 Form đăng nhập



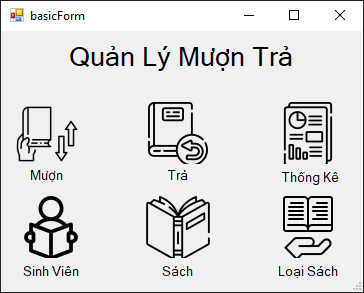
### 4.2.2. Form quên mật khẩu



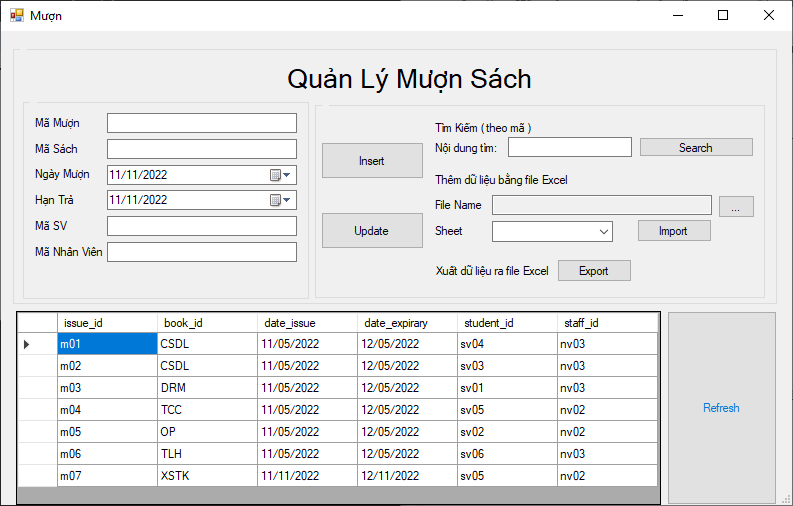
### 4.2.3. Form cho admin



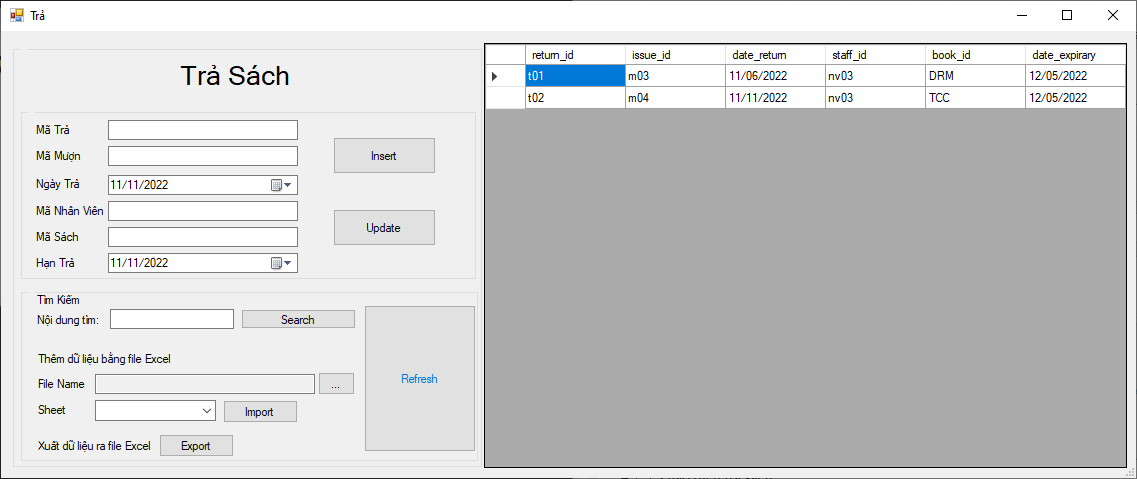
### 4.2.4. Form cho nhân viên



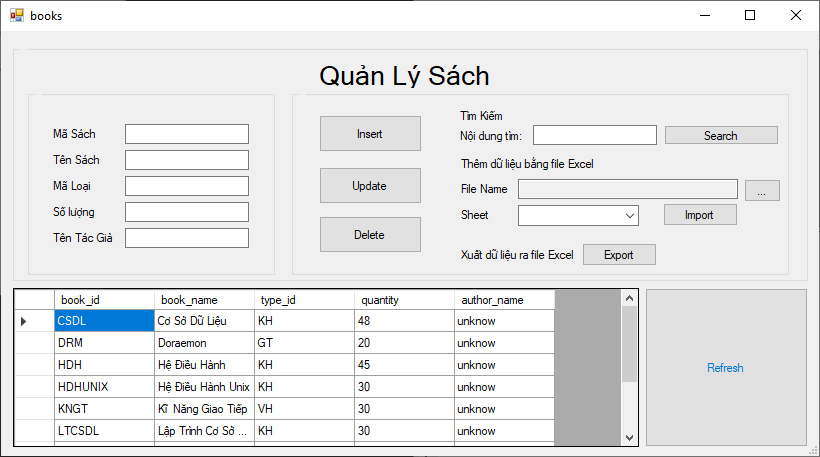
### 4.2.5. Form quản lý mượn sách



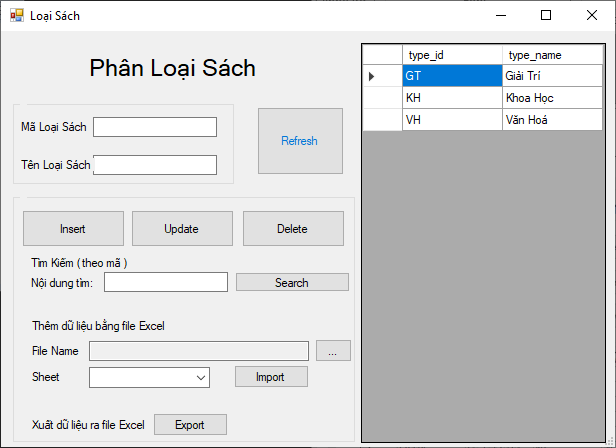
### 4.2.6. Form quản lý trả sách



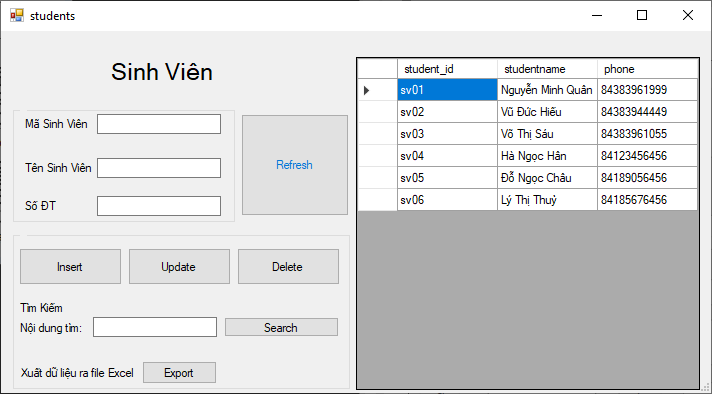
### 4.2.7. Form quản lý sách



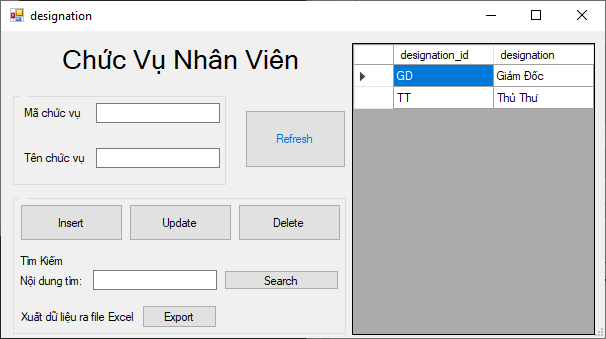
### 4.2.8. Form quản lý loại sách



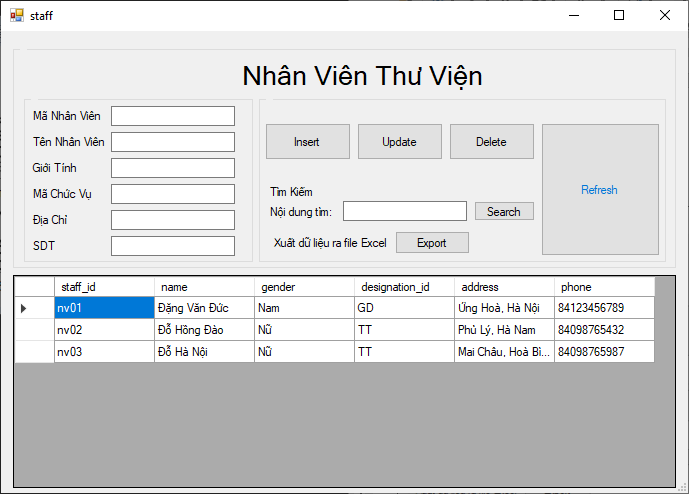
### 4.2.9. Form quản lý đọc giả



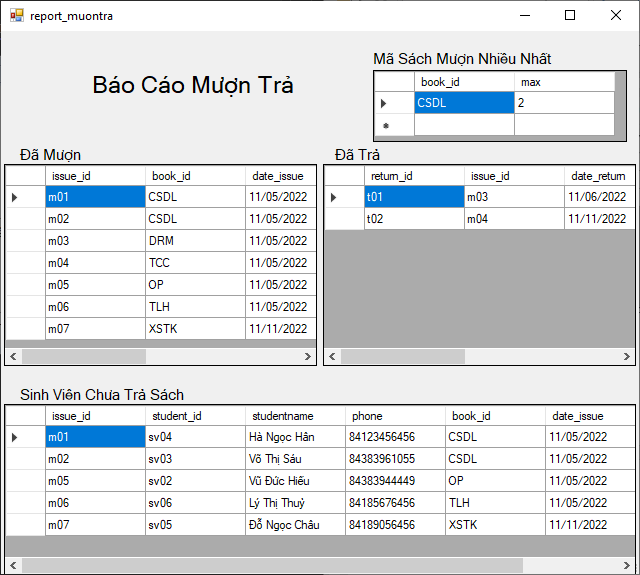
### 4.2.10. Form quản lý chức vụ nhân viên



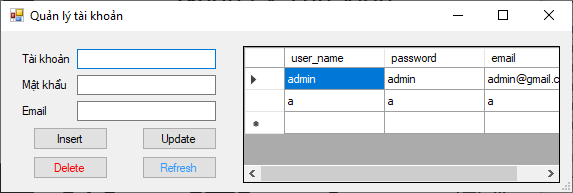
### 4.2.11. Form quản lý nhân viên thư viện



### 4.2.12. Form báo cáo thống kê



### 4.2.13. Form quản lý tài khoản



# CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

### 5.1.1. Kết quả đạt được

Qua quá trình tìm hiểu, khảo sát quy trình nghiệp vụ quản lý thư viện tại Trường. Trong thời gian một tháng nguyên cứu, tìm hiểu, em đã đạt được một số kết quả sau:

- Hiểu được quy trình, nghiệp vụ quản lý thư viện: quy trình nhập tài liệu, quy trình mượn trả tài liệu, quy trình thống kê báo cáo tài liệu.

- Phân tích được quy trình hoạt động và các chức năng của hệ thống biểu diễn qua các biểu đồ UML và xây dựng cơ sở dữ liệu phù hợp trên hệ quản trị cơ sở dữ liệu SQL Server.

- Xây dựng phần mềm quản lý và website tìm kiếm thông qua quá trình khảo sát, phân tích thiết kế hệ thống, phân tích thiết kế cơ sở dữ liệu, thiết kế và đặc tả giao diện.

### 5.1.2. Hạn chế

Vì là lần đầu nghiên cứu tìm hiểu xây dựng một hệ thống mang tính thực tế, tuy đề tài đã hoàn thành nhưng vẫn còn nhiều hạn chế:

- Chương trình chỉ mang tính chất tìm hiểu ngôn ngữ, chưa có khả năng ứng dụng vào

thực tế.

## 5.2. Hướng phát triển

Từ những hạn chế trên, cần có một hướng phát triển mới để đề tài ngày càng hoàn thiện hơn:

- Tiếp tục nguyên cứu, xây dựng để phần mềm ngày càng hoàn thiện và có thể áp dụng vào thư viện của trường.

- Áp dụng công nghệ mã vạch vào quá trình quản lý tài liệu, quản lý độc giả.

- Hỗ trợ gửi email tự động hay thông qua hệ thống SMS để thông báo tới độc giả mượn quá hạn.

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))

[5]. ***Import - export*** (https://www.youtube.com/c/foxlearn)