**LỜI CẢM ƠN!**

“ Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn – Thầy Lê Hoàng Việt Tuấn đã *quan tâm hướng dẫn*, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học Phát triển ứng dụng giao diện của thầy, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để chúng em có thể vững bước sau này.

Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù, chúng em đã cố gắng hết sức nhưng chắc chắn khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong quý thầy xem xét và góp ý để đồ án của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!”

TP.HCM, tháng 5 năm 2020

**LỜI NÓI ĐẦU**

Công nghệ Thông tin là một trong những ngành khoa học ngày càng được quan tâm và sử dụng rộng rãi trong mọi lĩnh vực của cuộc sống. Với những ưu điểm mạnh, có thể ứng dụng được nhiều trong cuộc sống thì công nghệ thông tin đã giúp cho công việc quản lý được dễ dàng hơn. Để quản lý một công việc cụ thể của một cơ quan nào đó thì người quản lý cần thực hiện rất nhiều công việc. Nhờ có các phần mềm thì công việc đó trở nên đơn giản hơn rất nhiều.

Hệ thống quản lý này sẽ giúp giải quyết được những khó khăn và tăng tính hiệu quả trong công tác quản lý thư viện. Vì sự hiểu biết và kinh nghiệm còn hạn chế nên chắc chắn đồ án của chúng em vẫn còn nhiều thiếu sót. Rất mong nhận được ý kiến đóng góp của thầy và các bạn.

MỤC LỤC

[PHẦN 1: GIỚI THIỆU LẬP TRÌNH ỨNG DỤNG GIAO DIỆN 4](#_Toc56161089)

[1. Giao diện lập trình ứng dụng là gì? 4](#_Toc56161090)

[2. Có những loại API nào? 4](#_Toc56161091)

[3. Ưu điểm của API 5](#_Toc56161092)

[4. Nhược điểm của API 5](#_Toc56161093)

[PHẦN 2: PHẦN MỀM QUẢN LÝ THƯ VIỆN 6](#_Toc56161094)

[CHƯƠNG 1: TỔNG QUAN HỆ THỐNG QUẢN LÝ THƯ VIỆN 6](#_Toc56161095)

[1. Giới thiệu 6](#_Toc56161096)

[2. Nhiệm vụ hệ thống 6](#_Toc56161097)

[3. Chức năng của hệ thống 6](#_Toc56161098)

[4. Các biểu mẫu sử dụng trong hệ thống 7](#_Toc56161099)

[CHƯƠNG 2: CÁC CHỨC NĂNG CHÍNH CỦA PHẦN MỀM QUẢN LÝ 10](#_Toc56161100)

[THƯ VIỆN 10](#_Toc56161101)

[1. Quản lý độc giả 10](#_Toc56161102)

[2. Quản lý Sách 10](#_Toc56161103)

[4. Các form phụ 11](#_Toc56161104)

[CHƯƠNG 3: SƠ ĐỒ QUAN HỆ VÀ CÁC LỚP GIẢI QUYẾT CHỨC NĂNG 14](#_Toc56161105)

[1. Sơ đồ quan hệ 14](#_Toc56161106)

[2. Các lớp giải quyết vấn đề 14](#_Toc56161107)

[a. Lớp độc giả 14](#_Toc56161108)

[b. Lớp Sách 27](#_Toc56161109)

[CHƯƠNG 4: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM 42](#_Toc56161110)

[CHƯƠNG 5: ĐÁNH GIÁ PHẦN MỀM QUẢN LÝ THƯ VIỆN 44](#_Toc56161111)

[Tổng kết 44](#_Toc56161112)

[1. Khả thi về kỹ thuật 44](#_Toc56161113)

[2. Công cụ phát triển 44](#_Toc56161114)

[3. Bảng tổng kết 44](#_Toc56161115)

[4. Yêu cầu bảo mật 45](#_Toc56161116)

[5. Ngôn ngữ viết 45](#_Toc56161117)

[KẾT LUẬN 45](#_Toc56161118)

[BẢNG PHÂN CÔNG 46](#_Toc56161119)

[NGUỒN THAM KHẢO: 46](#_Toc56161120)

# PHẦN 1: GIỚI THIỆU LẬP TRÌNH ỨNG DỤNG GIAO DIỆN

# Giao diện lập trình ứng dụng là gì?

API (viết tắt từ Application Programming Interface) là một giao diện mà một hệ thống máy tính hay ứng dụng cung cấp cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng. Chẳng hạn, một chương trình máy tính có thể (và thường là phải) dùng các hàm API của [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh" \o "Hệ điều hành) để xin cấp phát bộ nhớ và truy xuất tập tin. Nhiều loại hệ thống và ứng dụng thực hiện API, như các hệ thống đồ họa, cơ sở dữ liệu, mạng, dịch vụ web, và ngay cả một số trò chơi máy tính.

Đây là phần mềm hệ thống cung cấp đầy đủ các chức năng và các tài nguyên mà các lập trình viên có thể rút ra từ đó để tạo nên các tính năng giao tiếp người- máy như: các trình đơn kéo xuống, tên lệnh, hộp hội thoại, lệnh bàn phím và các cửa sổ. Một trình ứng dụng có thể sử dụng nó để yêu cầu và thi hành các dịch vụ cấp thấp do hệ điều hành của máy tính thực hiện. Hệ giao tiếp lập trình ứng dụng giúp ích rất nhiều cho người sử dụng vì nó cho phép tiết kiệm được nhiều thời gian tìm hiểu các chương trình mới, do đó khích lệ mọi người dùng nhiều ứng dụng hơn.

# Có những loại API nào?

* Dựa trên phân ngành trong ngành công nghệ thông tin thì API có các loại sau:

#### ****Web API – hệ thống API trên nền tảng web****

Loại này rất phổ biến, các website lớn đều [thiết kế web-app](https://freelancervietnam.vn/thiet-ke-web-application-dich-vu-freelancer-lap-trinh-web-app-co-phai-lua-chon-tot-danh-cho-ban/) với nền tảng hệ thống API giúp bạn kết nối, lấy dữ liệu hoặc đôi khi là cập nhật dữ liệu vào hệ thống.

#### ****Hệ thống API trên hệ điều hành****

**Windows**cũng như **Linux** cung cấp các tài liệu API đặc tả các hàm, phương thức và các giao thức kết nối. Nhờ đó, lập trình viên có thể tạo được các phần mềm ứng dụng có chức năng tương tác với hệ điều hành. Đối với Windows thì bạn có thể [tạo ứng dụng chạy trên máy tính bằng C++ và Win32 API](https://docs.microsoft.com/en-us/windows/win32/). Còn với Linux bạn có thể tham khảo thử [Electron API](https://electronjs.org/).

#### ****API của thư viện phần mềm (framework)****

API mô tả cũng như quy định các hoạt động mong muốn mà thư viện cung ứng. Một API có thể có nhiều cách triển khai hoạt động khác nhau. API cũng có thể giúp cho một chương trình được viết bằng ngôn ngữ này nhưng có thể sử dụng được thư viện được viết bằng ngôn ngữ khác.

* Dựa trên quyền truy cập thì có thể phân API thành các loại sau:
* **API mở (Open API):** Có sẵn, công khai, không hạn chế quyền truy cập.
* **API đối tác (Partner API):** Cần có quyền hoặc giấy phép cụ thể mới truy cập được.
* **API nội bộ (Internal API):** Chỉ dùng cho các hệ thống nội bộ (chẳng hạn như công ty, tổ chức). Các đội ngũ phát triển nội bộ khác nhau có thể sử dụng chúng để cải thiện cho các sản phẩm hay dịch vụ chỉ phục vụ cho riêng nhân viên công ty.

# **Ưu điểm của API**

**- Tính tự động hóa rất cao:** API có thể thay thế chúng ta quản lý công việc cực kì hiệu quả. API giúp các cơ quan có thể cập nhật, xử lý hoàn thiện công việc nhanh và chất lượng hơn.

**- Ứng dụng cực kỳ linh hoạt:** API có thể truy cập vào các thành phần ứng dụng giúp việc cung cấp dịch vụ và thông tin linh hoạt hơn nhiều.

**- Khả năng thích ứng tốt:** API có chức năng thay đổi cũng như dự đoán thay đổi theo thời gian cho nên dữ liệu được truyền tốt hơn, thông tin được chọn lọc kĩ hơn, dịch vụ tốt hơn.

**- Cá nhân hóa:** Người dùng có thể tinh chỉnh API cho phù hợp nhu cầu sử dụng.

**- Dữ liệu có sẵn:** API cho phép mọi thông tin được tạo ở dạng chính chủ luôn có sẵn cho mọi người xem được.

**- Phạm vi:** Ngoài dữ liệu mới có sẵn được chia sẻ rộng rãi, người dùng còn có thể hiệu chỉnh web API để cung cấp dịch vụ, thông tin cá nhân hóa.

# **Nhược điểm của API**

- API **chỉ hỗ trợ mặc định get, post** chưa hoàn toàn là restful service.

- Cấu hình cố hữu của WFC làm cấu hình của **API cực kì khó nhớ**. Nếu mới sử dụng, người dùng khó mà nhớ và dùng dễ dàng như các ứng dụng khác.

*WCF (Windows Communication Foundations) là mô hình phát triển ứng dụng hướng dịch vụ trên nền tảng của Microsoft.*

**- Web API tốn kha khá chi phí** vận hành, phát triển, hiệu chỉnh và đòi hỏi kiến thức chuyên sâu. Khi bị tấn công hệ thống, người dùng đôi lúc gặp rắc rối về bảo mật.

# PHẦN 2: PHẦN MỀM QUẢN LÝ THƯ VIỆN

# CHƯƠNG 1: TỔNG QUAN HỆ THỐNG QUẢN LÝ THƯ VIỆN

# **Giới thiệu**

Để xây dựng một hệ thống thông tin hoàn chỉnh chúng ta cần trải qua 6 giai đoạn: Khảo sát, phân tích hệ thống, thiết kế hệ thống, kiểm thử, cài đặt, vận hành và bảo trì hệ thống.

Nhưng đối với những sinh viên năm 2 như chúng em thì nhu cầu cấp thiết nhất hiện nay là qua môn và đạt điểm. Cùng với danh sách đề tài được gợi ý nên vì thế chúng em đã xây dựng một chương trình quản lý sinh viên để phục vụ nhu cầu này.

# **Nhiệm vụ hệ thống**

Theo dõi quá trình đọc sách của những độc giả trong suốt quá trình của các thành viên của thư viện, đảm bảo không gặp vấn đề trong quá trình đọc sách và mượn. Hệ thống này giúp cho nhân viên như thủ thư quan sát thành viên một cách nhanh nhất và chính xác nhất. Nó sẽ làm giảm đáng kể công sức, giấy tờ, sổ sách lưu trư.

Ngoài việc quan sát và đảm bảo những trãi nghiệm tốt nhất của từng thành viên, hệ thống cũng có thể tra cứu một số thông tin cần thiết về thành viên như ngày sinh, chỗ ở hiện tại...

Không chỉ để cho thủ thư kiểm soát thành viên, độc giả cũng có thể sử dụng hệ thống dưới quyền thành viên để có thể xem thông tin cá nhân của mình,có thể tra cứu sách có thể mượn mà không cần đến thư viện. Hay là thử quỹ cũng sẽ được sử dụng hệ thống để quản lý về tiền bạc, thủ kho sẽ quản lý sách và thủ trưởng dễ dàng quản lý nhân viên hơn

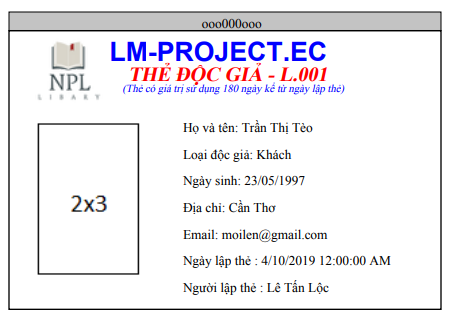
# **Chức năng của hệ thống**

Hệ thống quản lý thư viện bao gồm 3 bộ phận, hoạt động có mối liên hệ chặt chẽ với nhau.

* Bộ phận quản lý thành viên(độc giả): Thay đổi, đăng ký, hủy và kiểm soát các thông tin của độc giả.
* Bộ phận quản lý sách: Bộ phận này có nhiệm vụ quản lý toàn bộ sách trong thư viện,như xử lý những quyển sách bị hư, mất cũng như cập nhật những cuốn sách mới về cho thư viện
* Bộ phận quản lý nhân viên:là bộ phận chỉ có thủ trưởng sử dụng để kiểm soát nhân viên cũng như để giúp cho thủ trưởng giải quyết những vấn đề phát sinh trong nhân viên thư viện

# **Các biểu mẫu sử dụng trong hệ thống**

* Thẻ Độc Giả



* Thông tin Sách

Ảnh có chứa bàn

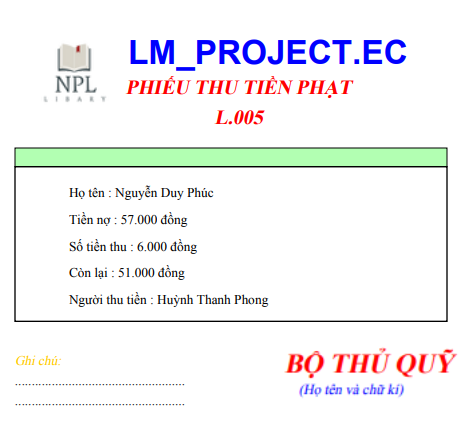
Mô tả được tạo tự động

* Phiếu trả sách

Ảnh có chứa bàn

Mô tả được tạo tự động

* Phiếu thu tiền phạt



* Phiếu ghi nhận mất sách

Ảnh có chứa bàn

Mô tả được tạo tự động

* Phiếu mượn sách

Ảnh có chứa bàn

Mô tả được tạo tự động

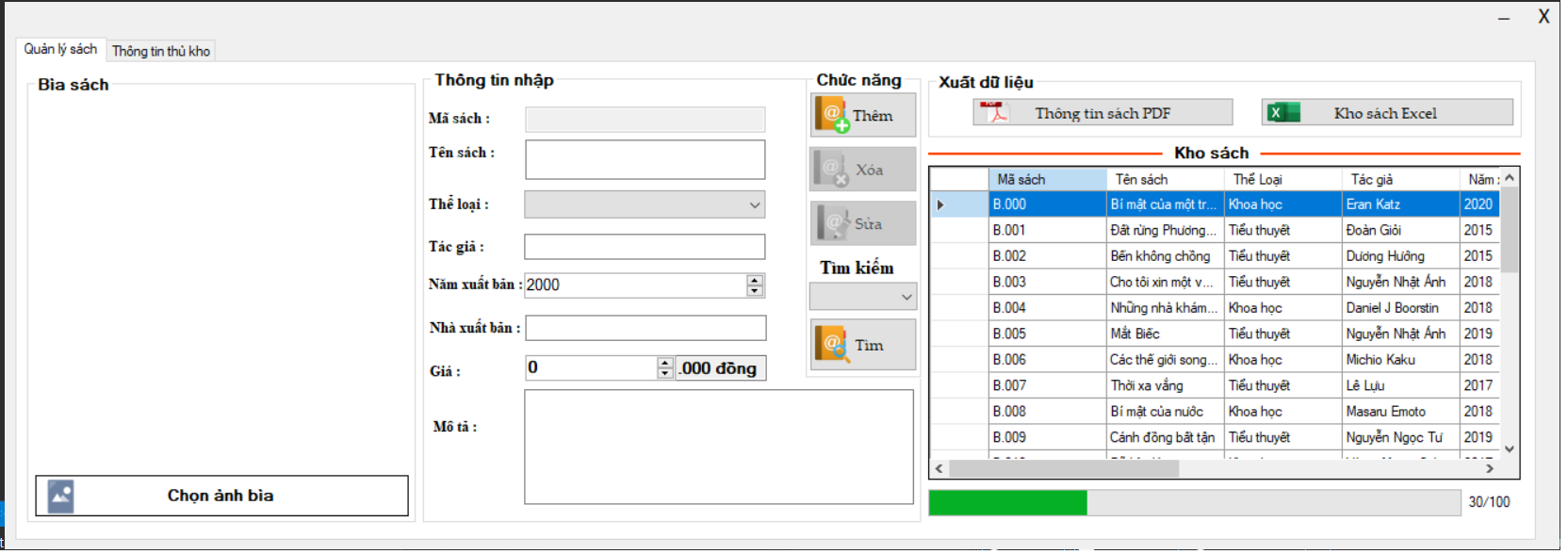
# CHƯƠNG 2: CÁC CHỨC NĂNG CHÍNH CỦA PHẦN MỀM QUẢN LÝ

# THƯ VIỆN

# **Quản lý độc giả**

Form này dung để đăng ký độc giả bởi thủ thư,những người muốn đăng ký phải trục tiếp tới gặp thủ thư để đăng ký.Để có thể cho thủ thư quản lý độc giả có kiểm soát thành viên dễ dàng trong việc tìm kiếm và có thể ngay lập tức cấp thẻ thành viên cho người đăng ký

# **Quản lý Sách**

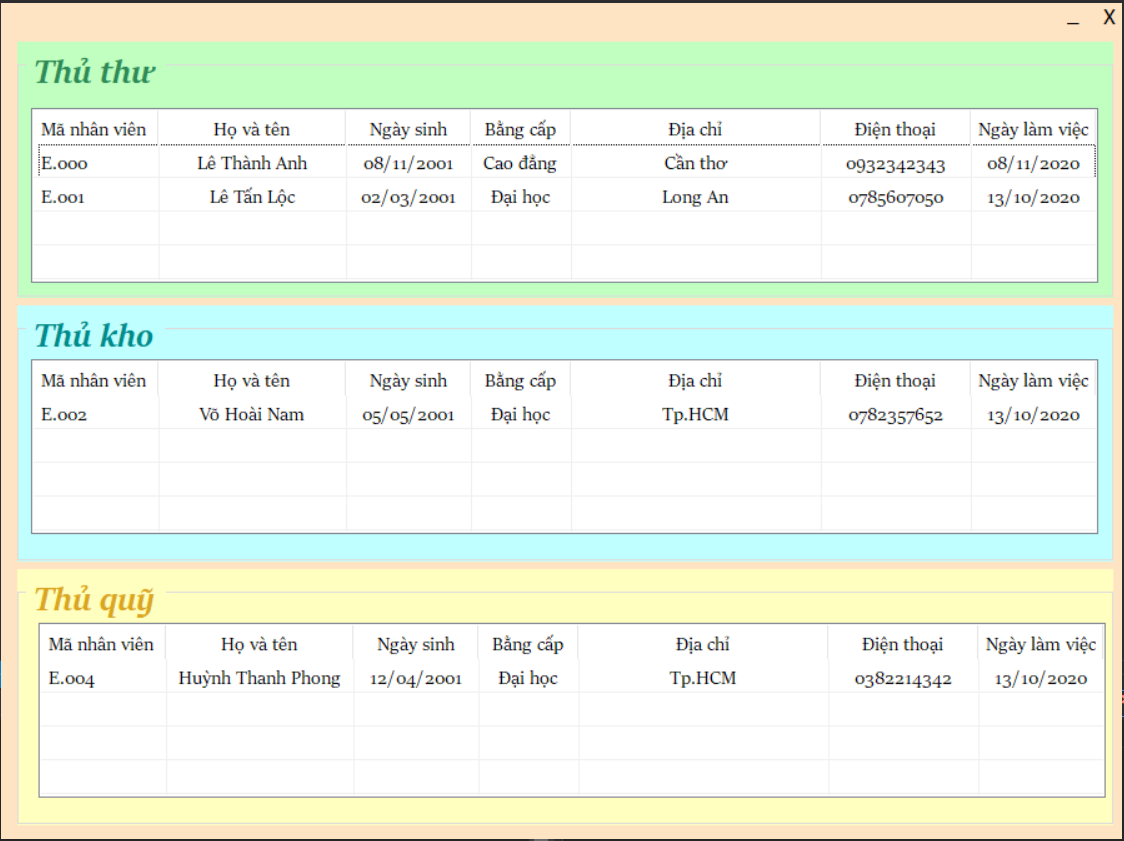


Form được sử dụng bởi thủ kho để quản lý sách trong thư viện giúp những công việc thêm sách

, sách mất và thanh lý trơn tru hơn.Và có thể xuất danh sách trong việc báo cáo

Thủ thư cũng sẽ được sử dụng để tra cứu sách để thông báo khác hàng

1. **Quản lý nhân viên**

****

Được sử dụng bởi thủ trưởng để có thể quản lý nhân viên

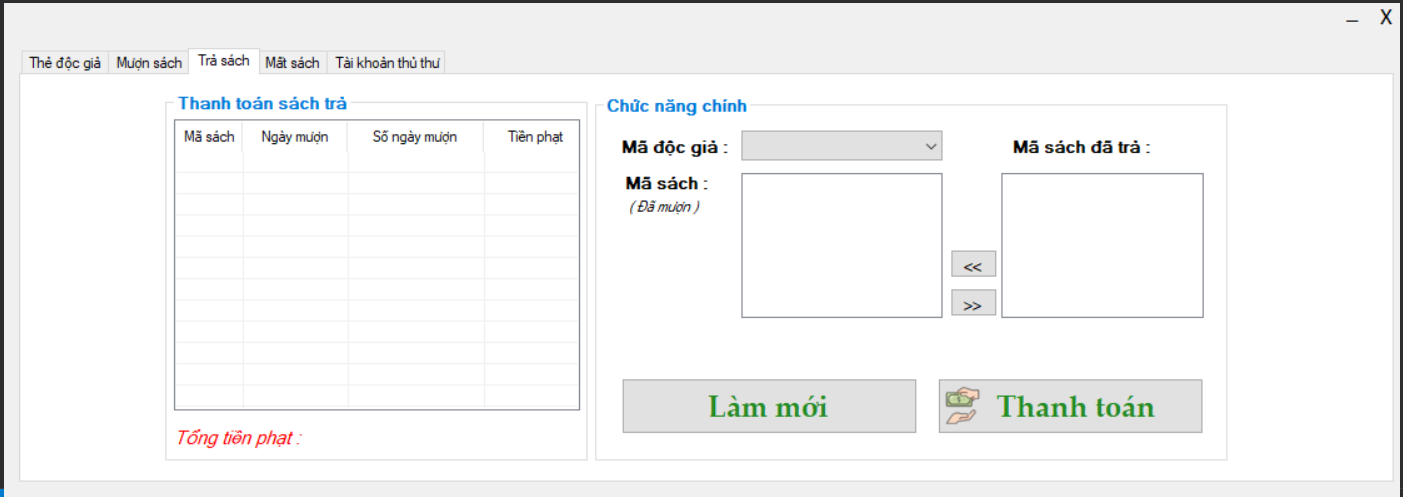
1. Các form phụ

Form mượn sách: để thủ thư có thể quản lý phiếu mượn

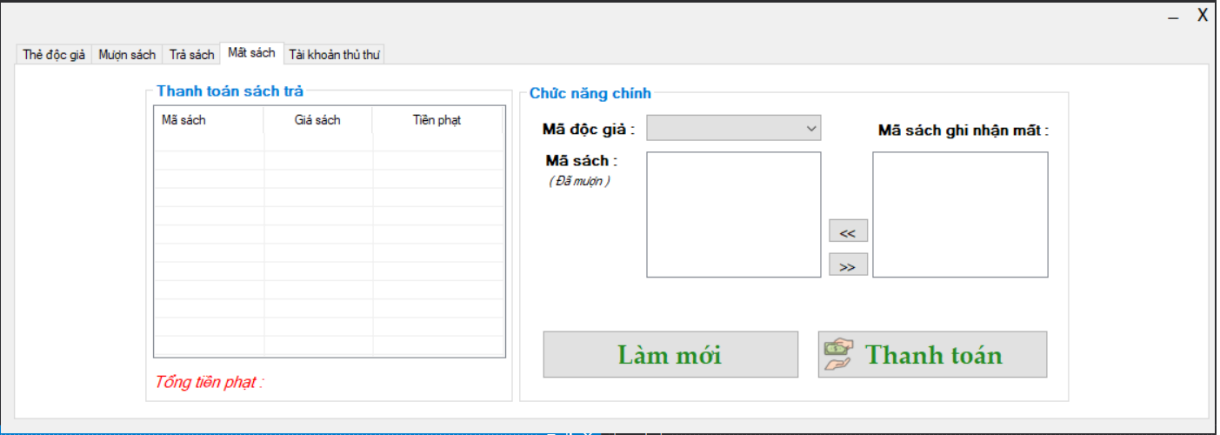
Ảnh có chứa bàn

Mô tả được tạo tự động

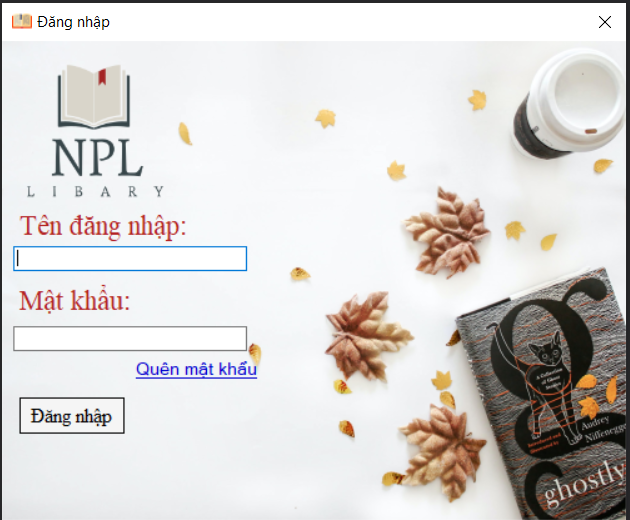
Form trả sách: để thủ thư tiến hành lập phiếu mượn cho độc giả



Form trả sách: Sẽ giúp thủ thư trong trường hợp độc giả làm mất sách

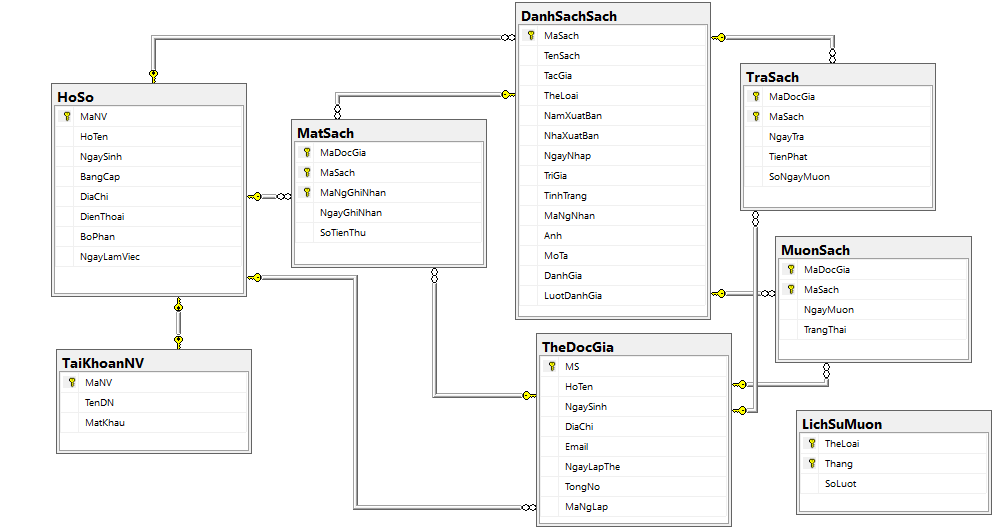


Form đăng nhập: Vì hệ thống có nhiều người sử dụng nên



# CHƯƠNG 3: SƠ ĐỒ QUAN HỆ VÀ CÁC LỚP GIẢI QUYẾT CHỨC NĂNG

# Sơ đồ quan hệ



1. Các lớp giải quyết vấn đề

## **Lớp độc giả**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using data = System.Data.DataTable;

namespace QLDG

{

public partial class QLDG : Form

{

public QLDG()

{

InitializeComponent();

}

SqlConnection con;

// tạo biến để kiểm tra đúng sau khi nhấn button

bool pname = false;

bool paddress = false;

bool pcreater = false;

bool ptype = false;

bool pmail = false;

bool pdate = false;

bool dn = false;

bool mk = false;

int SizeBang;

//===================================bổ trợ======================

int i = 1;

public string TaoMa()

{

if (i < 10) return $"L.00{i.ToString()}";

else if (i < 100) return $"L.0{i.ToString()}";

else return $"L.{i.ToString()}";

}

public void Hienthi()

{

string sql\_select = "select MS as N'Mã độc giả'," +

"G.TenDN as N'Tên đăng nhập'," +

"G.HoTen as N'Họ và tên'," +

"CONVERT (varchar(10), G.NgaySinh, 103) AS N'Ngày sinh'," +

"G.DiaChi as N'Địa chỉ'," +

"G.Loai as N'Loại'," +

"G.Email," +

"CONVERT (varchar(10), G.NgayLapThe, 103) AS N'Ngày lập thẻ'," +

"G.TongNo as N'Tổng nợ'," +

"S.HoTen as N'Tên người lập' " +

"from TheDocGia G,HoSo S where MaNgLap=MaNV ";

SqlCommand cmd = new SqlCommand(sql\_select, con);

SqlDataReader rd = cmd.ExecuteReader();

data d = new data();

d.Load(rd);

dataTheDocGia.DataSource = d;

}

public void Set\_true()

{

pname = true;

paddress = true;

pcreater = true;

ptype = true;

pmail = true;

pdate = true;

dn = true;

mk = true;

er.SetError(f2Name, null);

}

public void Set\_false()

{

Tick.Clear();

pname = false;

paddress = false;

pcreater = false;

ptype = false;

pmail = false;

pdate = false;

dn = false;

mk = false;

er.SetError(f2Name, null);

buttonSua.Enabled = false;

}

void Reset\_ThuocTinh()

{

f2Name.Text = "";

f2Address.Text = "";

textBox2.Text = "";

textBox1.Text = "";

f2Email.Text = "";

f2Type.SelectedIndex = -1;

textMS.Text = "";

}

public void ThongKe()

{

double thongke = (double)((SizeBang - 1) \* 0.2);

progressBar\_thongke.Value = (int)Math.Round(thongke, 0);

// MessageBox.Show($"{progressBar\_thongke.Value.ToString()}");

label\_thongke.Text = $"{(SizeBang - 1).ToString()}/500";

}

//=========================================== Xử lý =================

private void f2Name\_Leave(object sender, EventArgs e)

{

if (f2Name.Text.Length != 0)

{

string fullname = f2Name.Text;

string space = " ";

// Xử lý cắt dấu cách đầu tên

bool dau = false; //dung de check dau cach dung dau

for (int i = 0; i < fullname.Length && dau == false; i++)

{

if (fullname[i] == ' ' && dau == false)

{

fullname = fullname.Remove(i, 1);

i = -1;

}

else dau = true;

}

if (f2Name.Text.Length == 0)

{

er.SetError(this.f2Name, "Vui lòng nhập lại!");

pname = false;

}

//Xử lý cắt dấu cách cuối tên

bool cuoi = false;

for (int i = fullname.Length - 1; i >= 0 && cuoi == false; i--)

{

if (fullname[i] == ' ' && cuoi == false)

{

fullname = fullname.Remove(fullname.Length - 1, 1);

i = fullname.Length;

}

else cuoi = true;

}

if (f2Name.Text.Length == 0)

{

er.SetError(this.f2Name, "Vui lòng nhập lại!");

pname = false;

}

// Xử lý phần dấu cách trước và sau chuỗi xong

f2Name.Text = fullname; // gán lại chuỗi sau khi sửa

//Tiếp tục check phần ký tự

for (int i = 0; i < f2Name.Text.Length; i++)

{

if (f2Name.Text[i] >= 33 && f2Name.Text[i] <= 57 || f2Name.Text[i] >= 58 && f2Name.Text[i] <= 64 || f2Name.Text[i] >= 91 && f2Name.Text[i] <= 96 || f2Name.Text[i] >= 123 && f2Name.Text[i] <= 126)

{

f2Name.Text = f2Name.Text.Remove(i, 1);

i = -1;

}

}

if (f2Name.Text.Length == 0)

{

Tick.Clear();

er.SetError(this.f2Name, "Vui lòng nhập lại!");

pname = false;

}

if (f2Name.Text.Length != 0)

{

er.Clear();

//Đếm số lượng từ có trong tên để viết hoa chữ đầu

int slc = 1;

for (int i = 0; i < f2Name.Text.Length; i++)

if (f2Name.Text[i] == ' ') slc++;

//Viết hoa chữ đầu và nhấn đầu chữ sau

string[] parts = f2Name.Text.Split(new string[] { space }, StringSplitOptions.RemoveEmptyEntries);

f2Name.Text = "";

bool hetchu = false;

for (int i = 0; i < slc; i++)

{

parts[i] = parts[i].Substring(0, 1).ToUpper() + parts[i].Substring(1, parts[i].Length - 1).ToLower();

f2Name.Text = f2Name.Text + parts[i];

if (i >= 0 && i < slc) hetchu = true;

if (hetchu == true) f2Name.Text += " ";

}

}

}

if (f2Name.Text.Length == 0)

{

Tick.Clear();

er.SetError(this.f2Name, "Bạn chưa nhập tên! ");

pname = false;

}

else

{

er.Clear();

Tick.SetError(f2Name, "xong");

pname = true;

}

}

private void f2NgaySinh\_Leave(object sender, EventArgs e)

{

if (DateTime.Now.Year - f2NgaySinh.Value.Year >= 18 && DateTime.Now.Year - f2NgaySinh.Value.Year <= 55)

{

pdate = true;

er.Clear();

Tick.SetError(f2NgaySinh, "xong");

}

else

{

Tick.Clear();

er.SetError(f2NgaySinh, "Độ tuổi yêu cầu từ 18 đến 55!!");

pdate = false;

}

}

private void f2Address\_Leave(object sender, EventArgs e)

{

if (f2Address.Text.Length == 0)

{

Tick.Clear();

er.SetError(this.f2Address, "Bạn chưa nhập địa chỉ! ");

paddress = false;

}

else

{

er.Clear();

Tick.SetError(f2Address, "xong");

paddress = true;

}

}

private void f2Email\_Leave(object sender, EventArgs e)

{

if (Regex.IsMatch(f2Email.Text, @"^([a-zA-Z0-9]+)@([a-zA-Z0-9]+)\.(([a-zA-Z0-9])+)$"))

{

er.Clear();

Tick.SetError(f2Email, "xong");

pmail = true;

}

else

{

pmail = false;

Tick.Clear();

er.SetError(this.f2Email, "Bạn nhập sai định dạng email! "); //Phản hồi ngay sau khi nhập sai email và leave

//pmail = true;

}

}

private void f2Type\_Leave(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(f2Type.Text))

{

er.SetError(this.f2Type, "Vui lòng chọn độc giả! ");

ptype = false;

}

if (f2Type.Text != "X" && f2Type.Text != "Y")

{

er.SetError(this.f2Type, "Vui lòng chọn đúng loại đọc giả sẵn có! ");

ptype = false;

}

else

{

er.SetError(this.f2Type, null);

Tick.SetError(f2Type, "xong");

//checkDocGia.Clear();

ptype = true;

}

}

private void textBox2\_Leave(object sender, EventArgs e)

{

if (textBox2.Text.Length == 0)

{

Tick.Clear();

er.SetError(this.textBox2, "Bạn chưa nhập tên đăng nhập! ");

dn = false;

}

else if (Regex.IsMatch(textBox2.Text, "^[a-zA-Z0-9]\*$"))

{

er.Clear();

Tick.SetError(textBox2, "xong");

dn = true;

}

else

{

Tick.Clear();

er.SetError(textBox2, "Không đúng dữ liệu nhập");

dn = false;

}

}

private void textBox1\_Leave(object sender, EventArgs e)

{

if (textBox1.Text.Length == 0)

{

Tick.Clear();

er.SetError(this.textBox1, "Bạn chưa nhập mật khẩu! ");

mk = false;

}

else if (textBox1.Text.Length < 9)

{

Tick.Clear();

er.SetError(textBox1, "Mật khẩu quá ngắn");

mk = false;

}

else if (Regex.IsMatch(textBox1.Text, "^[!-~]\*$"))

{

er.Clear();

Tick.SetError(textBox1, "xong");

mk = true;

}

else

{

Tick.Clear();

er.SetError(textBox1, "Không đúng dữ liệu nhập");

mk = false;

}

}

private void QLDG\_Load(object sender, EventArgs e)

{

this.ActiveControl = f2Name; // Con trỏ đặt ngay tại Name

string conString = ConfigurationManager.ConnectionStrings["QuanLyDG"].ConnectionString.ToString();

con = new SqlConnection(conString);

con.Open();

Hienthi();

SizeBang = dataTheDocGia.Rows.Count;

f2NgaySinh.Text = DateTime.Today.ToString();

for (int k = 0; k < dataTheDocGia.Rows.Count - 1; k++)

{

string chuoi = dataTheDocGia.Rows[k].Cells[0].Value.ToString();

string chuoi\_so = chuoi.Substring(2);

// MessageBox.Show($"{chuoi\_so}");

int so = int.Parse($"{chuoi\_so}");

//MessageBox.Show($"{(so).ToString()}");

if (so != i) break;

else i++;

}

//MessageBox.Show($"{i.ToString()}");

er.SetError(f2Name, null);

buttonSua.Enabled = false;

buttonXoa.Enabled = false;

ThongKe();

}

private void QLDG\_FormClosing(object sender, FormClosingEventArgs e)

{

if (MessageBox.Show("Trở về \"Đăng nhập\" ", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) != System.Windows.Forms.DialogResult.OK)

{

e.Cancel = true;

con.Close();

}

}

private void buttonSign\_up\_Click(object sender, EventArgs e)

{

if (paddress == true && pdate == true && pmail == true && pname == true && dn == true && mk == true && ptype == true)

{

string sql\_in = $@"Insert into TheDocGia values('{TaoMa()}','{textBox2.Text}','{textBox1.Text}',N'{f2Name.Text}','{f2NgaySinh.Value.ToString("yyyyMMdd")}',N'{f2Address.Text}','{f2Type.Text}','{f2Email.Text}','{DateTime.Today.ToString("yyyyMMdd")}',0,'E.001')";

SqlCommand cmd = new SqlCommand(sql\_in, con);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Đăng kí thành công", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);

Reset\_ThuocTinh();

Set\_false();

SizeBang++;

ThongKe();

Hienthi();

i++;

}

catch

{

MessageBox.Show("Đăng kí thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else MessageBox.Show("Đăng kí thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

int dem = 0;

private void button\_Eye\_Click(object sender, EventArgs e)

{

dem++;

if (dem % 2 == 1) { textBox1.UseSystemPasswordChar = false; button\_Eye.Image = global::QLDG.Properties.Resources.\_1490793853\_user\_interface15\_82360; }

else { textBox1.UseSystemPasswordChar = true; button\_Eye.Image = global::QLDG.Properties.Resources.\_1490793864\_user\_interface16\_82335; }

}

private void dataTheDocGia\_CellClick(object sender, DataGridViewCellEventArgs e)

{

/\*select MS as N'Mã độc giả'," +

"G.TenDN as N'Tên đăng nhập'," +

"G.HoTen as N'Họ và tên'," +

"CONVERT (varchar(10), G.NgaySinh, 103) AS N'Ngày sinh'," +

"G.DiaChi as N'Địa chỉ'," +

"G.Loai as N'Loại'," +

"G.Email," +

"CONVERT (varchar(10), G.NgayLapThe, 103) AS N'Ngày lập thẻ'," +

"G.TongNo as N'Tổng nợ'," +

"S.HoTen as N'Tên người lập' " +

"from TheDocGia G,HoSo S where MaNgLap=MaNV ";\*/

try

{

buttonSua.Enabled = true;

buttonXoa.Enabled = true;

f2Name.Text = dataTheDocGia.SelectedCells[2].Value.ToString();

f2Address.Text = dataTheDocGia.SelectedCells[4].Value.ToString();

textBox2.Text = dataTheDocGia.SelectedCells[1].Value.ToString();

f2Email.Text = dataTheDocGia.SelectedCells[6].Value.ToString();

f2Type.Text = dataTheDocGia.SelectedCells[5].Value.ToString();

f2NgaySinh.Value = DateTime.ParseExact(dataTheDocGia.SelectedCells[3].Value.ToString(), "dd/MM/yyyy", CultureInfo.InvariantCulture);

textMS.Text = dataTheDocGia.SelectedCells[0].Value.ToString();

buttonSign\_up.Enabled = false;

//f2Type.Enabled = false;

Set\_true();

}

catch

{

buttonSua.Enabled = false;

buttonXoa.Enabled = false;

buttonSign\_up.Enabled = true;

Reset\_ThuocTinh();

Set\_false();

Hienthi();

f2NgaySinh.Value = DateTime.Now;

}

}

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

{

if (paddress == true && pdate == true && pmail == true && pname == true && dn == true && mk == true && ptype == true)

{

er.SetError(f2Name, null);

string sql\_ed = $"update TheDocGia set TenDN='{textBox2.Text}',HoTen=N'{f2Name.Text}',NgaySinh='{f2NgaySinh.Value.ToString("yyyyMMdd")}',DiaChi=N'{f2Address.Text}',Loai='{f2Type.Text}',Email='{f2Email.Text}',TongNo=0 where MS='{textMS.Text}'";

SqlCommand cmd = new SqlCommand(sql\_ed, con);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Sửa thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

Hienthi();

}

catch

{

MessageBox.Show("Sửa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else MessageBox.Show("Sửa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

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

{

if (comboBox1.SelectedIndex == -1) MessageBox.Show("Vui lòng chọn phương thức tìm !", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

else if (comboBox1.SelectedIndex == 4)

{

buttonSua.Enabled = false;

buttonXoa.Enabled = false;

buttonSign\_up.Enabled = true;

Reset\_ThuocTinh();

Set\_false();

Hienthi();

f2NgaySinh.Value = DateTime.Now;

}

else

{

string sqlFind = "select MS as N'Mã độc giả'," +

"G.TenDN as N'Tên đăng nhập'," +

"G.HoTen as N'Họ và tên'," +

"CONVERT (varchar(10), G.NgaySinh, 103) AS N'Ngày sinh'," +

"G.DiaChi as N'Địa chỉ'," +

"G.Loai as N'Loại'," +

"G.Email," +

"CONVERT (varchar(10), G.NgayLapThe, 103) AS N'Ngày lập thẻ'," +

"G.TongNo as N'Tổng nợ'," +

"S.HoTen as N'Tên người lập' " +

"from TheDocGia G,HoSo S where MaNgLap=MaNV ";

if (comboBox1.SelectedIndex == 0) sqlFind += $"and MS ='{text\_tìm.Text}'";

else if (comboBox1.SelectedIndex == 1) sqlFind += $"and G.HoTen like N'%{text\_tìm.Text}%'";

else if (comboBox1.SelectedIndex == 2) sqlFind += $"and Loai ='{text\_tìm.Text}'";

else if (comboBox1.SelectedIndex == 3) sqlFind += $"and DATEDIFF(day,NgayLapThe,Getdate()) > 180";

SqlCommand cmd = new SqlCommand(sqlFind, con);

cmd.ExecuteNonQuery();

SqlDataReader rd = cmd.ExecuteReader();

data d = new data();

d.Load(rd);

dataTheDocGia.DataSource = d;

}

}

private void button3\_Click(object sender, EventArgs e)

{

if (paddress == true && pdate == true && pmail == true && pname == true && dn == true && mk == true && ptype == true)

{

/\* public string MS;

public string Ten;

public string NgaySinh;

public string NgayLapThe;

public string Loai;

public string DiaChi;

public string Email;

public string NgLap;\*/

var x = new XuatThe();

x.MS = textMS.Text;

x.Ten = f2Name.Text;

x.Loai = f2Type.Text;

x.DiaChi = f2Address.Text;

x.NgayLapThe = dataTheDocGia.SelectedCells[7].Value.ToString();

x.NgaySinh = f2NgaySinh.Value.ToString("dd/MM/yyyy");

x.Email = f2Email.Text;

x.NgLap= dataTheDocGia.SelectedCells[9].Value.ToString();

this.Hide();

x.ShowDialog();

this.Show();

}

else MessageBox.Show("Thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

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

{

string sqlXoa;

sqlXoa = $"DELETE FROM TheDocGia WHERE MS='{textMS.Text}'";

if (MessageBox.Show($"Có thật sự muốn xóa \"{textMS.Text}-{f2Name.Text}\"", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.OK)

{

SqlCommand cmd = new SqlCommand(sqlXoa, con);

cmd.ExecuteNonQuery();

Hienthi();

Reset\_ThuocTinh();

SizeBang--;

ThongKe();

}

}

private void button\_Excel\_Click(object sender, EventArgs e)

{

var x = new XuatTable();

x.BangThongTin = dataTheDocGia;

this.Hide();

x.ShowDialog();

this.Show();

}

}

}

## **Lớp Sách**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

using data = System.Data.DataTable;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Runtime.CompilerServices;

using System.Data.Entity.Migrations;

namespace QuanLyKho

{

public partial class QuanLyKho : Form

{

public QuanLyKho()

{

InitializeComponent();

AutoValidate = AutoValidate.EnableAllowFocusChange;

}

//========================================================Hàm bổ trợ ===================================================

SqlConnection con;

public int SizeBang;

public int i = 1;

public void HienThi()

{

SqlCommand cmd = new SqlCommand("select MaSach as N'Mã sách'," +

"TenSach as N'Tên sách'," +

"TheLoai as N'Thể loại'," +

"TacGia as N'Tác giả'," +

"NamXuatBan as N'Năm xuất bản'," +

"NhaXuatBan as N'Nhà xuất bản'," +

"CONVERT (varchar(10), NgayNhap, 103) AS N'Ngày nhập'," +

"TriGia as N'Trị giá'," +

"TinhTrang as N'Tình trạng'," +

"HoSo.HoTen as N'Họ và tên người nhận' " +

"from DanhSachSach,HoSo " +

"where DanhSachSach.MaNgNhan = HoSo.MaNV", con);

SqlDataReader rd = cmd.ExecuteReader();

data d = new data();

d.Load(rd);

dataKho.DataSource = d;

}

public void ThongKe()

{

double thongke = (double)((SizeBang - 1) \* 0.2);

progressBar\_thongke.Value = (int)Math.Round(thongke, 0);

// MessageBox.Show($"{progressBar\_thongke.Value.ToString()}");

label\_thongke.Text = $"{(SizeBang - 1).ToString()}/500";

}

public string TaoMa()

{

if (i < 10) return $"B.00{i.ToString()}";

else if (i < 100) return $"B.0{i.ToString()}";

else return $"B.{i.ToString()}";

}

public bool Only\_letter(string x)

{

char[] DS = { '\'','\\',',','.','/','<','>',';',':','\"','{','[','}',']',

'~','`','!','@','#','$','%','^','&','\*','(',')','-','\_','+','=',};

for (int i = 0; i < x.Length; i++)

{

if (DS.Contains(x[i]))

{

return false;

}

else if (Int32.TryParse(x[i].ToString(), out int y))

{

return false;

}

}

return true;

}

public void Reset\_ThuocTinh()

{

Kho\_MaSach.Text = "";

Kho\_TenSach.Text = "";

Kho\_TheLoai.SelectedIndex = -1;

Kho\_TacGia.Text = "";

Kho\_NamXuatBan.Value = Kho\_NamXuatBan.Minimum;

Kho\_NhaXuatBan.Text = "";

Kho\_Gia.Value = Kho\_Gia.Minimum;

}

//------------------------------------------------------------Xử lý chức năng----------------------------------------

private void QuanLyKho\_Load(object sender, EventArgs e)

{

string conString = ConfigurationManager.ConnectionStrings["QuanLyDG"].ConnectionString.ToString();

con = new SqlConnection(conString);

con.Open();

HienThi();

for (int k = 0; k < dataKho.Rows.Count - 1; k++)

{

string chuoi = dataKho.Rows[k].Cells[0].Value.ToString();

string chuoi\_so = chuoi.Substring(2);

// MessageBox.Show($"{chuoi\_so}");

int so = int.Parse($"{chuoi\_so}");

//MessageBox.Show($"{(so).ToString()}");

if (so != i) break;

else i++;

}

SizeBang = dataKho.Rows.Count;

//MessageBox.Show($"{i.ToString()}");

button\_Sua.Enabled = false;

button\_Xoa.Enabled = false;

button\_Them.Enabled = true;

ThongKe();

}

private void button\_Tim\_Click(object sender, EventArgs e)

{

if (comboBox\_Tim.SelectedIndex == -1) MessageBox.Show("Vui lòng chọn cách tìm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

else if (comboBox\_Tim.SelectedIndex == 5)

{

button\_Them.Enabled = true;

button\_Sua.Enabled = false;

button\_Xoa.Enabled = false;

Reset\_ThuocTinh();

HienThi();

}

else

{

string sql\_find = "select MaSach as N'Mã sách'," +

"TenSach as N'Tên sách'," +

"TheLoai as N'Thể loại'," +

"TacGia as N'Tác giả'," +

"NamXuatBan as N'Năm xuất bản'," +

"NhaXuatBan as N'Nhà xuất bản'," +

"CONVERT (varchar(10), NgayNhap, 103) AS N'Ngày nhập'," +

"TriGia as N'Trị giá'," +

"TinhTrang as N'Tình trạng'," +

"HoSo.HoTen as N'Họ và tên người nhận' " +

"from DanhSachSach,HoSo " +

"where DanhSachSach.MaNgNhan = HoSo.MaNV";

if (comboBox\_Tim.SelectedIndex == 0) sql\_find += $" and MaSach='{Kho\_MaSach.Text}'";

else if (comboBox\_Tim.SelectedIndex == 1) sql\_find += $" and TenSach like N'%{Kho\_TenSach.Text}%'";

else if (comboBox\_Tim.SelectedIndex == 2) sql\_find += $" and TheLoai='{Kho\_TheLoai.Text}'";

else if (comboBox\_Tim.SelectedIndex == 3) sql\_find += $" and TacGia like N'%{Kho\_TacGia.Text.Trim()}%'";

else sql\_find += $" and YEAR(GETDATE())-NamXuatBan > 8";

SqlCommand cmd = new SqlCommand(sql\_find, con);

cmd.ExecuteNonQuery();

SqlDataReader rd = cmd.ExecuteReader();

data d = new data();

d.Load(rd);

dataKho.DataSource = d;

}

}

private void comboBox\_Tim\_SelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox\_Tim.SelectedIndex == 0)

{

Kho\_Gia.Enabled = false;

Kho\_NamXuatBan.Enabled = false;

Kho\_TenSach.Enabled = false;

Kho\_TheLoai.Enabled = false;

Kho\_MaSach.Enabled = true;

Kho\_TacGia.Enabled = false;

Kho\_NhaXuatBan.Enabled = false;

}

else if (comboBox\_Tim.SelectedIndex == 1)

{

Kho\_Gia.Enabled = false;

Kho\_NamXuatBan.Enabled = false;

Kho\_TenSach.Enabled = true;

Kho\_TheLoai.Enabled = false;

Kho\_MaSach.Enabled = false;

Kho\_TacGia.Enabled = false;

Kho\_NhaXuatBan.Enabled = false;

}

else if (comboBox\_Tim.SelectedIndex == 3)

{

Kho\_Gia.Enabled = false;

Kho\_NamXuatBan.Enabled = false;

Kho\_TenSach.Enabled = false;

Kho\_TheLoai.Enabled = false;

Kho\_MaSach.Enabled = false;

Kho\_TacGia.Enabled = true;

Kho\_NhaXuatBan.Enabled = false;

}

else if (comboBox\_Tim.SelectedIndex == 2)

{

Kho\_Gia.Enabled = false;

Kho\_NamXuatBan.Enabled = false;

Kho\_TenSach.Enabled = false;

Kho\_TheLoai.Enabled = true;

Kho\_MaSach.Enabled = false;

Kho\_TacGia.Enabled = false;

Kho\_NhaXuatBan.Enabled = false;

}

else

{

Kho\_Gia.Enabled = true;

Kho\_NamXuatBan.Enabled = true;

Kho\_TenSach.Enabled = true;

Kho\_TheLoai.Enabled = true;

Kho\_MaSach.Enabled = false;

Kho\_TacGia.Enabled = true;

Kho\_NhaXuatBan.Enabled = true;

}

}

private void Kho\_NamXuatBan\_Validating(object sender, CancelEventArgs e)

{

if (DateTime.Now.Year - Kho\_NamXuatBan.Value <= 8 && DateTime.Now.Year >= Kho\_NamXuatBan.Value)

{

e.Cancel = false;

errorProvider1.SetError(Kho\_NamXuatBan, null);

Tick.SetError(Kho\_NamXuatBan, "xong");

}

else

{

e.Cancel = true;

Tick.Clear();

errorProvider1.SetError(Kho\_NamXuatBan, "Yêu cầu sách xuất bản trong 8 năm đổi lại");

}

}

private void Kho\_TacGia\_Validating(object sender, CancelEventArgs e)

{

try

{

if (Kho\_TacGia.Text.Length != 0)

{

string fullname = Kho\_TacGia.Text;

string space = " ";

// Xử lý cắt dấu cách đầu tên

bool dau = false; //dung de check dau cach dung dau

for (int i = 0; i < fullname.Length && dau == false; i++)

{

if (fullname[i] == ' ' && dau == false)

{

fullname = fullname.Remove(i, 1);

i = -1;

}

else dau = true;

}

if (Kho\_TacGia.Text.Length == 0)

{

errorProvider1.SetError(this.Kho\_TacGia, "Vui lòng nhập lại!");

e.Cancel = true;

}

//Xử lý cắt dấu cách cuối tên

bool cuoi = false;

for (int i = fullname.Length - 1; i >= 0 && cuoi == false; i--)

{

if (fullname[i] == ' ' && cuoi == false)

{

fullname = fullname.Remove(fullname.Length - 1, 1);

i = fullname.Length;

}

else cuoi = true;

}

if (Kho\_TacGia.Text.Length == 0)

{

errorProvider1.SetError(this.Kho\_TacGia, "Vui lòng nhập lại!");

e.Cancel = true;

}

// Xử lý phần dấu cách trước và sau chuỗi xong

Kho\_TacGia.Text = fullname; // gán lại chuỗi sau khi sửa

//Tiếp tục check phần ký tự

for (int i = 0; i < Kho\_TacGia.Text.Length; i++)

{

if (Kho\_TacGia.Text[i] >= 33 && Kho\_TacGia.Text[i] <= 57 || Kho\_TacGia.Text[i] >= 58 && Kho\_TacGia.Text[i] <= 64 || Kho\_TacGia.Text[i] >= 91 && Kho\_TacGia.Text[i] <= 96 || Kho\_TacGia.Text[i] >= 123 && Kho\_TacGia.Text[i] <= 126)

{

Kho\_TacGia.Text = Kho\_TacGia.Text.Remove(i, 1);

i = -1;

}

}

if (Kho\_TacGia.Text.Length == 0)

{

Tick.Clear();

errorProvider1.SetError(this.Kho\_TacGia, "Vui lòng nhập lại!");

e.Cancel = true;

}

if (Kho\_TacGia.Text.Length != 0)

{

errorProvider1.Clear();

//Đếm số lượng từ có trong tên để viết hoa chữ đầu

int slc = 1;

for (int i = 0; i < Kho\_TacGia.Text.Length; i++)

if (Kho\_TacGia.Text[i] == ' ') slc++;

//Viết hoa chữ đầu và nhấn đầu chữ sau

string[] parts = Kho\_TacGia.Text.Split(new string[] { space }, StringSplitOptions.RemoveEmptyEntries);

Kho\_TacGia.Text = "";

bool hetchu = false;

for (int i = 0; i < slc; i++)

{

parts[i] = parts[i].Substring(0, 1).ToUpper() + parts[i].Substring(1, parts[i].Length - 1).ToLower();

Kho\_TacGia.Text = Kho\_TacGia.Text + parts[i];

if (i >= 0 && i < slc) hetchu = true;

if (hetchu == true) Kho\_TacGia.Text += " ";

}

}

}

if (Kho\_TacGia.Text.Length == 0)

{

Tick.Clear();

errorProvider1.SetError(this.Kho\_TacGia, "Bạn chưa nhập tên! ");

e.Cancel = true;

}

else

{

errorProvider1.Clear();

Tick.SetError(Kho\_TacGia, "xong");

e.Cancel = false;

}

}

catch

{

Tick.Clear();

errorProvider1.SetError(this.Kho\_TacGia, "Vui lòng nhập lại!");

e.Cancel = true;

}

}

private void Kho\_NhaXuatBan\_Validating(object sender, CancelEventArgs e)

{

if (!Only\_letter(Kho\_NhaXuatBan.Text) || Kho\_NhaXuatBan.Text.Length == 0)

{

e.Cancel = true;

errorProvider1.SetError(Kho\_NhaXuatBan, "Vui lòng nhập đúng yêu cầu");

Tick.Clear();

}

else

{

e.Cancel = false;

errorProvider1.SetError(Kho\_NhaXuatBan, null);

Tick.SetError(Kho\_NhaXuatBan, "xong");

}

}

private void Kho\_TheLoai\_Validating(object sender, CancelEventArgs e)

{

if (Kho\_TheLoai.SelectedIndex == -1)

{

Tick.Clear();

errorProvider1.SetError(Kho\_TheLoai, "Vui lòng chọn thể loại sách");

e.Cancel = true;

}

else

{

errorProvider1.Clear();

Tick.SetError(Kho\_TheLoai, "xong");

e.Cancel = false;

}

}

private void Kho\_TenSach\_Validating(object sender, CancelEventArgs e)

{

if (!(Kho\_TenSach.Text.Length == 0))

{

errorProvider1.SetError(Kho\_TenSach, null);

Tick.SetError(Kho\_TenSach, "xong");

e.Cancel = false;

}

else

{

Tick.SetError(Kho\_TenSach, null);

e.Cancel = true;

errorProvider1.SetError(Kho\_TenSach, "Vui lòng nhập tên sách");

}

}

private void Kho\_Gia\_Validating(object sender, CancelEventArgs e)

{

if (Kho\_Gia.Value - Kho\_Gia.Minimum <= 0)

{

Tick.Clear();

e.Cancel = true;

errorProvider1.SetError(Kho\_MenhGia, "Chưa nhập giá sách");

}

else

{

e.Cancel = false;

errorProvider1.SetError(Kho\_MenhGia, null);

Tick.SetError(Kho\_MenhGia, "xong");

}

}

private void QuanLyKho\_FormClosing(object sender, FormClosingEventArgs e)

{

AutoValidate = AutoValidate.Disable;

Application.Exit();

con.Close();

}

private void button\_Them\_Click(object sender, EventArgs e)

{

if (ValidateChildren(ValidationConstraints.Enabled))

{

string sql\_in = $@"Insert into DanhSachSach values(

'{TaoMa()}',

N'{Kho\_TenSach.Text}',

N'{Kho\_TacGia.Text}',

'{Kho\_TheLoai.Text}',

'{Kho\_NamXuatBan.Value.ToString()}',

N'{Kho\_NhaXuatBan.Text}',

'{DateTime.Today.ToString("yyyyMMdd")}',

{Kho\_Gia.Value \* 1000},

N'Còn',

'E.002'

)";

SqlCommand cmd = new SqlCommand(sql\_in, con);

cmd.ExecuteNonQuery();

SizeBang++;

ThongKe();

i++;

Reset\_ThuocTinh();

HienThi();

if (MessageBox.Show("Thêm thành công. Bạn có muốn xuất thông tin sách - PDF không ?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.OK)

{

var x = new XuatThongTinSach();

x.MaSach = TaoMa();

x.TheLoai = Kho\_TheLoai.Text;

x.TenSach = Kho\_TenSach.Text;

x.TacGia = Kho\_TacGia.Text;

x.NamXuatBan = Kho\_NamXuatBan.Text;

x.NhaXuatBan = Kho\_NhaXuatBan.Text;

x.TriGia = Kho\_Gia.Value.ToString()+".000 đồng";

x.NgayNhap = DateTime.Today.ToString("dd/MM/yyyy");

x.NguoiTiepNhan = "??";

this.Hide();

x.ShowDialog();

this.Show();

}

}

else MessageBox.Show("Đăng kí thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

private void button\_Xoa\_Click(object sender, EventArgs e)

{

string sqlXoa;

sqlXoa = $"DELETE FROM DanhSachSach WHERE MaSach='{Kho\_MaSach.Text}'";

if (MessageBox.Show($"Có thật sự muốn xóa \"{Kho\_MaSach.Text}-{Kho\_TenSach.Text}\"", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.OK)

{

SqlCommand cmd = new SqlCommand(sqlXoa, con);

cmd.ExecuteNonQuery();

HienThi();

Reset\_ThuocTinh();

SizeBang--;

ThongKe();

}

}

private void button\_Sua\_Click(object sender, EventArgs e)

{

if (ValidateChildren(ValidationConstraints.Enabled))

{

string sql\_ed = $"update DanhSachSach set TenSach=" +

$"N'{Kho\_TenSach.Text}',TacGia=N'{Kho\_TacGia.Text}'," +

$"TheLoai='{Kho\_TheLoai.Text}',NamXuatBan={Kho\_NamXuatBan.Value}," +

$"NhaXuatBan=N'{Kho\_NhaXuatBan.Text}',TriGia={Kho\_Gia.Value \* 1000}" +

$"where MaSach='{Kho\_MaSach.Text}'";

SqlCommand cmd = new SqlCommand(sql\_ed, con);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Sửa thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

HienThi();

}

catch

{

MessageBox.Show("Sửa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else MessageBox.Show("Sửa thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

private void button\_PDF\_Click(object sender, EventArgs e)

{

if (ValidateChildren(ValidationConstraints.Enabled))

{

var x = new XuatThongTinSach();

x.MaSach = Kho\_MaSach.Text;

x.TheLoai = Kho\_TheLoai.Text;

x.TenSach = Kho\_TenSach.Text;

x.TacGia = Kho\_TacGia.Text;

x.NamXuatBan = Kho\_NamXuatBan.Text;

x.NhaXuatBan = Kho\_NhaXuatBan.Text;

x.TriGia = Kho\_Gia.Value.ToString() + ".000 đồng";

x.NgayNhap = dataKho.SelectedCells[6].Value.ToString();

x.NguoiTiepNhan = dataKho.SelectedCells[8].Value.ToString();

this.Hide();

x.ShowDialog();

this.Show();

}

else

{

MessageBox.Show("Thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button\_Excel\_Click(object sender, EventArgs e)

{

var x = new XuatKhoSach();

x.khoSach = dataKho;

this.Hide();

x.ShowDialog();

this.Show();

}

private void dataKho\_CellClick\_1(object sender, DataGridViewCellEventArgs e)

{

var dgv = sender as DataGridView;

try

{

if (dgv?.CurrentRow?.Cells != null)

{

// MessageBox.Show("sá");

Kho\_Gia.Enabled = true;

Kho\_NamXuatBan.Enabled = true;

Kho\_TenSach.Enabled = true;

Kho\_TheLoai.Enabled = true;

Kho\_MaSach.Enabled = false;

Kho\_TacGia.Enabled = true;

Kho\_NhaXuatBan.Enabled = true;

Kho\_MaSach.Text = dataKho.SelectedCells[0].Value.ToString();

Kho\_TenSach.Text = dataKho.SelectedCells[1].Value.ToString();

Kho\_TheLoai.Text = dataKho.SelectedCells[2].Value.ToString();

Kho\_TacGia.Text = dataKho.SelectedCells[3].Value.ToString();

Kho\_NamXuatBan.Value = int.Parse(dataKho.SelectedCells[4].Value.ToString());

Kho\_NhaXuatBan.Text = dataKho.SelectedCells[5].Value.ToString();

double money = double.Parse(dataKho.SelectedCells[7].Value.ToString());

Kho\_Gia.Value = int.Parse((money \* 0.001).ToString());

button\_Them.Enabled = false;

button\_Sua.Enabled = true;

button\_Xoa.Enabled = true;

}

else

{

button\_Them.Enabled = true;

button\_Sua.Enabled = false;

button\_Xoa.Enabled = false;

Reset\_ThuocTinh();

}

}

catch

{

button\_Them.Enabled = true;

button\_Sua.Enabled = false;

button\_Xoa.Enabled = false;

Reset\_ThuocTinh();

}

}

private void button1\_Click(object sender, EventArgs e)

{

var x = new DanhSachSach();

x.Dss = dataKho;

this.Hide();

x.ShowDialog();

this.Show();

}

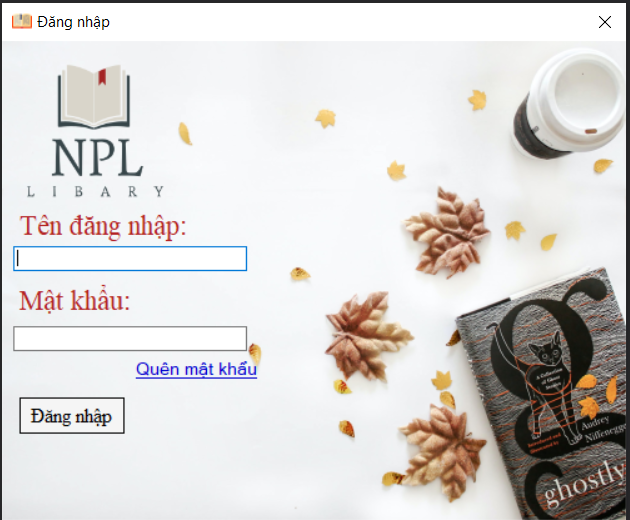
}

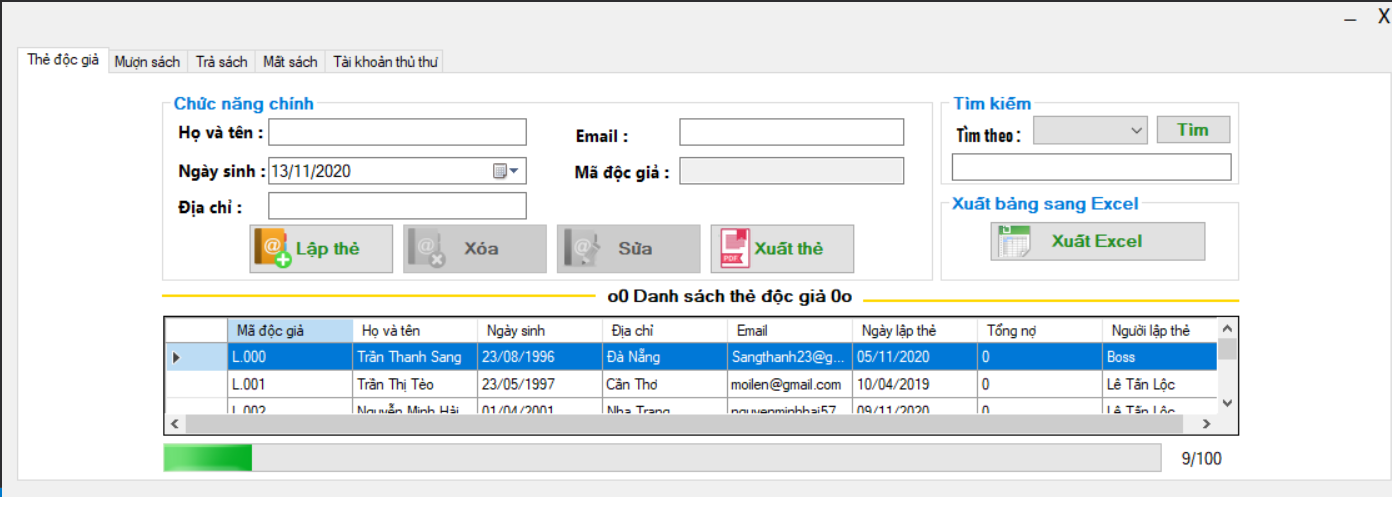
}

# CHƯƠNG 4: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

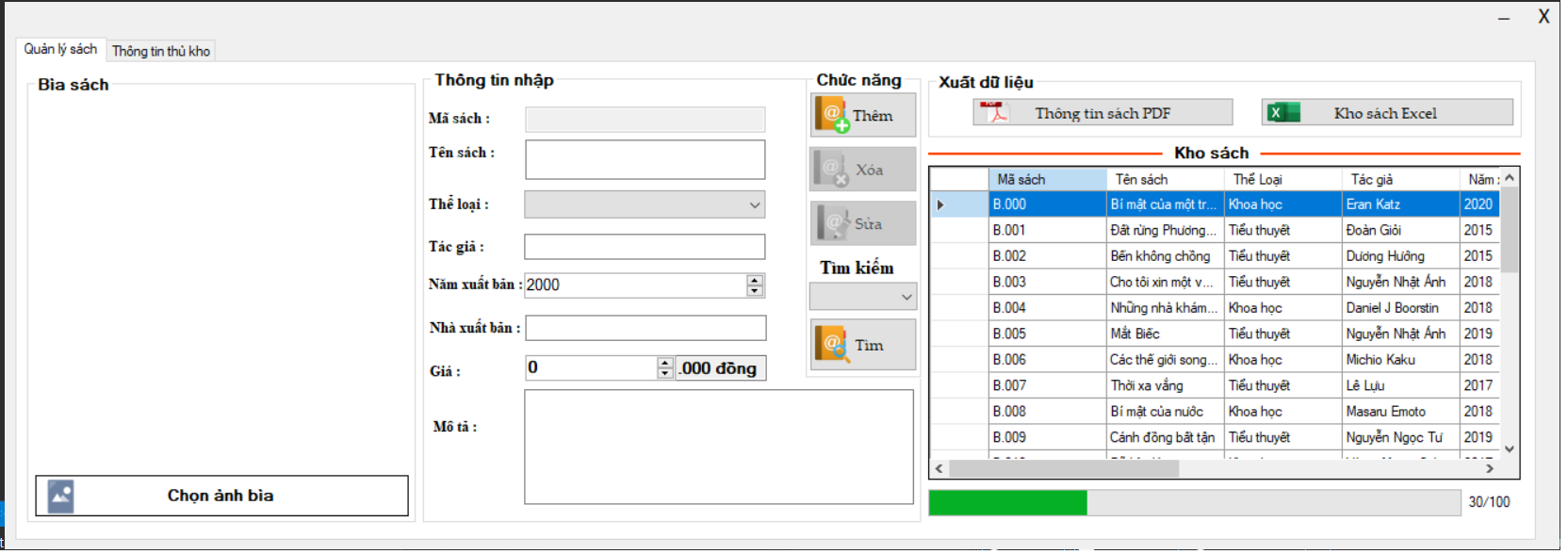
Sau đây nhóm sẽ hướng dẫn sử dụng phần mềm Quản lý điểm thư viện như sau:

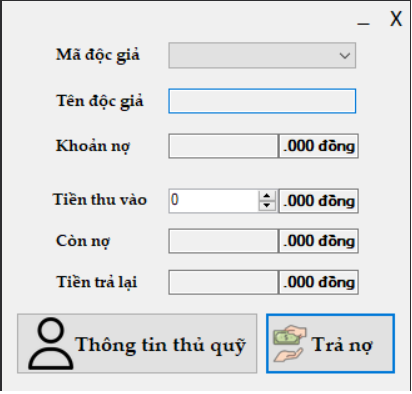
Đầu tiên là màn hình đăng nhập. Cần nhập đúng tên chọn vai trò(bộ phận) của mình trong thư viện và tài khoản để sử dụng



Nếu người đăng nhập dưới vai trò của thủ thư sẽ mặc định dẫn tới form quản lý độc giả

Nếu người đăng nhập dưới vai trò của thủ kho thì sẽ là form Quản lý kho

Nếu người đăng nhập là thủ quỹ sẽ vào để thanh toán chi phí với độc giả



Nếu người đăng nhập là thủ trưởng sẽ vào form sau:



# CHƯƠNG 5: ĐÁNH GIÁ PHẦN MỀM QUẢN LÝ THƯ VIỆN

# Tổng kết

# 1. **Khả thi về kỹ thuật**

-Thành viên tham gia dự án: 3 thành viên

- Thời gian thực hiện: 3 tuần

- Kích thước hệ thống: trung bình

# 2. **Công cụ phát triển**

- Hệ điều hành Windows 10

- Công cụ lập trình: Microsoft Visual Studio 2019 (ngôn ngữ C sharp)

- Cơ sở dữ liệu: Microsoft SQL Server 2012

- Công cụ khác: Word, PPT…

# 3. **Bảng tổng kết**

| **STT** | **Rủi ro** | **Mức độ** | **Mô tả đánh giá** | **Chú thích** |
| --- | --- | --- | --- | --- |
| 1 | Tâm lý người dùng chưa quen với hệ thống mới | Trung bình | Nhân viên nơi làm việc đều có khả năng sử dụng máy vi tính | Tổ chức đào tạo huấn luyện cho người sử dụng hệ thống |
| Hỗ trợ trực tiếp người dùng trong thời gian bắt đầu.  Tiến hành triển khai thử từng bước. |
| 2 | Kích thước hệ thống | Trung bình | Thời gian triển khai hệ thống là 1 tuần |  |
| 3 | Chuyển từ cách hoạt động từ thủ công sang sử dụng phần mềm | Bình thường | Nhân viên sẽ nhanh chóng thích nghi với việc sử dụng hệ thống mới | Đào tạo một lớp sử dụng phần mềm trước khi sử dụng |
| 4 | Dữ liệu không thể phục hồi khi xóa | Cao | Chọn chức năng xóa | Có cảnh báo |
| 5 | Nhập dữ liệu sai | Khá cao | Trong quá trình nhập không tránh khỏi khả năng nhập sai | Có cảnh báo |
| 6 | Mở thêm tính năng cho phần mềm | Thấp | Khi muốn mở tính năng mới cho phần mềm | Nhóm phát triển phải chỉ cho phép người dùng hoạt động trên những model có sẵn |

1. Yêu cầu bảo mật

Chỉ người có user + password của hệ thống mới có quyền truy cập quản trị hệ thống

# **Ngôn ngữ viết**

Chương trình sử dụng ngôn ngữ C #:

* Phát triển tại Microsoft, đội nghiên cứu do Anders Hejlsberg và Scott Wiltamuth lãnh đạo
* Điều khiển sự kiện, hướng đối tượng, ngôn ngữ lập trình trực quan
* Dựa trên nền tảng C, C++ và Java
* Hợp nhất trong nền .NET platform
* Có thể phân phối các ứng dụng Web
* Các thiết bị và máy tính để bàn
* Các chương trình cho phép mọi người truy cập qua bất cứ thiết bị nào
* Cho phép giao tiếp với các ngôn ngữ máy khác.
* Integrated Design Environment (IDE)-Môi trường tích hợp phát triển phần mềm
* Dễ dàng hoá việc lập trình và gỡ lỗi.
* Phát triển ứng dụng nhanh- Rapid Application Development (RAD)

# KẾT LUẬN

* Ưu điểm:
* Các chức năng cơ bản của chương trình đã hoàn thành.
* Giao diện thân thiện, dễ sử dụng.
* Cài đặt trên nhiều hệ điều hành
* Có trợ giúp hướng dẫn sử dụng
* Nhược điểm
* Dữ liệu cập nhật hơi chậm.
* Độ bảo mật không cao.
* Thời gian hạn chế và nhóm chỉ có 3 thành viên cho nên trong quá trình xây dựng phần mềm còn gặp nhiều khó khăn.
* Chưa có kinh nghiệm nhiều trong việc xây dựng một chương trình quản lý với quy mô lớn.
* Hầu hết các chức năng hệ thống vẫn còn chưa hoàn thành

* Kinh nghiệm thu được:
* Hiểu được nhiều kinh nghiệm trong giao tiếp cũng như công việc thực tế
* Có cơ hội để vận dụng kiến thức lý thuyết trong nhà trường áp dụng vào thực tiễn.
* Hướng giải quyết:

Phần mềm được viết với các chức năng cơ bản và có thể dựa vào nó để nâng cấp lên quản trị thêm nhiều chức năng hơn.

Link GitHub:

# BẢNG PHÂN CÔNG

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ tên** | **Nhiệm vụ** |
| 4501104135 | Lê Tấn Lộc | Lên ý tưởng, bìa báo cáo, chỉnh word và thiết kế giao diện. |
| 4501104172 | Huỳnh Thanh Phong | Phân công chia công việc cho các thành viên, thiết kế giao diện, điều chỉnh PPT, lên ý tưởng. |
| 4501104147 | Võ Hoài Nam | Thiết kế giao diện, soạn word, lên ý tưởng |

# NGUỒN THAM KHẢO:

#### *[1] API là gì? Giới thiệu về Giao diện lập trình ứng dụng API- Freelancervn* (2019). Available at: https://freelancervietnam.vn/gioi-thieu-ve-giao-dien-lap-trinh-ung-dung-api/#:~:text=API%20%E2%80%93%20Application%20Programming%20Interface%20%E2%80%93%20giao,%C4%91%E1%BB%95i%20th%C3%B4ng%20tin%20v%E1%BB%9Bi%20nhau. (Accessed: 22 June 2020).

#### *[2] Lập trình Winform cơ bản | Website hướng dẫn lập trình miễn phí | How Kteam | Free Education* (2020). Available at: https://www.howkteam.vn/course/lap-trinh-winform-co-ban-27 (Accessed: 22 June 2020).

#### [3]Hồng, T., Tiến, P. and Tiến, P. (2015) *Code Export dữ liệu từ DataGrid tới PDF với C# | Tìm ở đây*, *Timoday.edu.vn*. Available at: https://timoday.edu.vn/code-export-du-lieu-tu-datagrid-toi-pdf-voi-c/ (Accessed: 22 June 2020).

[4] link youtube: <https://www.youtube.com/watch?v=NNXNYZWdyrg>

<https://www.youtube.com/watch?v=h7OYMHCa2WE&t=1235s>